



Zavod za telekomunikacije

Poslijediplomski studij  
za stjecanje doktorata  
znanosti

Ak.g. 2009./2010.

# Projektiranje telekomunikacijskih sustava

Telekomunikacijski sustavi –  
umreženi raspodijeljeni sustavi

18.5.2010

- ◆ Uvodno o raspodijeljenim sustavima
  - Definicija i obilježja raspodijeljenih sustava
  - Otvorenost, transparentnost i skalabilnost
- ◆ Teorijski model raspodijeljenog sustava
  - Interakcijski model
  - Model kvara
  - Model sigurnosti
- ◆ Vremensko odvijanje i usklađivanje procesa
- ◆ Replikacija i konzistentnost podataka
- ◆ Otpornost na kvarove
- ◆ Sigurnost: studijski primjer pokretne mreže

---

# Uvodno o raspodijeljenim sustavima

Andrew S. Tannenbaum:

- ◆ Skup neovisnih računala koji korisniku izgleda kao jedan cjeloviti sustav.

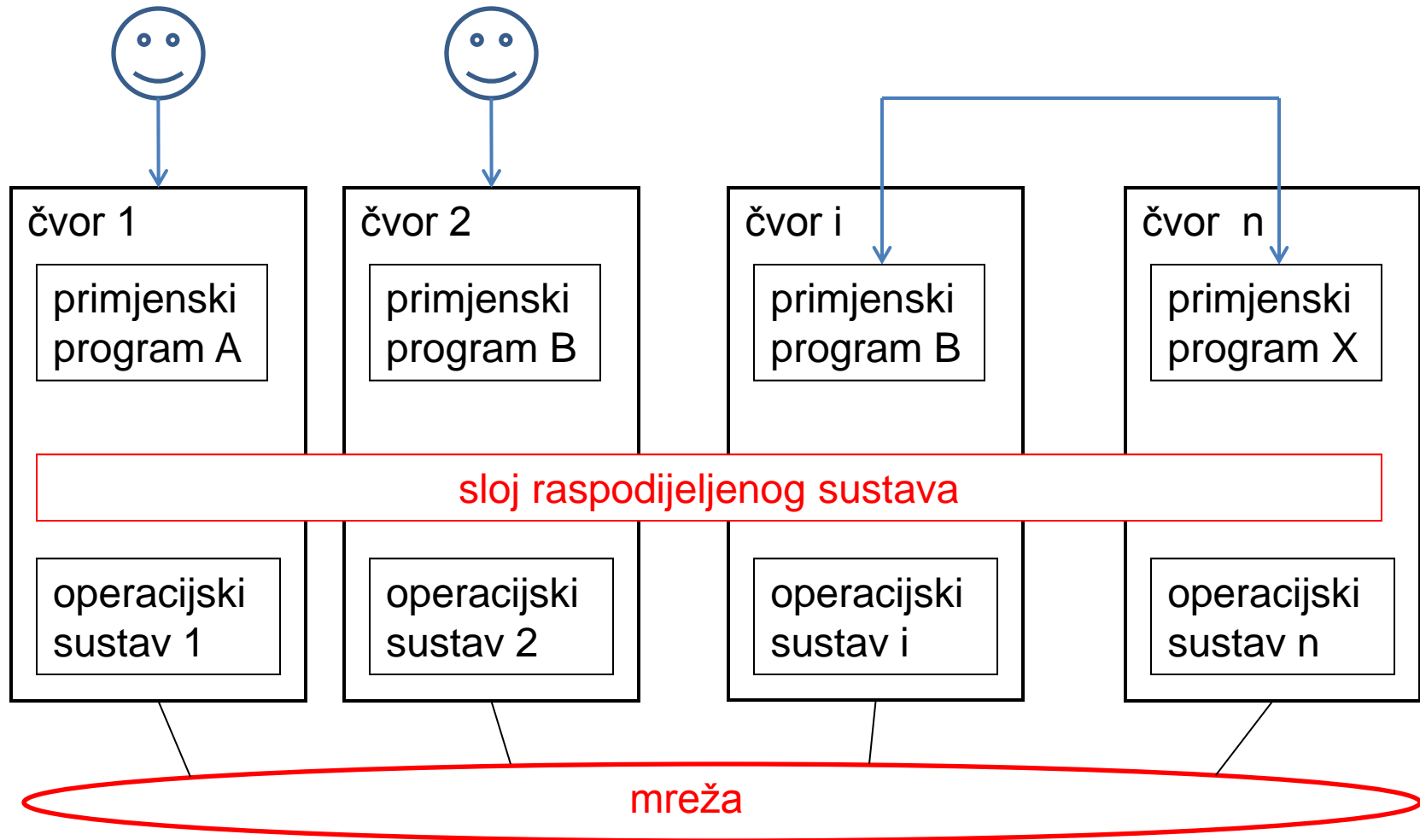
George Coulouris:

- ◆ Sustav u kojem programske i sklopovske komponente umreženih računala komuniciraju i usklađuju svoje aktivnosti isključivo razmjenom poruka.

Leslie Lamport:

- ◆ Sustav u kojem kvar računala za koje uopće ne znate da postoji može učiniti vaše računalo neupotrebljivim.

# Definicije (2)



## Programski posrednički sloj, međuoprema (engl. *middleware*)

- ◆ prikriva činjenicu da su procesi i sredstva (resursi) raspodijeljeni na više čvorova/računala u mreži
- ◆ omogućuje povezivanje i suradnju aplikacija, sustava i uređaja,
- ◆ omogućuje interakciju programa na aplikacijskoj razini,
- ◆ nalazi se između operacijskog sustava i primjenskih programa,
- ◆ nalazi se iznad mrežnog, a ispod aplikacijskog sloja

## Inherentna raspodijeljenost:

- ◆ korisnika, informacija, sredstava, ...

## Funkcionalno odvajanje:

- ◆ različite namjene, različite mogućnosti, različite uloge (korisnik – davatelj usluge, proizvođač – potrošač, prodavač – kupac, ....)

## Opterećenje:

- ◆ raspodjela i uravnoteženje

## Pouzdanost:

- ◆ više komponenata na različitim mjestima

## Cijena, troškovi

## Sustavi za pružanje informacijskih i komunikacijskih usluga:

- ◆ sloj usluga i aplikacija (engl. *Service & Application Layer*)  
ili
- ◆ uslužni stratum (engl. *Service Stratum*)

## Sustavi za transportne usluge:

- ◆ sloj povezivosti (engl. *Connectivity Layer*)  
ili
- ◆ transportni stratum (engl. *Transport Stratum*)

## Prožimajući sustavi (engl. *pervasive system*)

- ◆ senzorska mreža (engl. *sensor network*)

## Paralelne aktivnosti:

- ◆ autonomne komponente sustava istodobno izvode više aktivnosti

## Komunikacija razmjenom poruka:

- ◆ komponente sustava razmjenjuju podatke porukama, ne dohvaćaju ih iz zajedničke memorije

## Dijeljenje sredstava:

- ◆ zajedničkim sredstvima pristupa više komponenata sustava

## Nema globalnog stanja:

- ◆ niti jedan proces ne zna stanje svih procesa u svim komponentama sustava

## Nema globalnog vremenskog takta:

- ◆ sve komponente sustava nisu pokretane istim vremenskim taktom tako da je ograničena mogućnost vremenskog usklađivanja

## ◆ Otvorenost

- otvoreni sustav (engl. *open system*): pruža usluge sukladno normiranim pravilima te definiranoj sintaksi i semantici

## ◆ Transparentnost

- prikrivanje odabranih značajki raspodijeljenog sustava
- utječe na složenost, performanse i troškove sustava
- oblikuje se sukladno korisničkim zahtjevima

## ◆ Skalabilnost

- sposobnost razmjerne prilagodbe veličini (broj korisnika – količina sredstva), rasprostranjenosti (lokalno, regionalno, globalno, ...) i načinu upravljanja (jedna ili više administrativnih domena)

## Norma (standard) je pretpostavka za otvorenost:

- ◆ specifikacija koja je:
  - široko prihvaćena u industriji (**de facto standard**) ili zastupana od normizacijskog tijela (**de jure standard**),
  - dobro definirana,
  - neutralna, tj. vlasnički neovisna i
  - javno dostupna.

## Otvorenost je pretpostavka za:

- ◆ međudjelovanje (engl. *interoperability*)
- ◆ prenosivost (engl. *portability*)
- ◆ proširljivost (engl. *extensibility*)

## Sustavi transportnog stratuma:

- ◆ potpuno otvoreni (normirani)
  - potpuno međudjelovanje
  - prenosivost unutar programa jednog proizvođača
  - proširljivost unutar unaprijed definiranih granica

## Sustavi uslužnog stratuma:

- ◆ djelomično otvoreni: problem aplikacijskih programskih sučelja (*Application Programme Interface, API*)
  - mnogo vlasničkih rješenja koja ograničavaju međudjelovanje
  - prenosivost unutar programa jednog proizvođača
  - proširljivost unutar unaprijed definiranih granica

## Transparentnost pristupa (engl. *access transparency*)

- ◆ Prikrivanje razlika u predočavanju podataka i pristupa sredstvima (različite arhitekture, operacijski sustavi, baze podataka)

## Lokacijska transparentnost (engl. *location transparency*)

- ◆ Prikrivanje lokacije sredstva: položaj sredstva u sustavu ne treba biti i nije poznat korisniku
- ◆ Imenovanje sredstva omogućuje dostup sredstvu putem imena, što omogućuje promjenu njegove lokacije, ali ne tijekom uporabe

## Migracijska transparentnost (engl. *migration transparency*)

- ◆ Prikrivanje promjene lokacije: promjena lokacije sredstva ne utječe na način dostupa sredstvu

## Relokacijska transparentnost (engl. *relocation transparency*)

- ◆ Prikrivanje premještanja/kretanja sredstva: sredstvo smije mijenjati položaj tijekom uporabe

## Replikacijska transparentnost (engl. *replication transparency*)

- ◆ Prikrivanje više istovrsnih sredstava ili više preslika nekog sredstva, što zahtijeva isto ime za sve replike
- ◆ Lokacijska transparentnost pretpostavka je replikacijsku

**Konkurencijska transparentnost** (engl. *concurrency transparency*)

- ◆ Prikrivanje istodobne uporabe istog resursa od drugih korisnika: zajednička/dijeljena uporaba sredstva uz očuvanje konzistentnosti

**Transparentnost na kvar** (engl. *failure transparency*)

- ◆ Prikrivanje kvara: otkrivanje kvara i obnavljanje sustava nakon kvara nije uočljivo korisnicima
- ◆ Problem otkrivanja kvara: veliko opterećenje može se očitovati kao kvar (npr. nema odgovora u očekivanom vremenu)

Da bi se uz promjenu broja korisnika održale performanse sustava uz prihvatljive troškove treba:

više (istovrsnih) dijelova

koliko?

prostorno raspodijeljenih

gdje?

koji komuniciraju asinkrono

kako?

Oblikovanje raspodijeljenih sustava:

- ◆ Kakav je stupanj transparentnosti potreban i kako utječe na performanse?
- ◆ Kakva je skalabilnost sustava s motrišta veličine, rasprostranjenosti i upravljanja potrebna?

---

Studijski primjer:

Raspodijeljeni sustav weba

- ◆ Internetska aplikacija
- ◆ Model klijent – poslužitelj
- ◆ Otvoreni sustav
- ◆ Transformacija weba:
  - od “korisnik čita sadržaj” prema “korisnik stvara sadržaj”  
(Web 2.0)
  - od “korisnik odabire sadržaj” prema “korisnik traži uslugu”  
(usluge weba, *Web Services*)
  - sljedeći korak: uslužni web (Service Web 3.0)

## Transparentnost pristupa (engl. *access transparency*)

- ◆ predočavanje podataka (HTML) i pristupa sredstvima (HTTP)

## Lokacijska transparentnost:

- ◆ simbolička imenima koja se u sustavu domenskih imena (DNS) prevode u lokacije poslužitelja (mrežne adrese):
  - korisnik rabi simbolička imena, položaj poslužitelja (sjedišta weba) kao i bilo kojeg resursa ne treba biti je poznat korisniku

## Migracijska transparentnost:

- ◆ ne mijenja se simboličko ime, već samo mijenja lokacija poslužitelja (mrežna adresa) u DNS-u

## Relokacijska transparentnost:

- ◆ ne zahtijeva se, poslužitelj je stacionaran i ne kreće se tijekom pružanja usluge

## Replikacijska transparentnost:

- ◆ više poslužitelja

## Konkurencijska transparentnost

- ◆ istodobna uporaba istog resursa od više/mnogo korisnika:

## Transparentnost na kvar

- ◆ više poslužitelja

## Migracijska transparentnost:

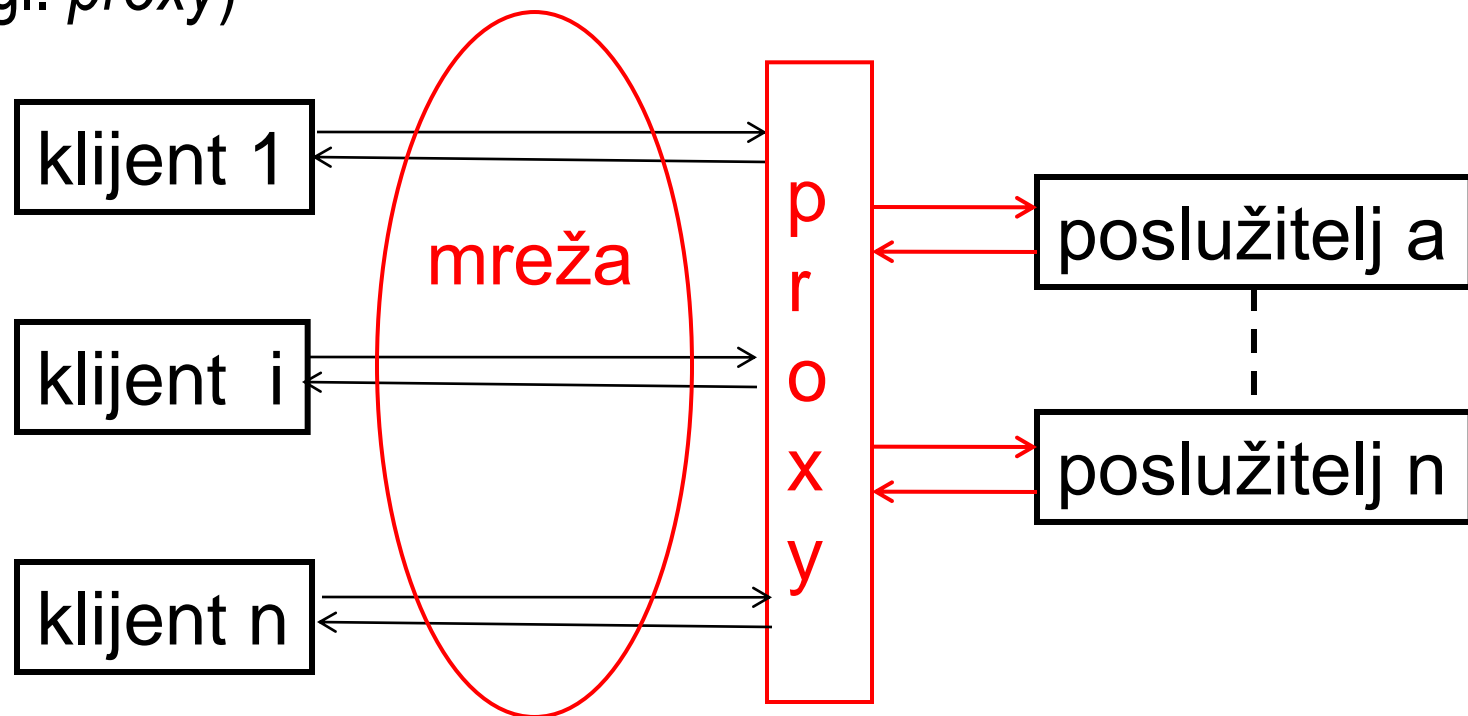
- ◆ omogućen pristup poslužitelju s različitih lokacija (pristupnih točaka)
  - nomadski pristup

## Relokacijska transparentnost:

- ◆ omogućeno kretanje, tj. promjena lokacije (promjena pristupne točke) tijekom pristupa poslužitelju
  - pristup putem pokretne mreže

## Replikacijska transparentnost + transparentnost na kvar:

- više poslužitelja kojima se dostupa putem zastupnika (engl. *proxy*)



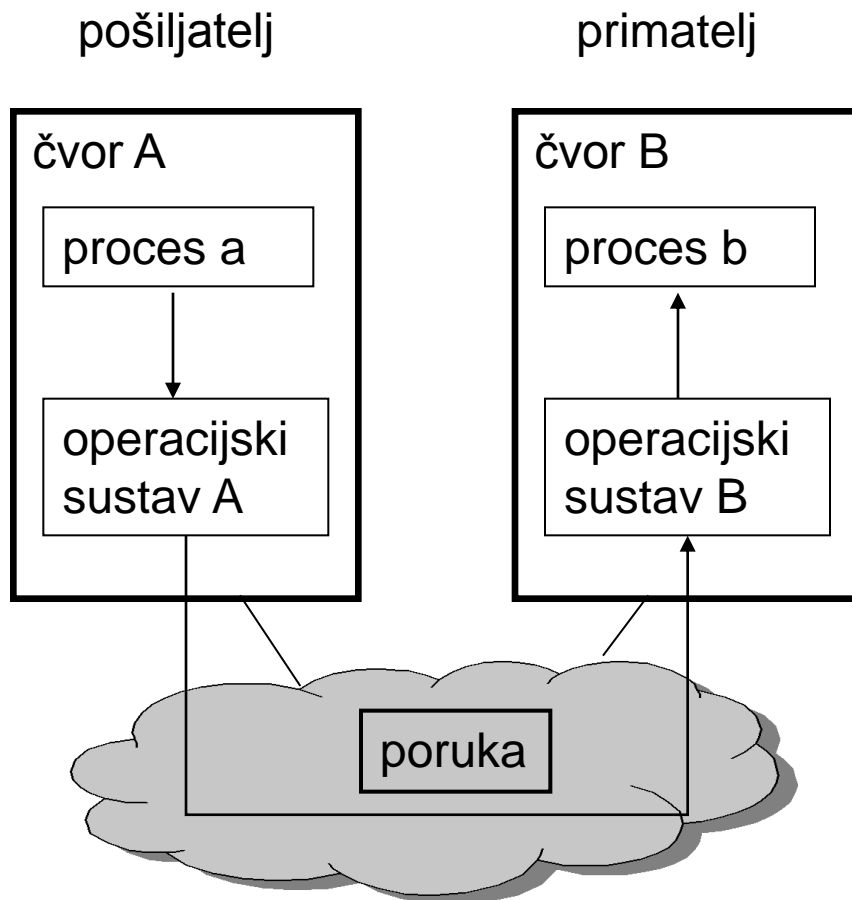
---

# Teorijski modeli raspodijeljenog sustava

Svi modeli raspodijeljenih sustava sastoje se od procesa koji međusobno komuniciraju razmjenom poruka

Temeljni formalizmi:

- ◆ Interakcijski model: procesi, komunikacija, vremenska usklađenost odvijanja i komunikacije procesa
- ◆ Model kvara: kvarovi i njihov utjecaj na odvijanje i komunikaciju procesa
- ◆ Model sigurnosti: prijetnje odvijanju i komunikaciji procesa te mjere zaštite



## Procesi

- izvode se na različitim čvorovima/računalima, autonomni su

## Međuprocena komunikacija

- prosljeđivanje poruka (engl. *message passing*), tj. razmjena poruka na mrežnom sloju
- osigurati vremensku usklađenost (sinkronost) odvijanja i komunikacije procesa

## ◆ Ispad (*failure*)

- stanje sustava koje se detektira kroz nemogućnost korištenja jedne ili više njegovih usluga
- posljedica neispravnosti, signalizira postojanje neispravnosti u raspodijeljenom sustavu
  - Ispad procesa: proces ne mijenja stanja (ne izvode se prijelazi) premda se ne nalazi u završnom stanju (*stopping failure*) ili proces generira proizvoljne izlaze (*Byzantine failure*)
  - Ispad kanala: proces  $p$  je poslao poruku procesu  $q$ , ali  $q$  poruku ne prima

## ◆ Neispravnost (*fault*)

- pogreške prilikom oblikovanja sustava, neispravan dio programskog koda (*bug*), ili komunikacijski kanal,
- prolazne, isprekidane i trajne neispravnosti

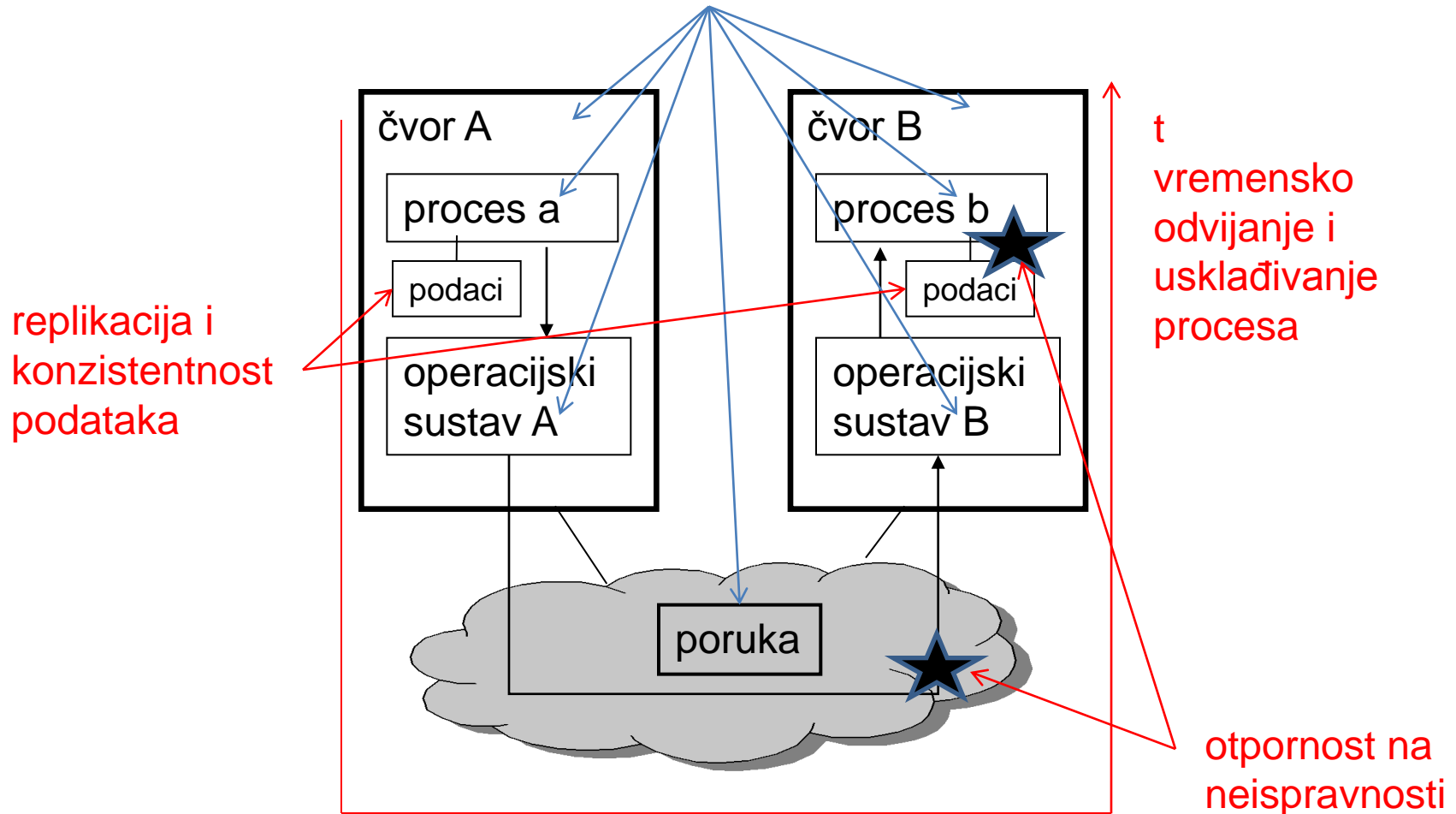
## Prijetnje odvijanju i komunikaciji procesa:

- ◆ presretanje/prisluškivanje
- ◆ prekidanje ili uskraćivanje usluge
- ◆ promjena, fabrikacija ili ponavljanje
- ◆ lažno predstavljanje, maskiranje, utjelovljivanje

## Mjere zaštite:

- ◆ provjera autentičnosti i kontrola pristupa
- ◆ očuvanje cjelovitosti (integriteta) i po potrebi povjerljivost
- ◆ osiguranje neporecivosti
- ◆ održavanje raspoloživost i radne sigurnosti

## Imenovanje



---

# Vremensko odvijanje i usklađivanje procesa

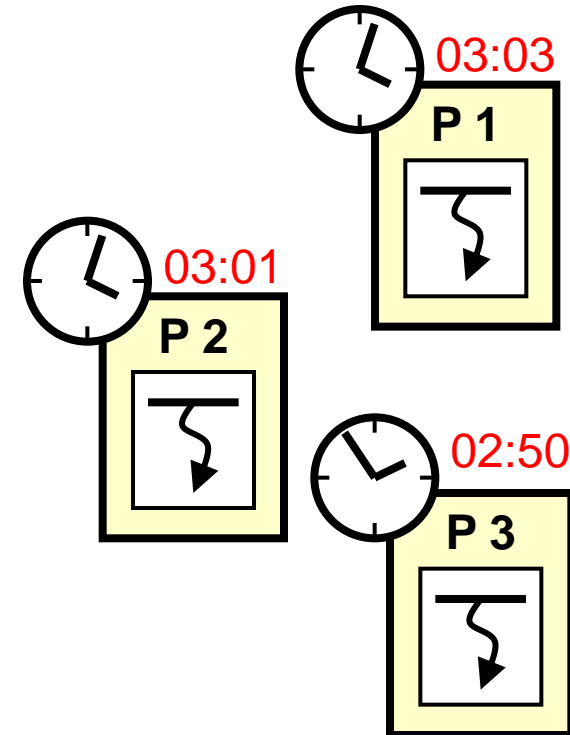
(prema: Raspodijeljeni sustavi, predavanja ak.g. 2009./10.)

- ◆ **Svako računalo ima vlastiti satni mehanizam**

- ◆ Satovi nisu usklađeni
- ◆ Satovi imaju različiti takt
- ◆ Satovi imaju različita odstupanja

- ◆ **Usuglašavanje vremena**

- ◆ Fizički sat
- ◆ Logički sat



## ◆ Algoritam Cristian

- ◆ Primjena poslužitelja s točnim vremenom
- ◆ Dohvaćanje informacije o vremenu prema potrebi

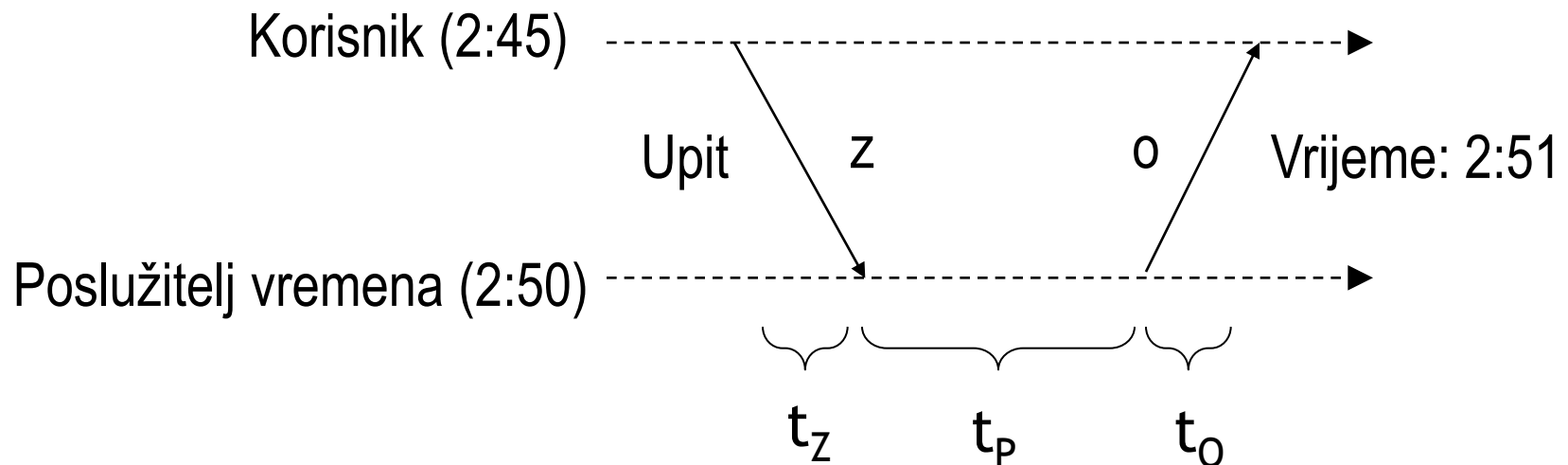
## ◆ Algoritam Berkeley

- ◆ Primjena upravitelja vremena
- ◆ Periodičko odašiljanje informacije o vremenu

## ◆ Primjena poslužitelja vremena

## ◆ Koraci algoritma

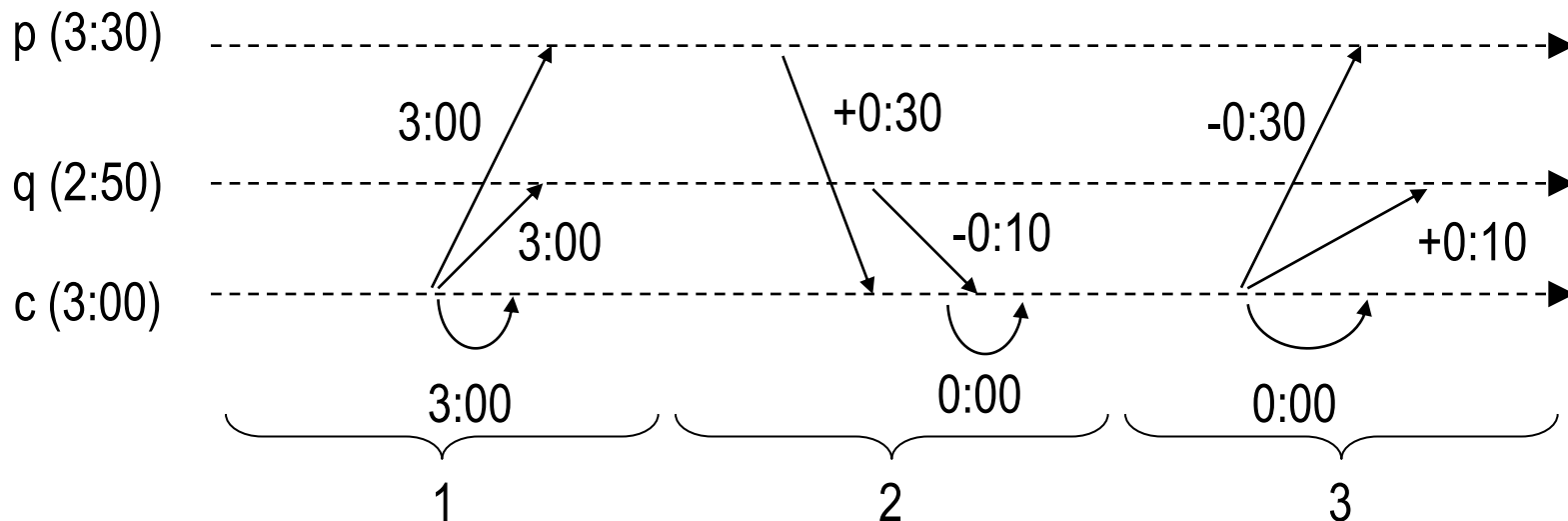
- 1) Korisnički proces upućuje zahtjev za dohvat vremena (z)
- 2) Proces poslužitelj šalje trenutno vrijeme (o)



## ◆ Primjena upravitelja vremena

## ◆ Koraci algoritma

- 1) Upravljački proces  $c$  šalje vrijeme procesima  $p$ ,  $q$ ,  $c$
- 2) Procesi  $p$ ,  $q$ ,  $c$  šalju razliku vremena upravljačkom procesu  $c$
- 3) Upravljački proces  $c$  šalje pomak procesima  $p$ ,  $q$ ,  $c$



- ◆ **Skup fizičkih satnih mehanizama**
  - ◆ Satni mehanizmi su potpuno nezavisni
  
- ◆ **Usklađivanje globalnog tijeka vremena**
  - ◆ Primjena logičkih oznaka vremena
  
- ◆ **Vrste logičkih oznaka**
  - ◆ Skalarnе oznake vremena
  - ◆ Vektorske oznake vremena

## ◆ Globalno logičko vrijeme

- ◆ Sva računala na jednak način bilježe tijekom globalnog logičkog vremena

## ◆ Oznake logičkog vremena

- ◆ Svakoj akciji  $a$  koju provode procesi u raspodijeljenoj okolini pridružena je jedinstvena oznaka vremena  $T(a)$
- ◆ Ako za događaj  $a$  i  $b$  vrijedi uzročna relacija  $a \rightarrow b$  tada vrijedi da je akcija  $a$  ostvarena u vremenu prije akcije  $b$  [  $T(a) < T(b)$  ]

## ◆ Primjena

- ◆ Uređena razmjena poruka
- ◆ Svi procesi na isti način vide redoslijed događaja

# Skalarne oznake vremena (2)

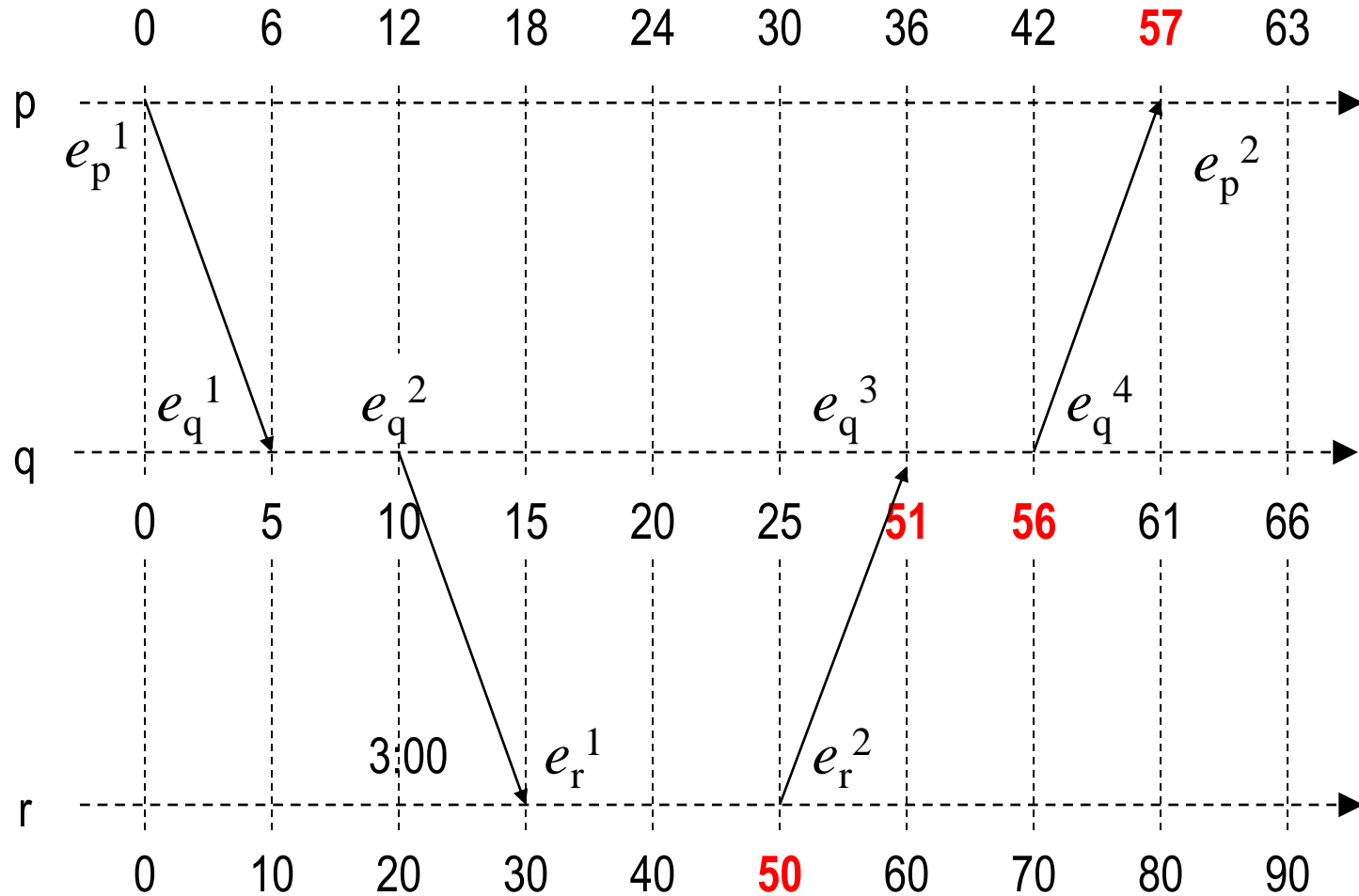


$$e_p^1 \rightarrow e_q^1 \quad e_q^2 \rightarrow e_r^1$$

$$e_r^2 \rightarrow e_q^3 \quad e_q^4 \rightarrow e_p^2$$

$$T(e_r^2) < T(e_q^3)$$

$$T(e_q^4) < T(e_p^2)$$



## ◆ Prednosti primjene skalarnih oznaka

- ◆ Tijek vremena zasnovan je na jednostavnom modelu
- ◆ Svi procesi usklađeni su s globalnim tijekom vremena
- ◆ Usuglašeni su vremenski trenutci nastupanja akcija u raspodijeljenom okružju

## ◆ Nedostatci primjene skalarnih oznaka

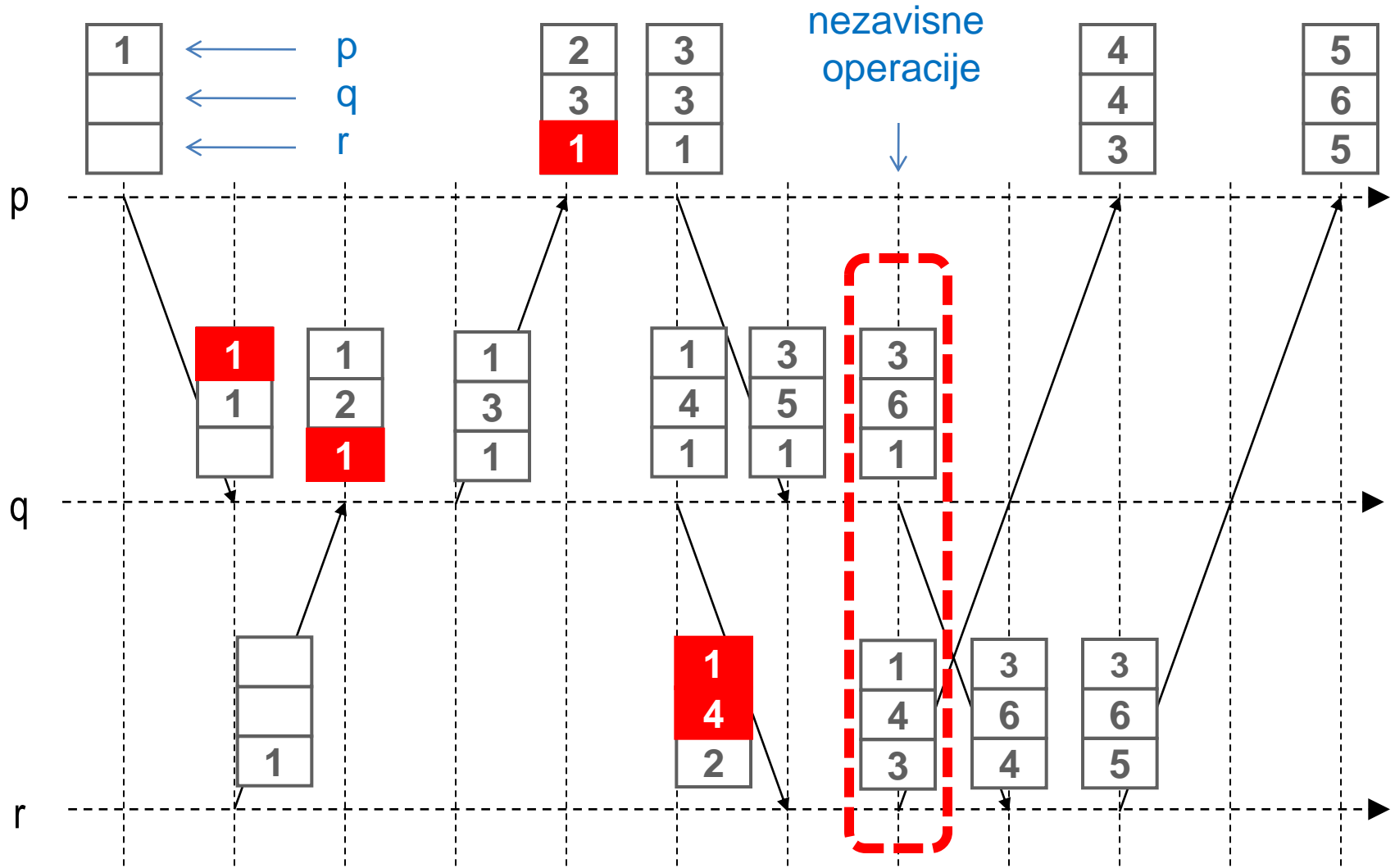
- ◆ Ako za događaje  $a$  i  $b$  vrijedi da je vremenska oznaka od  $a$  manja od vremenske oznake od  $b$ , to ne povlači nužno da je događaj  $a$  nastupio u vremenu prije događaja  $b$
- ◆  $T(a) < T(b)$  ne povlači  $a \rightarrow b$

- ◆ **Vektorska oznaka opisuje uzročno-posljedične veze između događaja u vremenu**
  - ◆ Polje elemenata  $V[M]$  koje opisuje broj akcija provedenih na skupu od  $N$  računala u raspodijeljenoj okolini
  - ◆ Računala razmjenjuju vektorske oznake tijekom razmjene poruka
  
- ◆ **Vektorska oznaka**
  - ◆  $V_p[i]$  sadrži broj akcija koje je ostvario proces  $P_p$
  - ◆  $V_p[m]$  sadrži broj akcija za koje proces  $P_p$  zna da su ostvarene od strane procesa  $P_m$
  - ◆ Ako za događaje  $a$  i  $b$  vrijedi  $V(a) < V(b)$  tada vrijedi da je događaj  $a$  nastupio u vremenu prije događaja  $b$ ,  $a \rightarrow b$

## ◆ Koraci algoritma za održavanje vektorskih oznaka:

- 1) Početna vrijednosti svih komponenata je 0
- 2) Za svaki interni događaj procesa  $p$  uvećaj oznaku  $V_p[p]$
- 3) Prije slanja poruke na procesu  $p$  uvećaj oznaku  $V_p[p]$  i poslanoj poruci pridruži izgrađeni vektor  $V_p$
- 4) Nakon primitka poruke od procesa  $p$  na procesu  $k$  uvećaj oznaku  $V_k[k]$  dok za sve ostale oznake  $j \neq k$  vrijedi:  $V_k[j] = V_p[j]$  ako je  $V_k[j] < V_p[j]$

# Vektorske oznake vremena (3)



## ◆ Semafor u raspodijeljenom okružju

- ◆ Semafor je proces koji u spremniku čuva  $n$  znački
- ◆ Ako u spremniku postoji traženi broj znački  $N$  ( $n \geq N$ ), zahtjev se prihvaća, inače se stavlja u rep

## ◆ Sinkronizacija zasnovana na razmjeni događaja (objavi/pretplati)

- ◆ Posrednik sadrži spremnik pretplata i spremnik s objavljenim događajima
- ◆ Procesi se pretplaćuju kod posrednika
- ◆ Procesi šalju poruke s događajima
- ◆ Ako posrednik ima pretplatu za objavljeni događaj, prosljeđuje događaj procesu pretplatniku

## Međusobno isključivanje:

- ◆ primjenom raspodijeljenog repa čekanja
- ◆ primjenom prstena

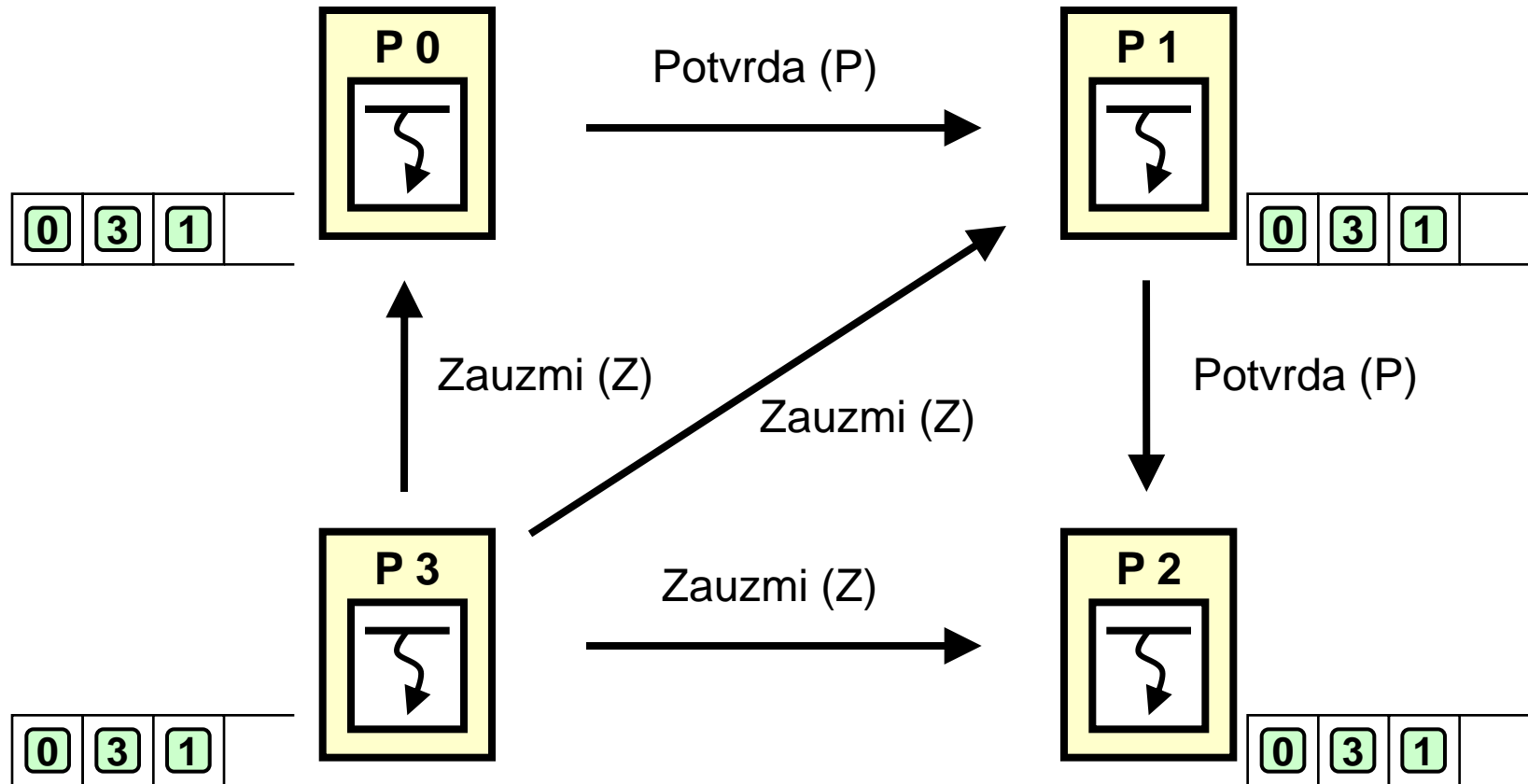
## ◆ Raspodijeljeni rep čekanja

- ◆ Svaki proces ima lokalni rep čekanja
- ◆ Procesi razmjenjuju informacije potrebne za usklađivanje stanja svih repova čekanja u sustavu

## ◆ Pretpostavke

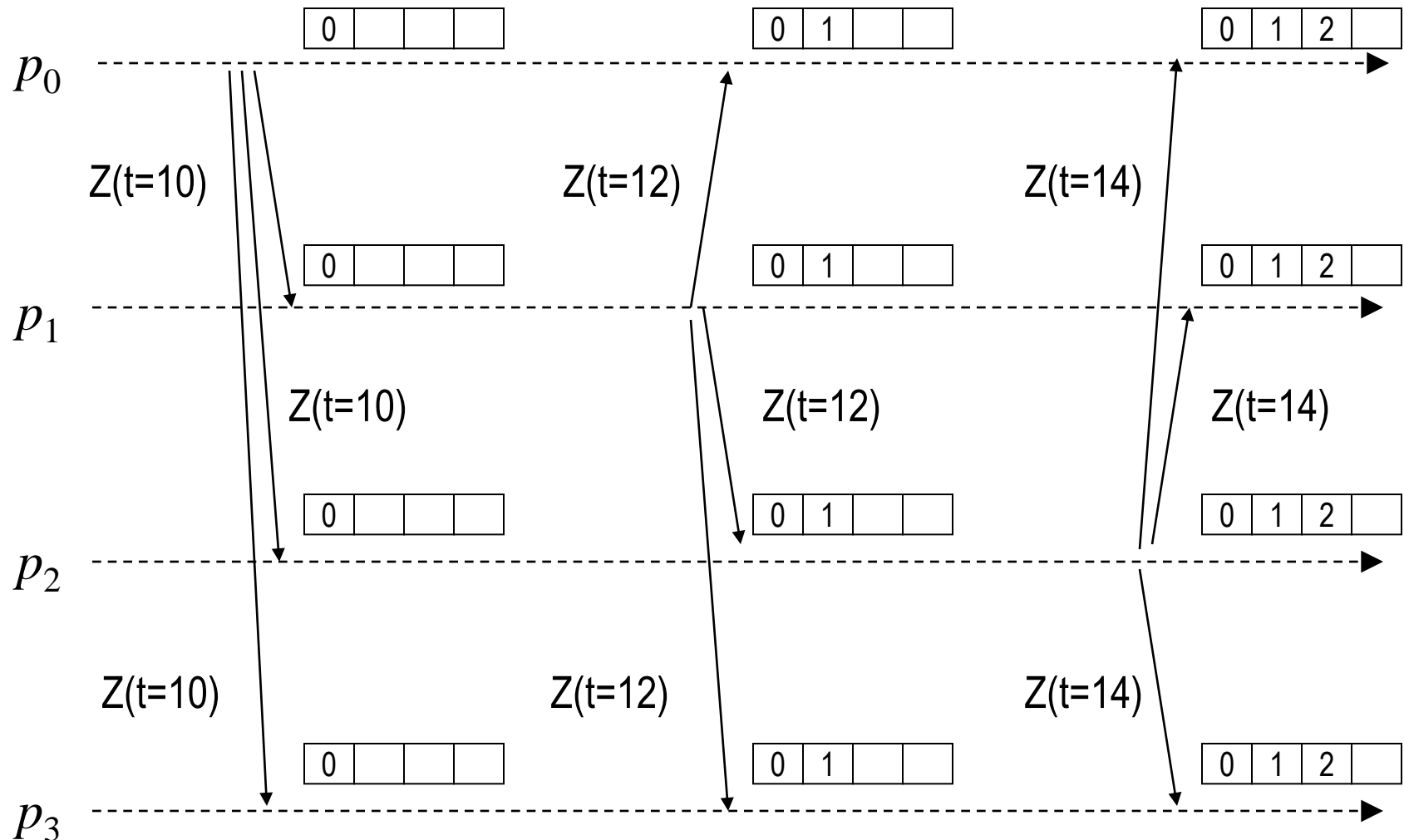
- ◆ Svaki proces ima lokalni satni mehanizam koji je usklađen s ostalim procesima
- ◆ Svaki zahtjev za pristup sredstvu uključuje oznaku trenutka u kojem je proces uputilo zahtjev
- ◆ Procesi ostvaruju pristup u skladu s vremenskim oznakama upućivanja zahtjeva

# Raspodijeljeni rep (2)



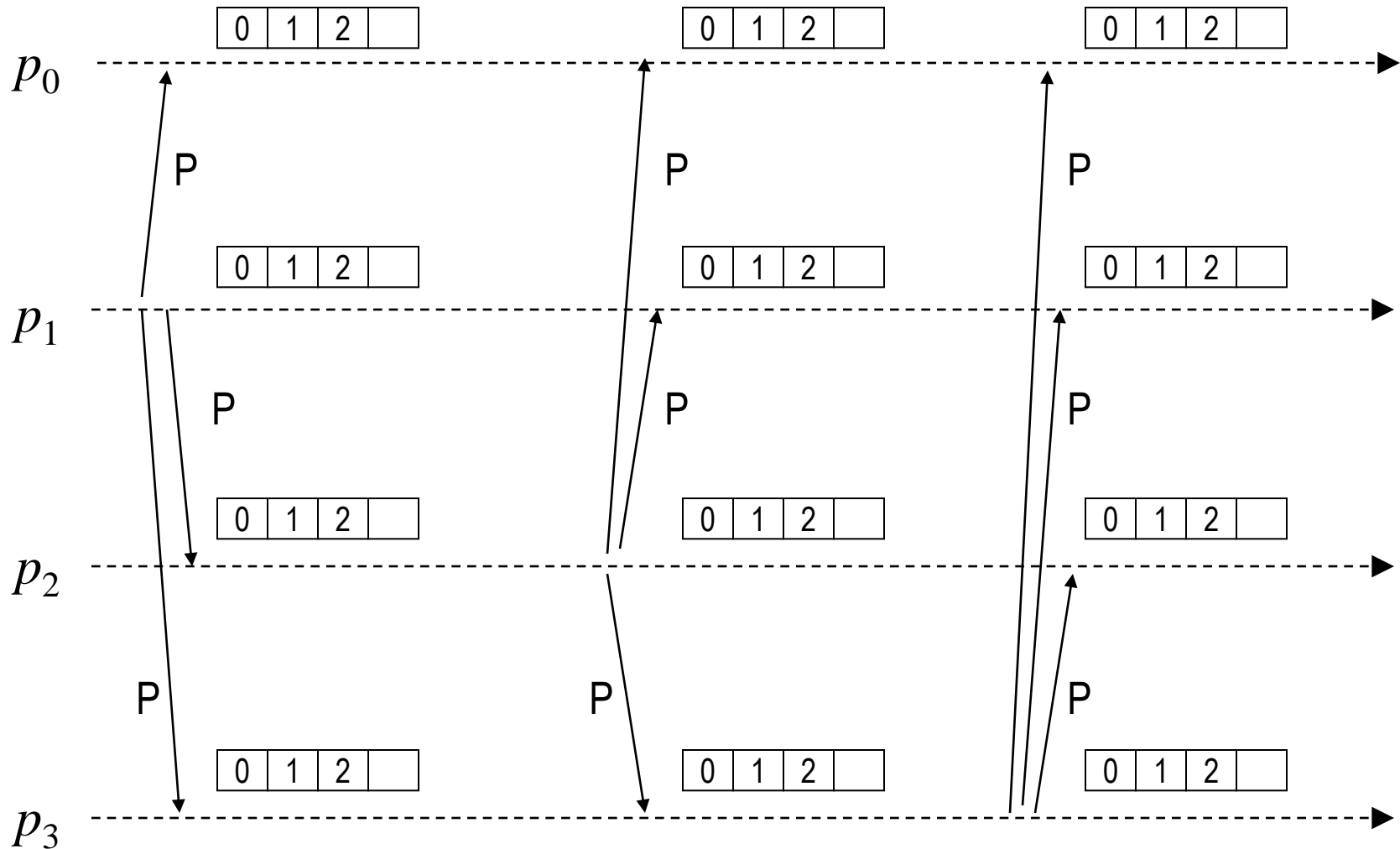
# Raspodijeljeni rep (3)

$Z(t=x)$  – Zauzmi, vremenska oznaka  $x$



# Raspodijeljeni rep (4)

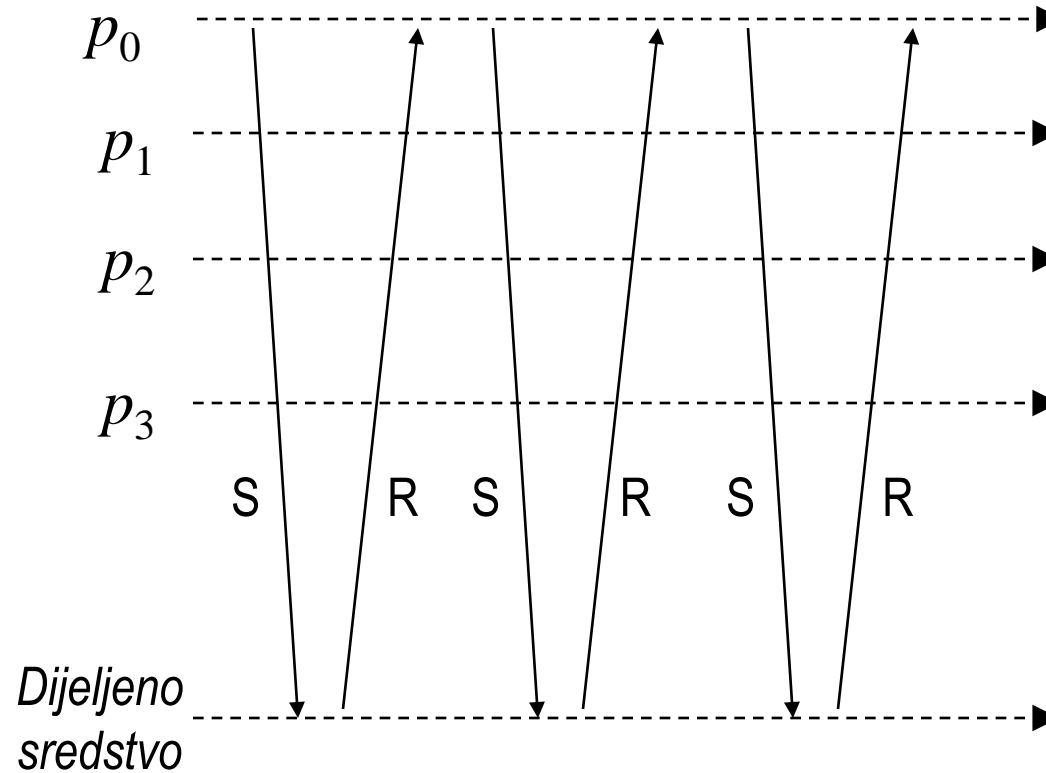
P – Potvrda



# Raspodijeljeni rep (5)

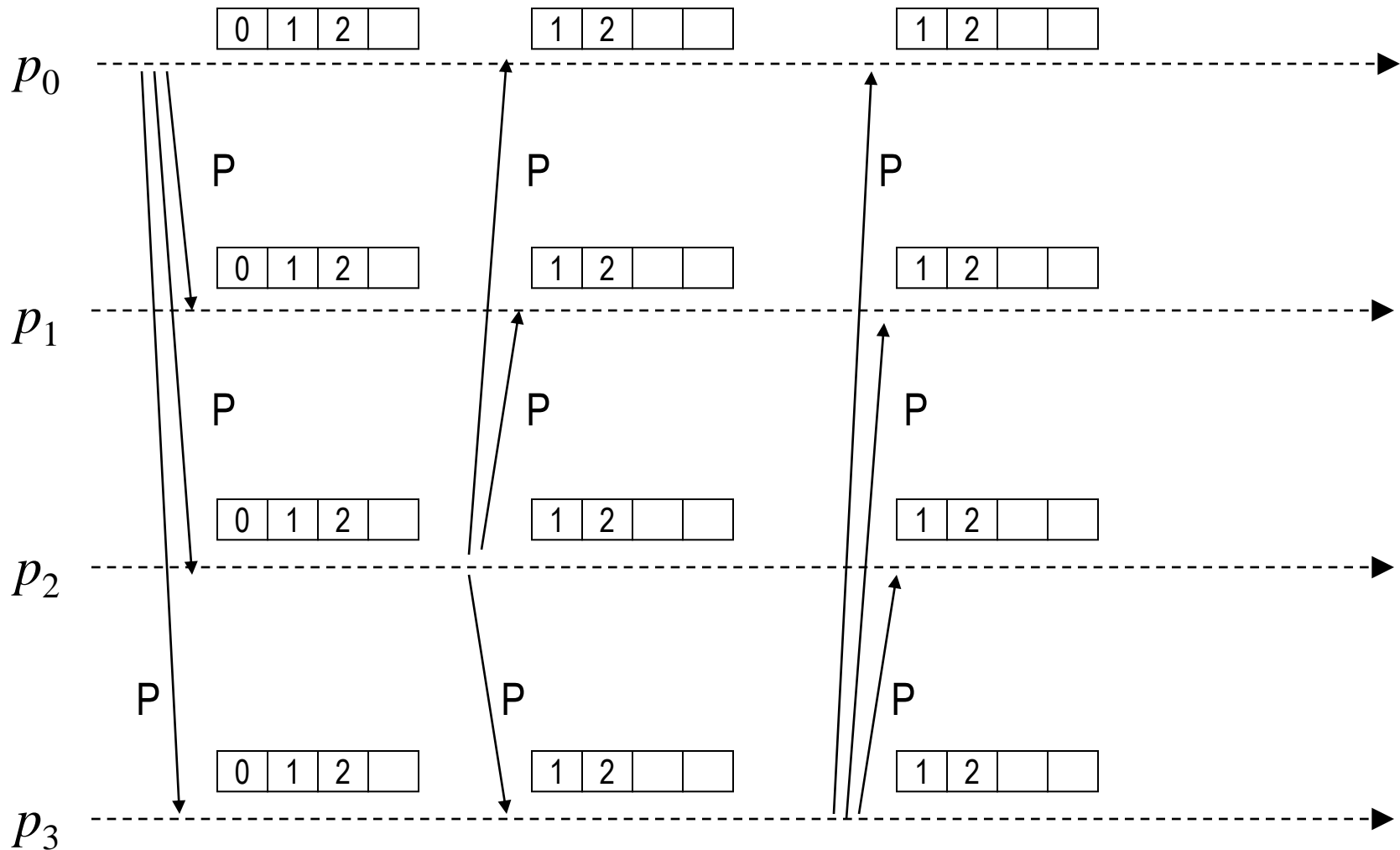


R – Dohvati, S – Spremi

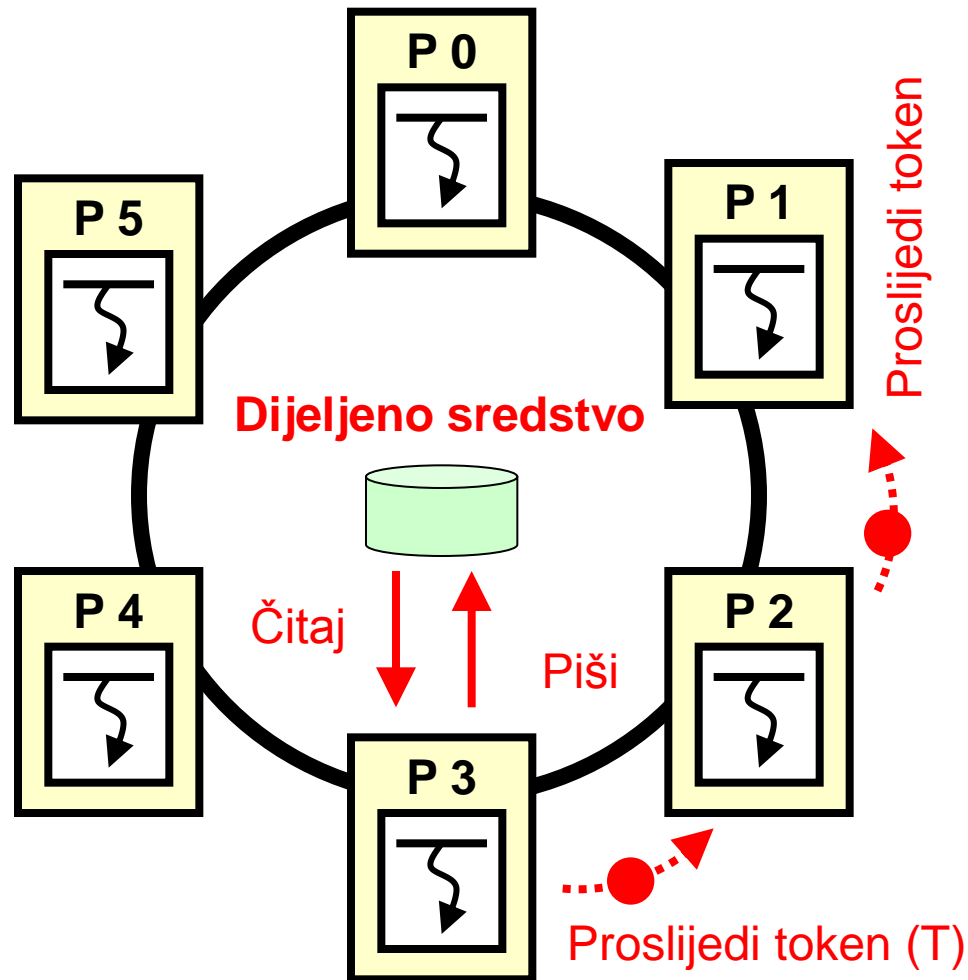


# Raspodijeljeni rep (6)

P – Potvrda



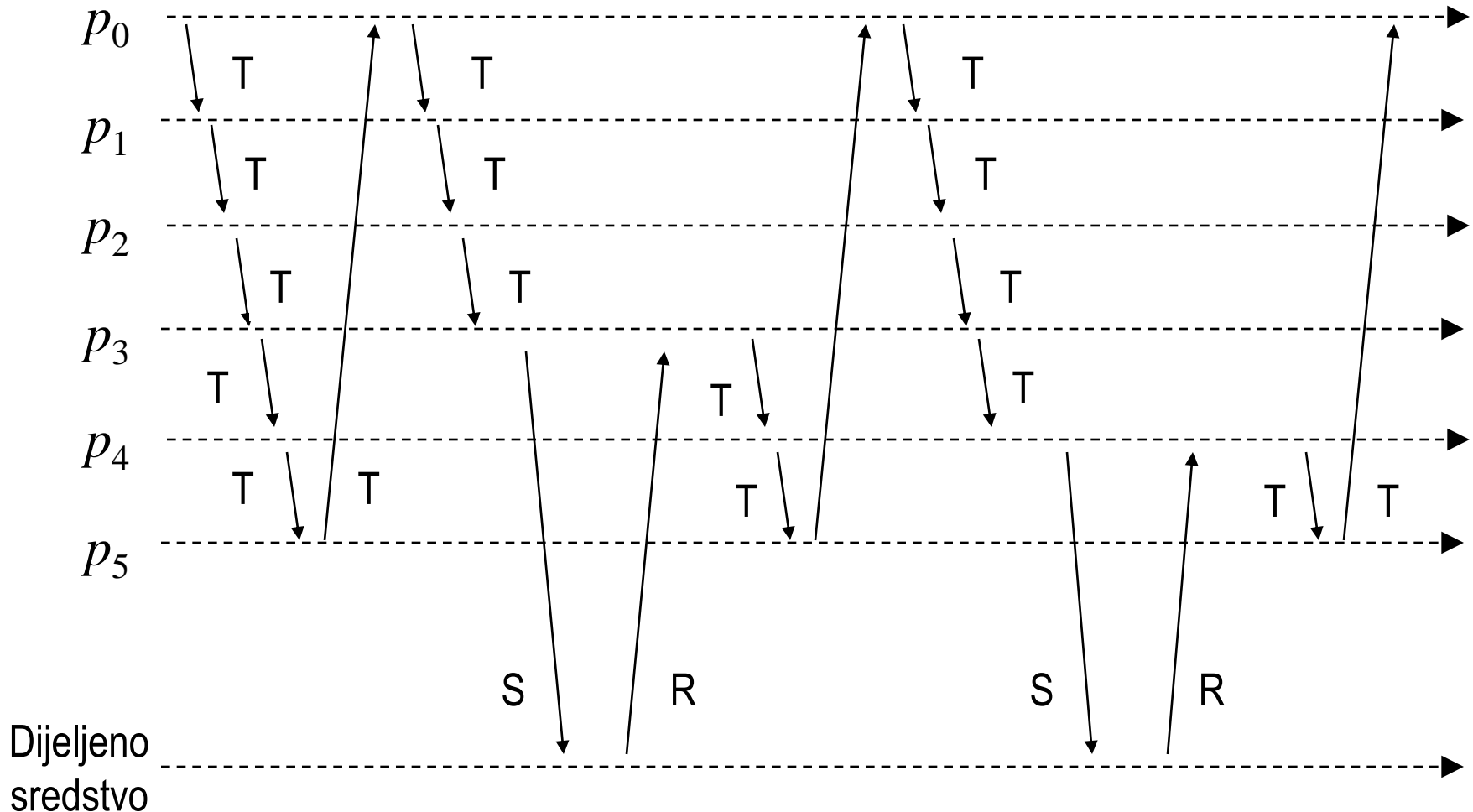
# Međusobno isključivanje primjenom prstena (1)



# Međusobno isključivanje primjenom prstena (2)



T – Prijenos tokena, S – Spremi, R – Dohvati

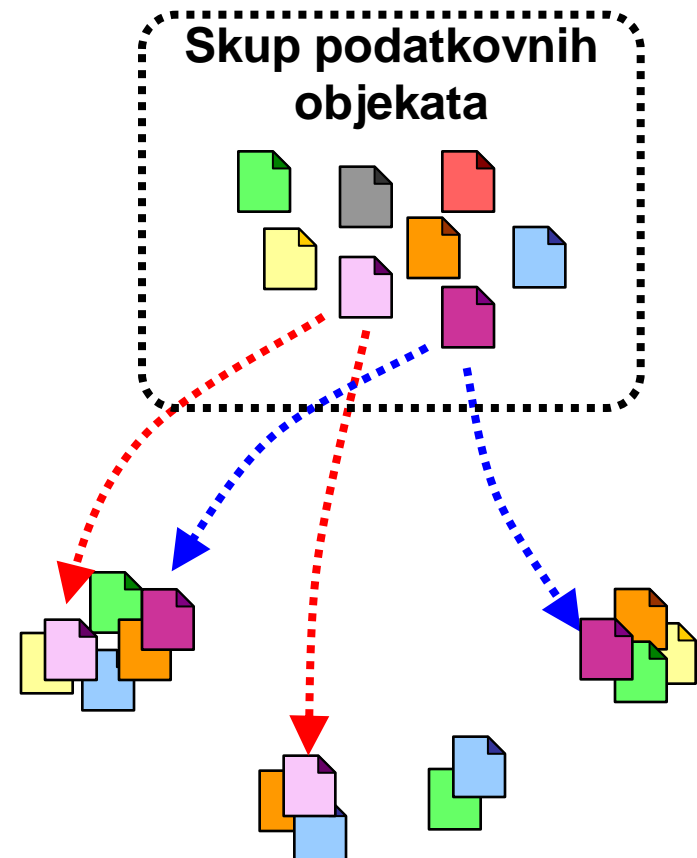


---

# Replikacija i konzistentnost podataka

(prema: Raspodijeljeni sustavi, predavanja ak.g. 2009./10.)

- ◆ Podatkovni objekti postavljeni na skupu čvorova/računala
- ◆ Podatkovni objekti su spremljeni u obliku nekoliko kopija (**replika**) na različitim čvorovima/računalima
- ◆ Replikacija je postupak stvaranja i upravljanja kopijama podatkovnih objekata



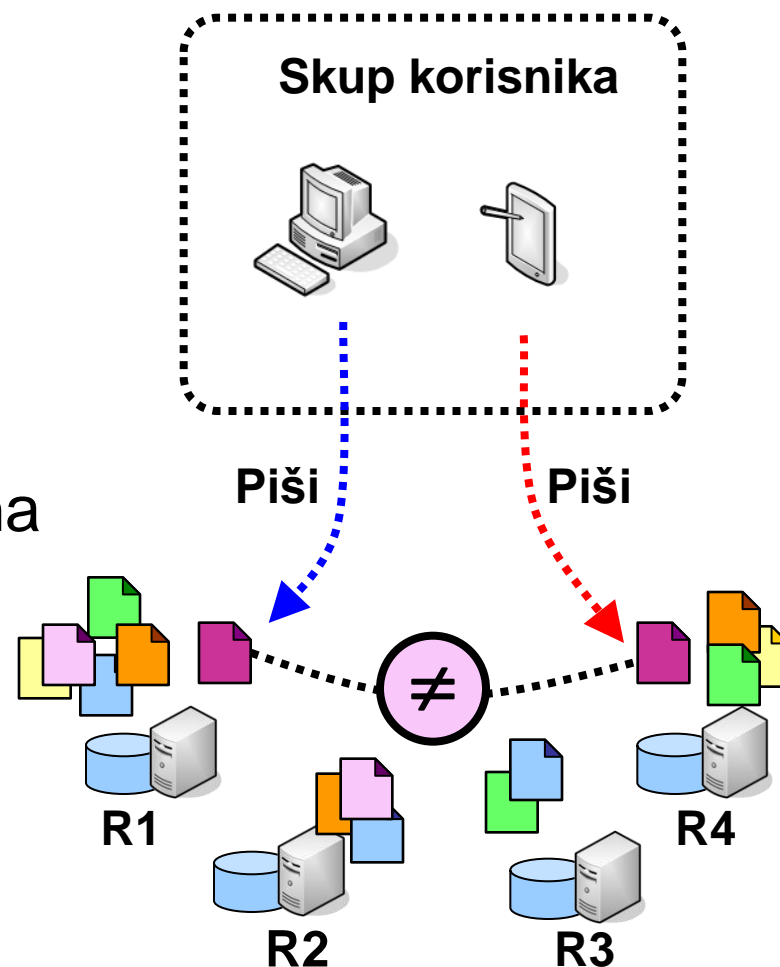
## ◆ Pouzdanost

- ◆ U slučaju da neka od replika postane nedostupna, sustav prosljeđuje zahtjeve preostalim dostupnim replikama
- ◆ U slučaju da neka od replika ima pogrešne zapise, usporedbom zapisa više replika ostvaruje se otpornost na pogreške

## ◆ Učinkovitost

- ◆ U slučaju da je neka od replika preopterećena, pristigli zahtjevi prosljeđuju se ostalim replikama
- ◆ Replike je moguće predodrediti za posluživanje različitih razreda zahtjeva

- ◆ Replike istog objekta mogu istodobno i nezavisno koristiti različiti korisnici
- ◆ Korisnici mogu stanje replika istog objekta mijenjati na različite načine tijekom vremena
- ◆ Konzistentnost je **narušena** kada postoje replike nekog objekta koje nemaju istovijetno stanje



# Modeli konzistentnosti (1)

Složenost ostvarivanja modela u raspodijeljenoj okolini

Složenost primjene modela tijekom razvoja sustava

teško napraviti

Složenost

teško koristiti

Specijalizirane operacije

Vrsta modela konzistentnosti

Stroga

Slijedna

Povezana

Prema redoslijedu prispjeca

Slaba

Otpuštanje

Zauzimanje

## ◆ Stroga konzistentnost

- Čitanje podatka na lokaciji  $X$  daje vrijednost koja je posljednja bila zapisana na lokaciju  $X$

## ◆ Slijedna konzistentnost

- Redoslijed izvođenja operacija može biti **proizvoljan**, ali svi procesi moraju konačni **slijed** izvođenja akcija u vremenu vidjeti **jednako**

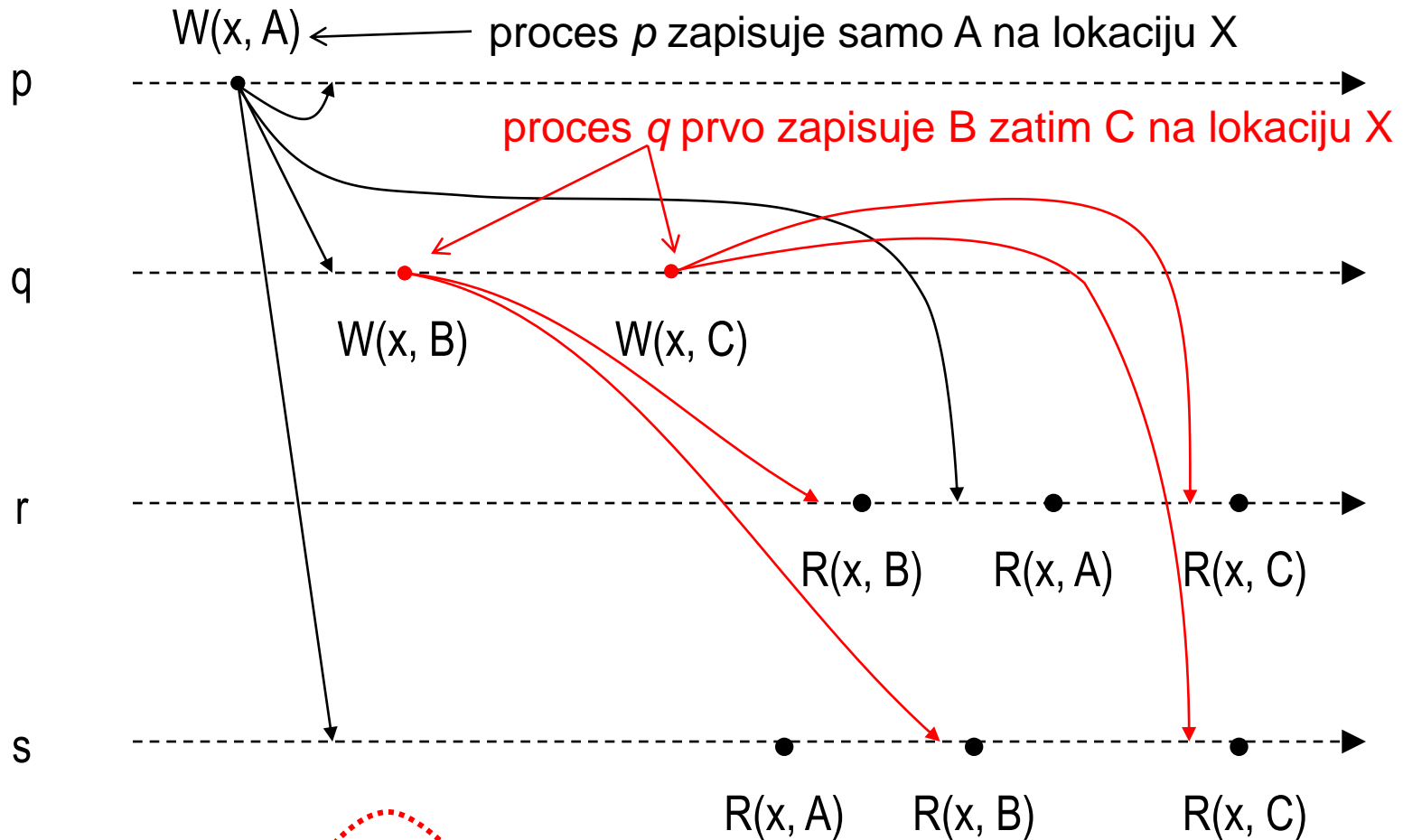
## ◆ Povezana konzistentnost

- Redoslijed izvođenja **povezanih** operacija pisanja vidljiv je svim procesima na jednak način, dok redoslijed izvođenja operacija pisanja koje **nisu povezane** svakom procesu može biti prikazan na drugačiji način

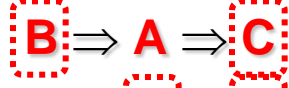
## ◆ Konzistentnost redoslijeda

- Redoslijed izvođenja operacija pisanja provedenih od strane **jednog** procesa vidljiv je na **jednak** način svim ostalim procesima, ali redoslijed izvođenja operacija pisanja **različitih** procesa može biti vidljiv na **proizvoljan** način ostalim procesima
- Ostvarenje zasnovano na pridruživanju jedinstvenih oznaka svakom zahtjevu za pisanje
- Jedinstvena oznaka uključuje identifikator procesa i redni broj izvođenja operacije: vektorska oznaka vremena

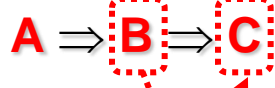
# Modeli konzistentnosti (4)



Čitanje r:



Čitanje s:



konzistentnost redoslijeda: uvijek B pa C

## ◆ Slaba konzistentnost

- Ostvaruje se primjenom sinkronizacijskih varijabli koje ostvaruju upravljanje trenutcima sinkronizacije replika u raspodijeljenoj okolini

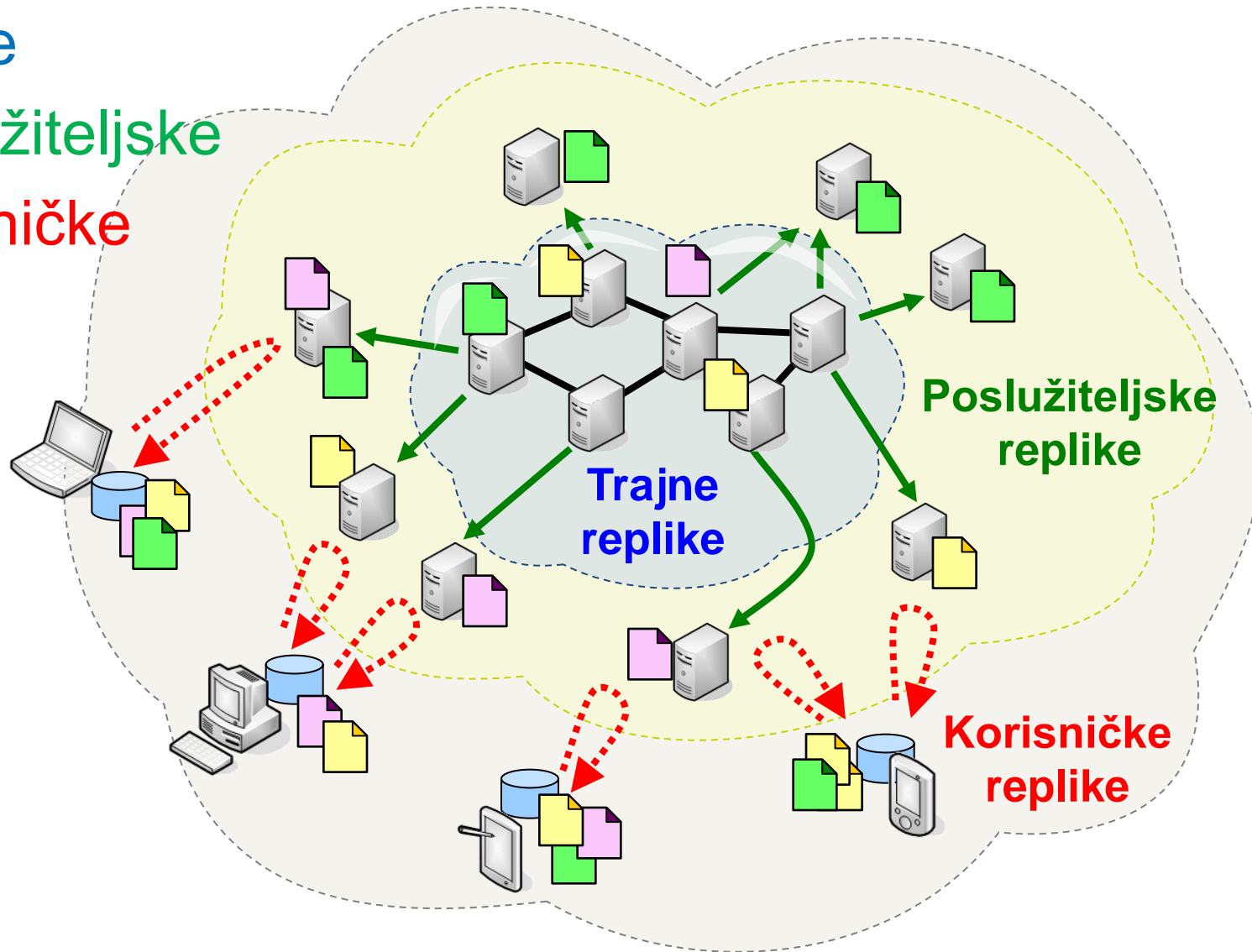
## ◆ Konzistentnost otpuštanja

- zasnovana na primjeni kritičnih odsječaka, održava se nakon izlaska iz kritičnog odsječaka

## ◆ Konzistentnost zauzimanja

- zasnovana na primjeni kritičnih odsječaka, održava se u trenutku ulaska u kritični odsječak

Trajne  
Poslužiteljske  
Korisničke



- ◆ Početni skup replika postavljen na skupu računala povezanih lokalnom mrežom
  - **Statička organizacija i postavke sustava**
  - Većina zahtjeva je čitanje podataka
  - Raspoređivanje zahtjeva na dostupne replike

## Poslužiteljske replike

- ◆ Poslužiteljske replike nastaju repliciranjem trajnih replika
- ◆ Poslužiteljske replike dostupne su korisnicima
- ◆ Stvaranje i raspoređivanje poslužiteljskih replika tijekom rada sustava, sukladno opterećenju sustava

- ◆ **Korisnički programi koriste lokalni spremnik replike**
  - Lokalni spremnik može biti na istom računalu kao i korisnički programi ili na dijeljenom računalu u lokalnoj mreži
  - Dohvaćeni podaci spremaju se u lokalni spremnik
  - U slučaju potrebe za istim podacima, podaci se dohvaćaju iz lokalnog spremnika i time smanjuje vrijeme dohvata podataka
  - Najpovoljnije je koristiti u slučajevima kada se najčešće provode operacije čitanja
  - Potrebno je održavati konzistentnost lokalnog spremnika s poslužiteljem s kojeg su podaci dohvaćeni
  - U slučaju kada nekoliko korisnika dijeli lokalni spremnik povećava se učinkovitost primjene korisničkih replika

## Obnavljanje stanja replika

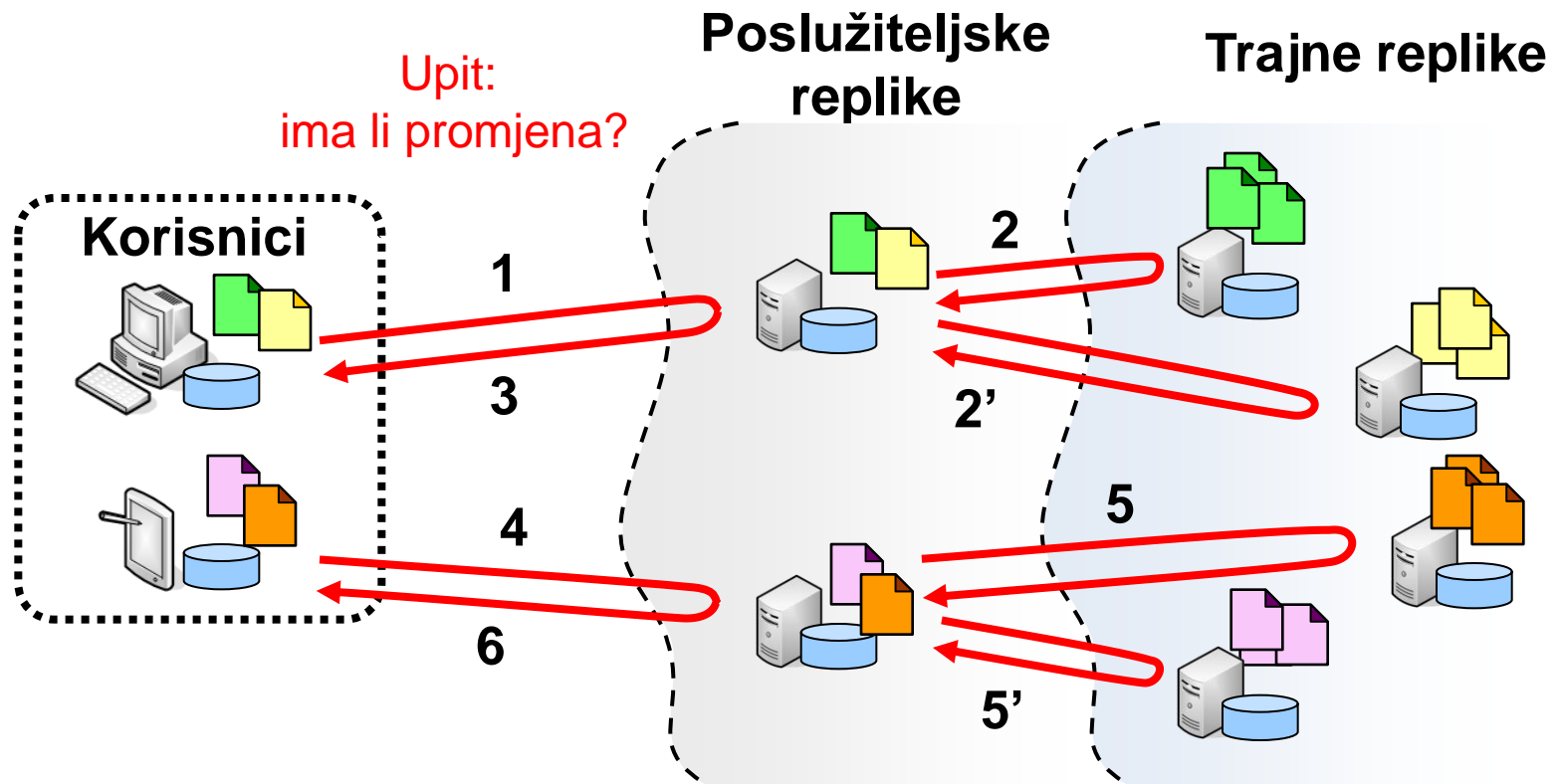
- ◆ Replike je potrebno usklađivati s promjenama stanja trajnih replika
- ◆ Obnavljanje sadržaja replika može biti ostvareno u trenutku promjene sadržaja ili trenutak prije ostvarivanja pristupa replici

## Osnovne metode održavanja konzistentnosti sadržaja replika

- ◆ Dohvaćanje promjena sadržaja (*pull*)
- ◆ Prosljeđivanje promjena sadržaja (*push*)

# Dohvaćanje promjena sadržaja (1)

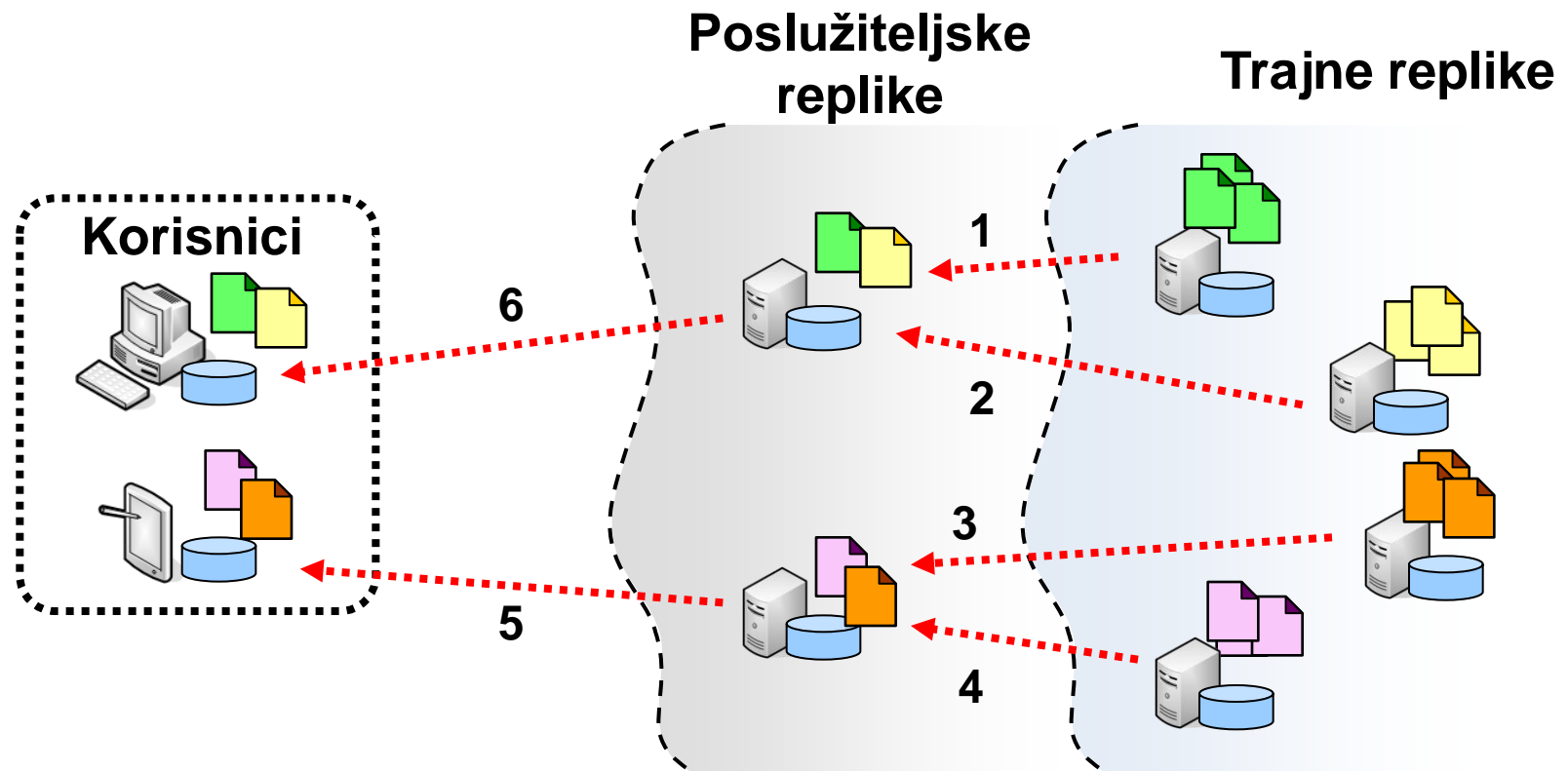
- ◆ Korisnici dohvaćaju promjene sadržaja trenutak prije pristupa replikama
- ◆ Primjenjuju se od strane korisnika prema poslužiteljskim replikama i poslužiteljima trajnih replika



- ◆ Pogodno za korištenje u slučajevima rijetkih izmjena sadržaja replika
- ◆ Poslužitelji trajnih replika ne moraju znati broj i identitet korisnika
- ◆ Smanjuje se mrežno opterećenje i rasterećuje poslužitelj replika
- ◆ U slučajevima da lokalno stanje replike nije obnovljeno povećava se vrijeme dohvata novog stanja replika

# Prosljeđivanje promjena sadržaja (1)

- ◆ Poslužitelji sadržaja prosljeđuju svim replikama promjene stanja sadržaja u trenutku nastanka promjene
- ◆ Primjenjuju se od strane poslužitelja trajnih replika prema korisnicima



- ◆ Ostvarivanje velikog stupnja konzistentnosti
- ◆ Poslužitelji trajnih replika moraju imati zabilježene adrese svih replika koje obnavljaju i opis njihova stanja
- ◆ U slučaju da neka od replika ukloni dio svojeg stanja, replika obavještava o promjenama stanja poslužitelja trajnih replika
- ◆ Najčešće se koristi u slučajevima kad veliki broj korisnika koristi dijeljenu repliku kada je vjerojatno da će uvijek neki od klijenata imati potrebu koristiti obnovljene podatke

---

# Otpornost na neispravnosti

(prema: Raspodijeljeni sustavi, predavanja ak.g. 2009./10.)

- ◆ Zamjena jednog procesa grupom identičnih procesa
  - replikacija procesa radi prikrivanja ispada jednog ili  $k$  procesa
- ◆ **tolerancija  $k$  ispada**
  - grupa može “preživjeti” ispad najviše  $k$  procesa
- ◆ dovoljan je  $k + 1$  proces da se osigura tolerancija na  $k$  ispada (jedan proces može preuzeti posao)
- ◆ potrebno je  $2k + 1$  procesa da se osigura tolerancija  $k$  bizantskih ispada, ( $k + 1$  ispravni proces će “nadglasati”  $k$  neispravnih)

## Jamstvo isporuke poruke svim procesima u grupi

### ◆ problemi

- Koji procesi čine grupu u trenutku slanja poruke?
- Što se događa ako novi proces ulazi u grupu procesa dok je isporuka poruke grupi u tijeku?
- Što se događa ako dođe do ispada pošiljatelja poruke tijekom isporuke poruke ostalim procesima?
- Što se događa ako jedan od primatelja ispadne tijekom isporuke poruke?

U slučaju ispada, jamstvo isporuke poruke svim ispravnim i dostupnim procesima u grupi ili niti jednom

### ◆ ujedno osigurati isporuku poruka u određenom slijedu

## Oporavak nakon ispada procesa i povratak u ispravno stanje

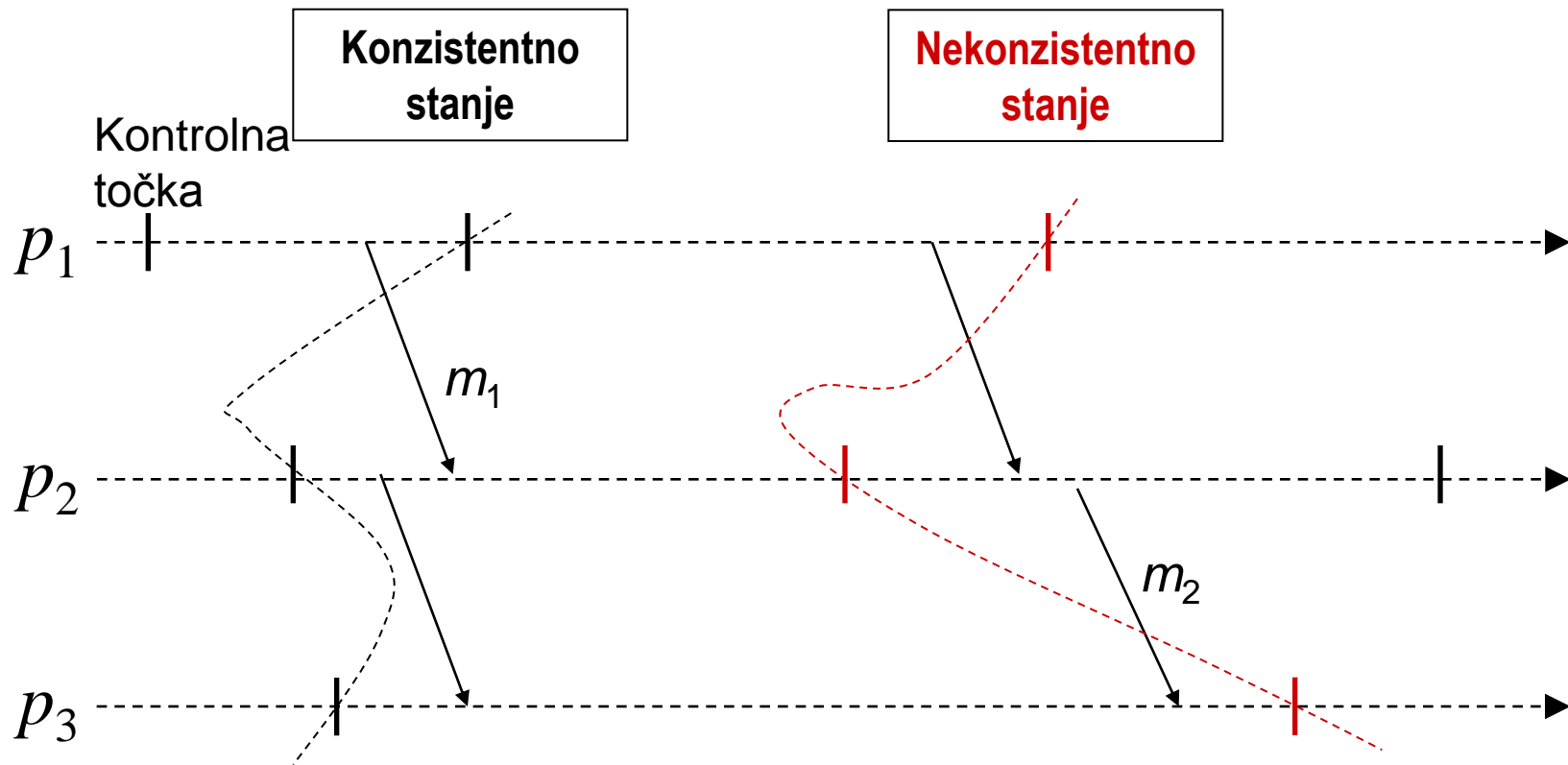
### ◆ Oporavak unazad

- vratiti sustav u ispravno stanje u prošlosti
- potrebno je s vremena na vrijeme pohraniti stanje sustava (kontrolne točke, *checkpoint*)
- procesi bilježe kontrolne točke neovisno jedan o drugom
- zapisuju se poslane i primljene poruke u dnevnički zapis (*log*)

### ◆ Oporavak korištenjem dnevničkog zapisa

- proces u ispadu vraća se u prethodno ispravno stanje nakon čega izvodi akcije iz dnevničkog zapisa

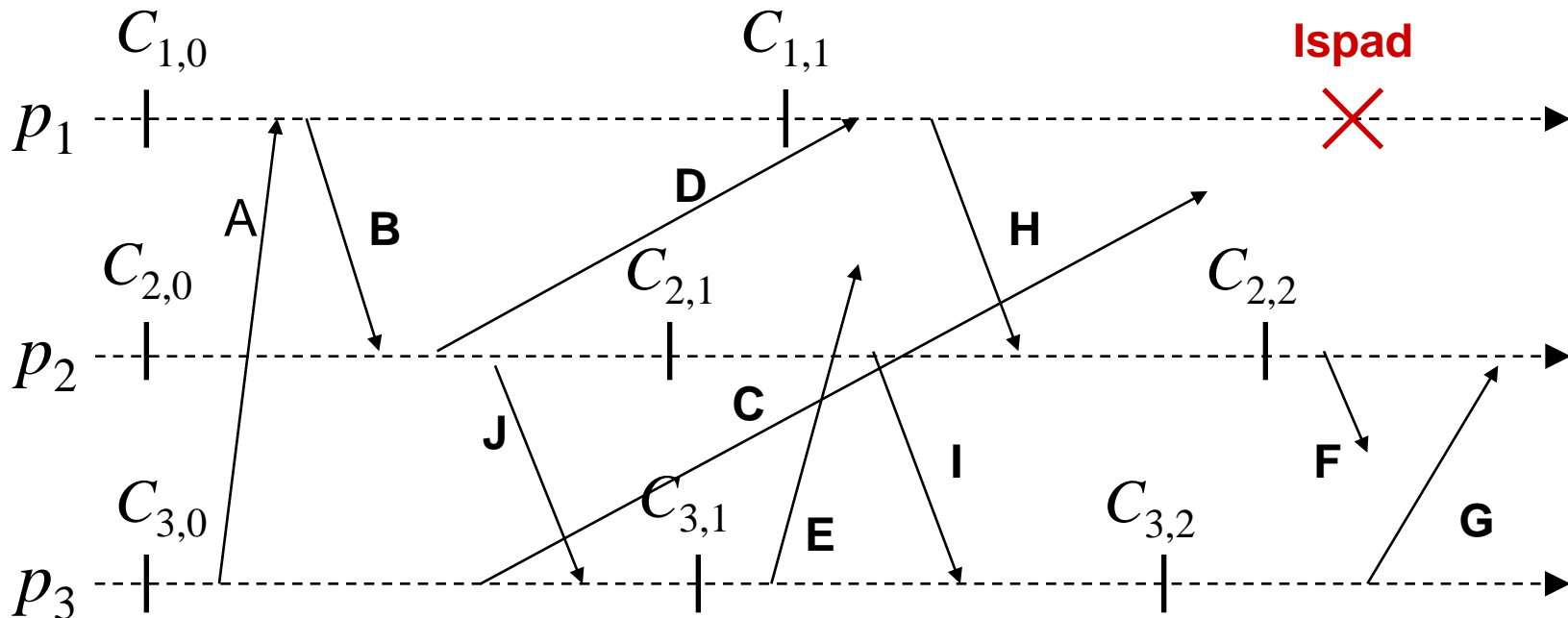
# Konzistentna i nekonzistentna stanja



P1 se vraća u stanje prije slanja  $m_1$   
a P2 prije primitka  $m_1$

P2 se vraća u stanje prije slanja  $m_2$ ,  
a P3 u stanje nakon primitka  $m_2$

# Oprava povratkom unazad



Konzistentno stanje sustava

$C_{1,1}$ ,  $C_{2,1}$ ,  $C_{3,1}$

A, B: normalne poruke

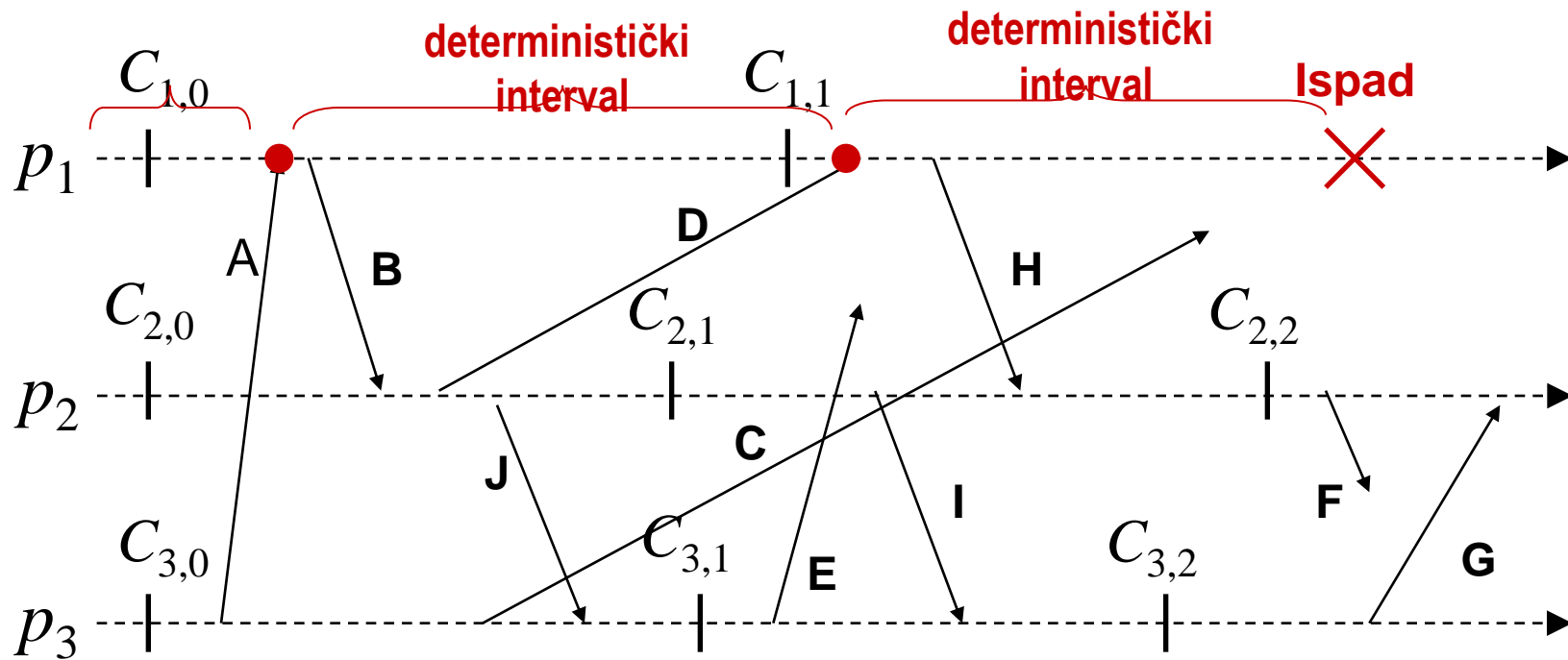
C: "odgođena" poruka (ne zna se kad će/da li će stići)

D: izgubljena poruka (neće se ponoviti)

E: "odgođena" suvišna poruka (treba se odbaciti)

# Opravak primjenom dnevničkog zapisa

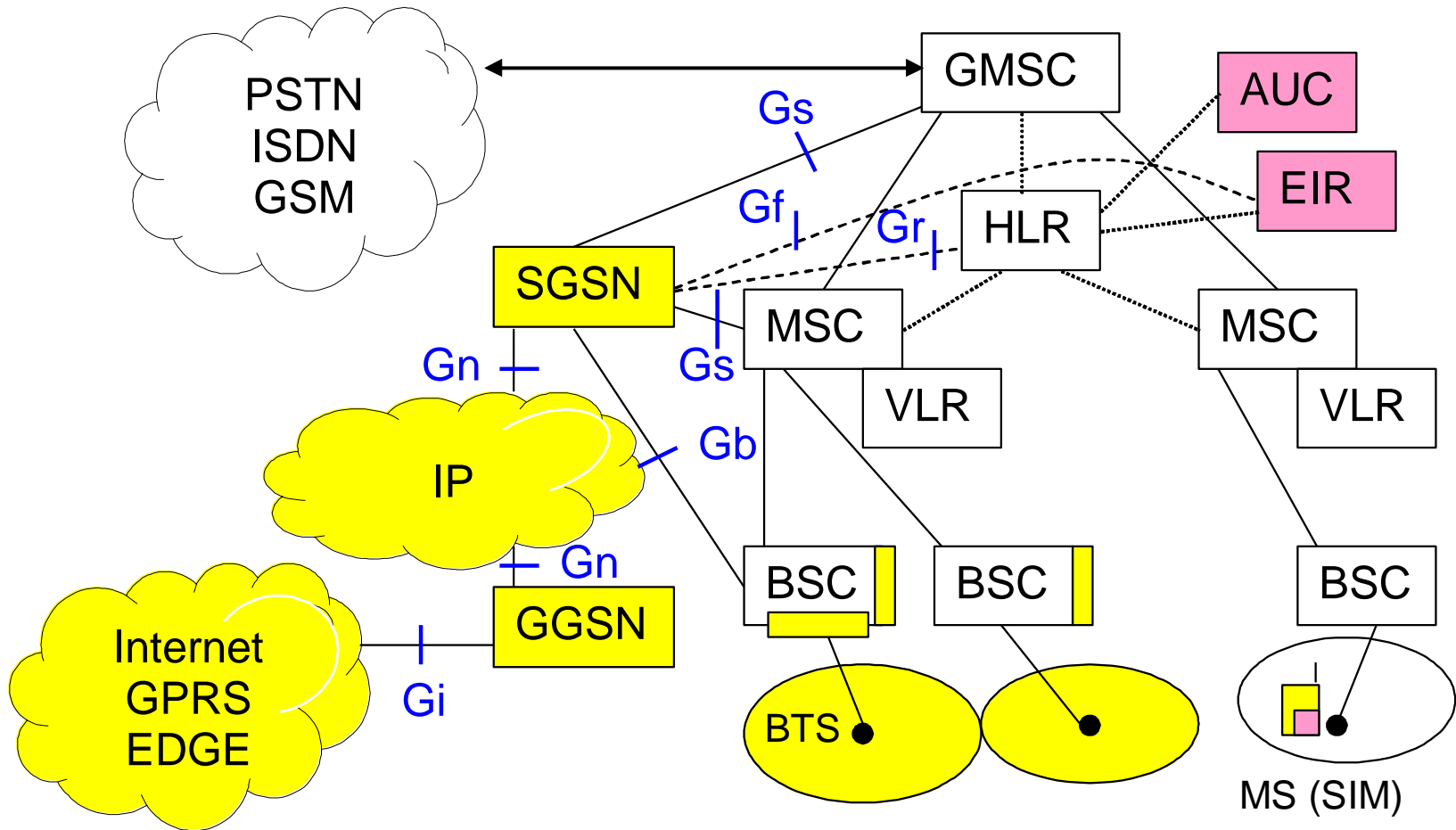
- ◆ Koristi kombinirano kontrolne točke i dnevnički zapis
- ◆ Deterministički interval: počinje nedeterminističkim događajem koji se zapisuje je dnevnički zapis



---

# Sigurnost: studijski primjer pokretne mreže

# Sigurnost u mreži GSM/GPRS/EDGE



## MSISDN (*Mobile Subscriber ISDN*) number

- ◆ pozivni broj pokretnog pretplatnika
- ◆ dodjeljuje mrežni operator

## IMSI (*International Mobile Subscriber Identity*)

- ◆ međunarodni identitet pokretnog pretplatnika
- ◆ dodjeljuje mrežni operator

## IMEI (*International Mobile Equipment Identity*)

- ◆ međunarodni identitet pokretne opreme
- ◆ dodjeljuje proizvođač opreme

$K_i$

- ◆ jedinstven, 128 bita
- ◆ osiguranje komunikacije na zračnom sučelju (MS-BTS)
- ◆ ne izmjenjuje se kroz mrežu, već se izravno upisuje u SIM i AUC
- ◆ algoritmi A3 (SRES) i A8 (Kc) za sigurnosni vektor

## *SIM (Subscriber Identity Module)*

- ◆ MSISDN, IMSI
- ◆ Ki, algoritmi A3 i A8

## *HLR (Home Location Register)*

- ◆ MSISDN, IMSI

## *AUC (Authentication Centre)*

- ◆ Ki

## *EIR (Equipment Identity Register)*

- ◆ IMEI

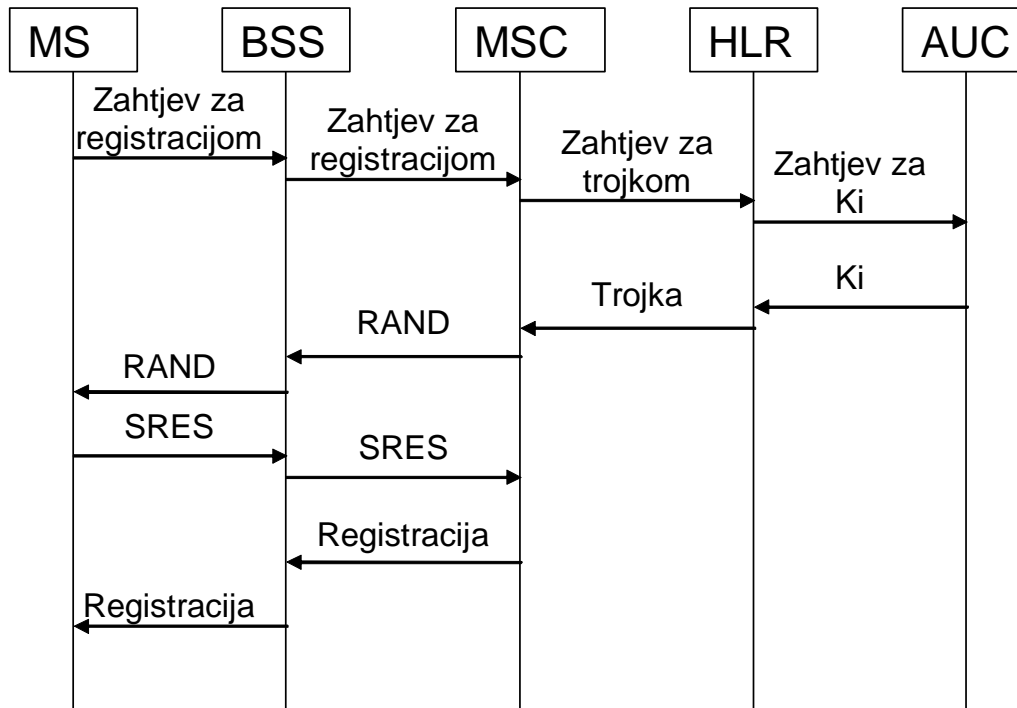
## Sigurnosna prijetnja

- ◆ poznavanje IMSI omogućuje krivo predstavljanje i neovlašteni pristup mreži, jer IMSI je jednoznačno povezan s MSISDN

## Zaštita

- ◆ provjera autentičnosti SIM-a prigodom zahtjeva za registracijom
- ◆ pretpostavke:
  - sigurni BSS, MSC, VLR i AUC
  - povjerenje između BTS i BSC, BSC i MSC, MSC i HLR te HLR i AUC

# Autentičnost pretplatnika (2)



**(RAND, SRES, Kc)**

**RAND – slučajni broj, 128 bita**

**SRES – odgovor na RAND  
generiran s Ki, 32 bita**

**Kc – sjednički ključ (tajnost),  
generiran s Ki, 64 bita**

**HLR generira 5 trojki**

**MSC odabire jednu trojku**

**MSC uspoređuje SRES-ove**

## Sigurnosna prijetnja

- ◆ gubitak ili krađa pokretne opreme

## Zaštita

- ◆ prijava gubitka ili krađe opreme mrežnom operatoru zapisuje se u EIR:
  - kompromitiranom MS ne omogućuje se autentifikacija
- ◆ i HLR :
  - kompromitiranom SIM-u zabranjuje se pristup mreži

## Sigurnosna prijetnja

- ◆ mreža upotrebljava IMSI za obradu poziva i usluga
- ◆ IMSI je jednoznačno povezan s MSISDN
- ◆ dohvaćanjem IMSI na zračnom sučelju može se ustanoviti pretplatnikova lokacija i pratiti kretanje

## Zaštita

- ◆ privremeni identitet pokretnog pretplatnika TMSI (*Temporary Mobile Subscriber Identity*), lokalnog značenja za razliku od IMSI
- ◆ nakon provjere autentičnosti pretplatnika, mreža mu dodjeljuje TMSI, čime se smanjuje upotreba IMSI na zračnom sučelju
- ◆ preslikavanje IMSI-TMSI provode VLR i MSC

## Sigurnosna prijetnja

- ◆ prisluškivanje na zračnom sučelju

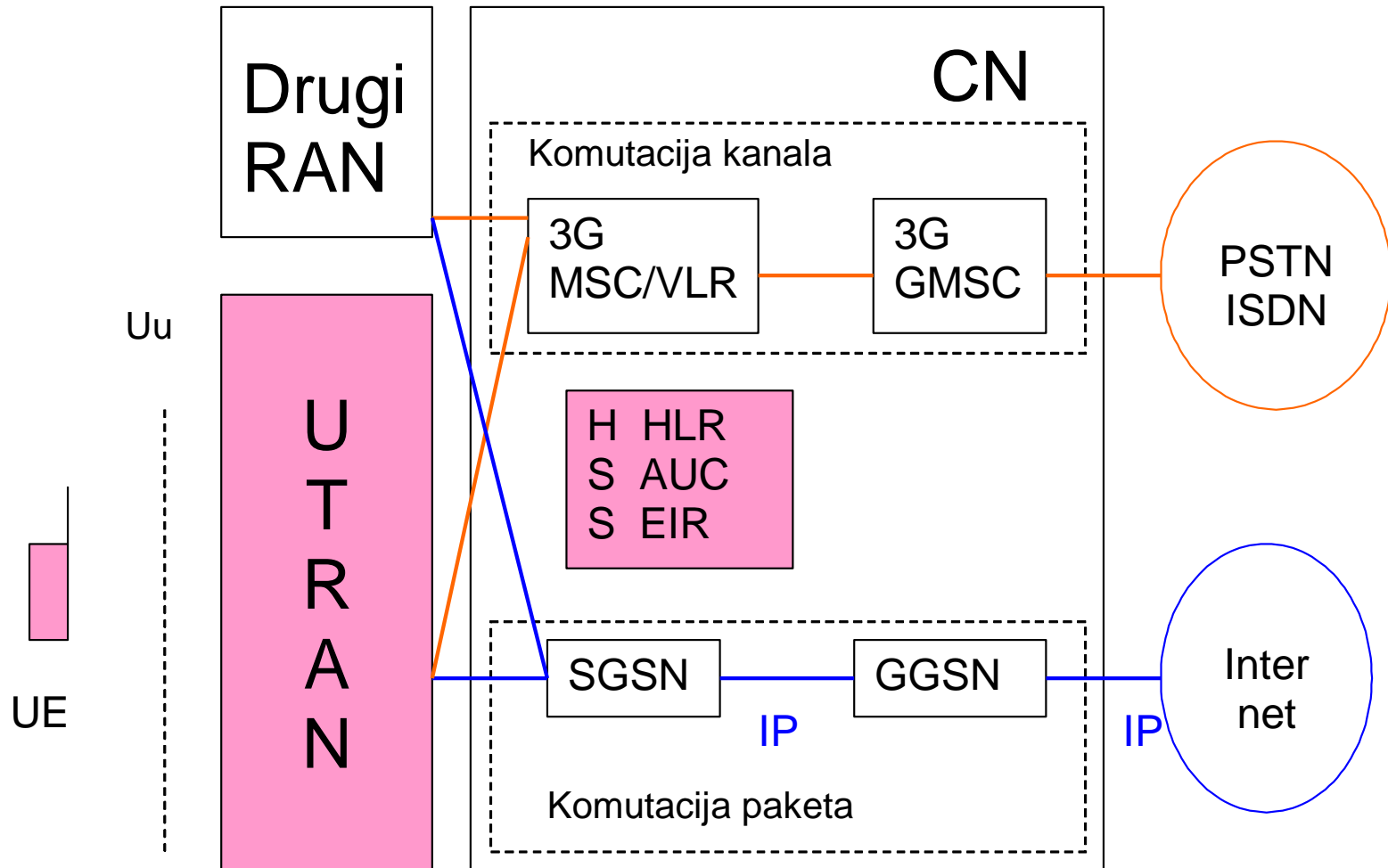
## Zaštita

- ◆ šifriranje podataka na zračnom sučelju:
  - algoritam A5 (u MS)
  - sjednički ključ Kc
- ◆ svaka komunikacija novi Kc

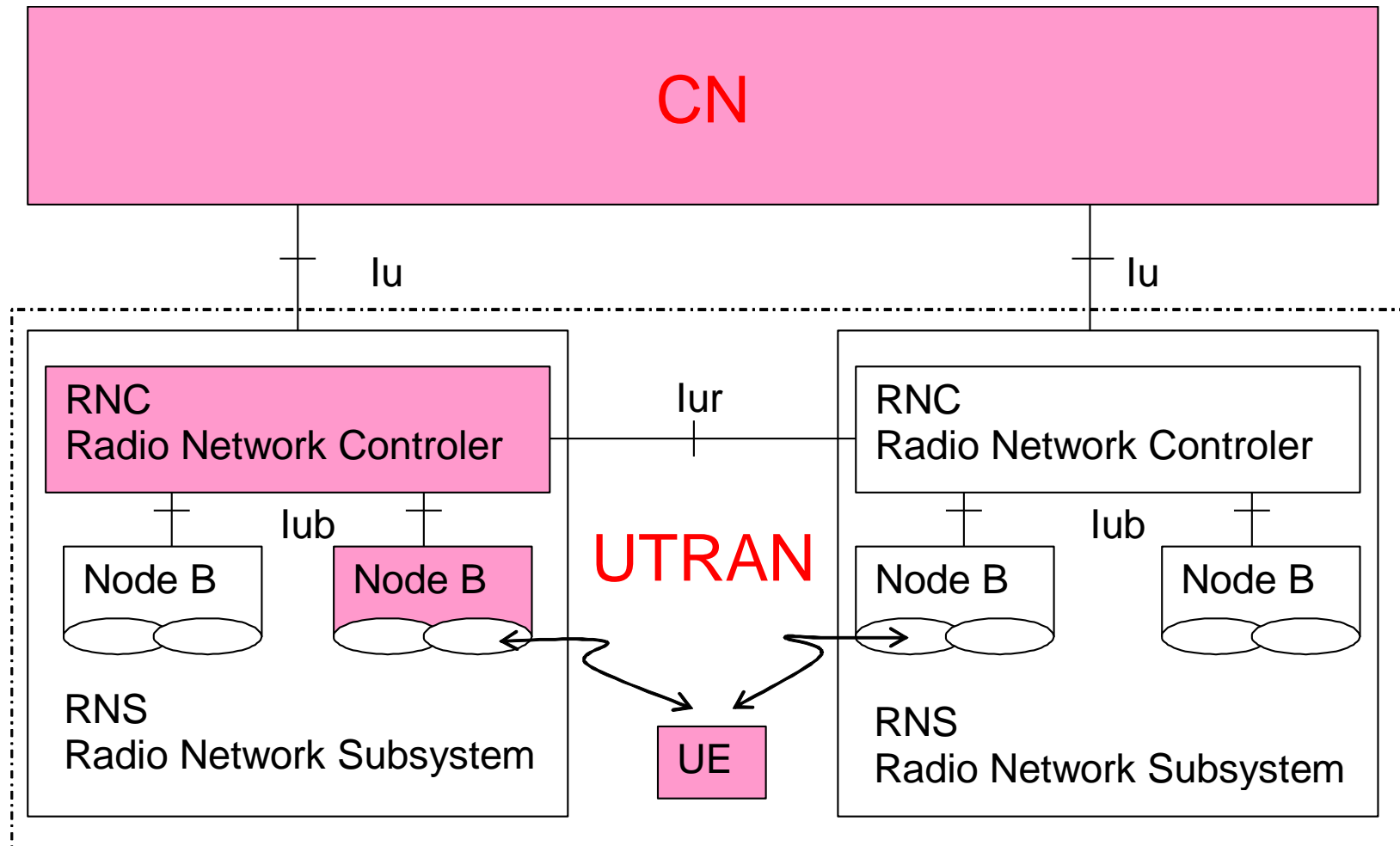
- ◆ sigurnosne mjere usredotočene su na zračno sučelje,
- ◆ veza BTS-BSC potencijalno izložena napadima,
- ◆ signalizacijski sustav (SS7) bez kriptografske zaštite,
- ◆ ne štiti se integritet podataka,
- ◆ pristup algoritmima se strogo kontrolira, a time je spriječeno istraživanje njihovih slabosti
  - ključ, a ne algoritam treba biti tajan – mišljenje kriptanalitičara!

- ◆ komunikacija u više vremenskih odsječaka preko više BTS-a dovodi do problema očuvanja tajnosti na relaciji MS-BTS, tako da se ista provodi na relaciji MS-SGSN
- ◆ MS ograničenih mogućnosti ne dopušta primjenu modela internetske sigurnosti (SSL – *Secure Socket Layer*) pa se primjenjuje WAP, odnosno njegov sloj WTLS (*WAP Transport Layer Security*)

# Sigurnost u mreži UMTS (1)



# Sigurnost u mreži UMTS (2)



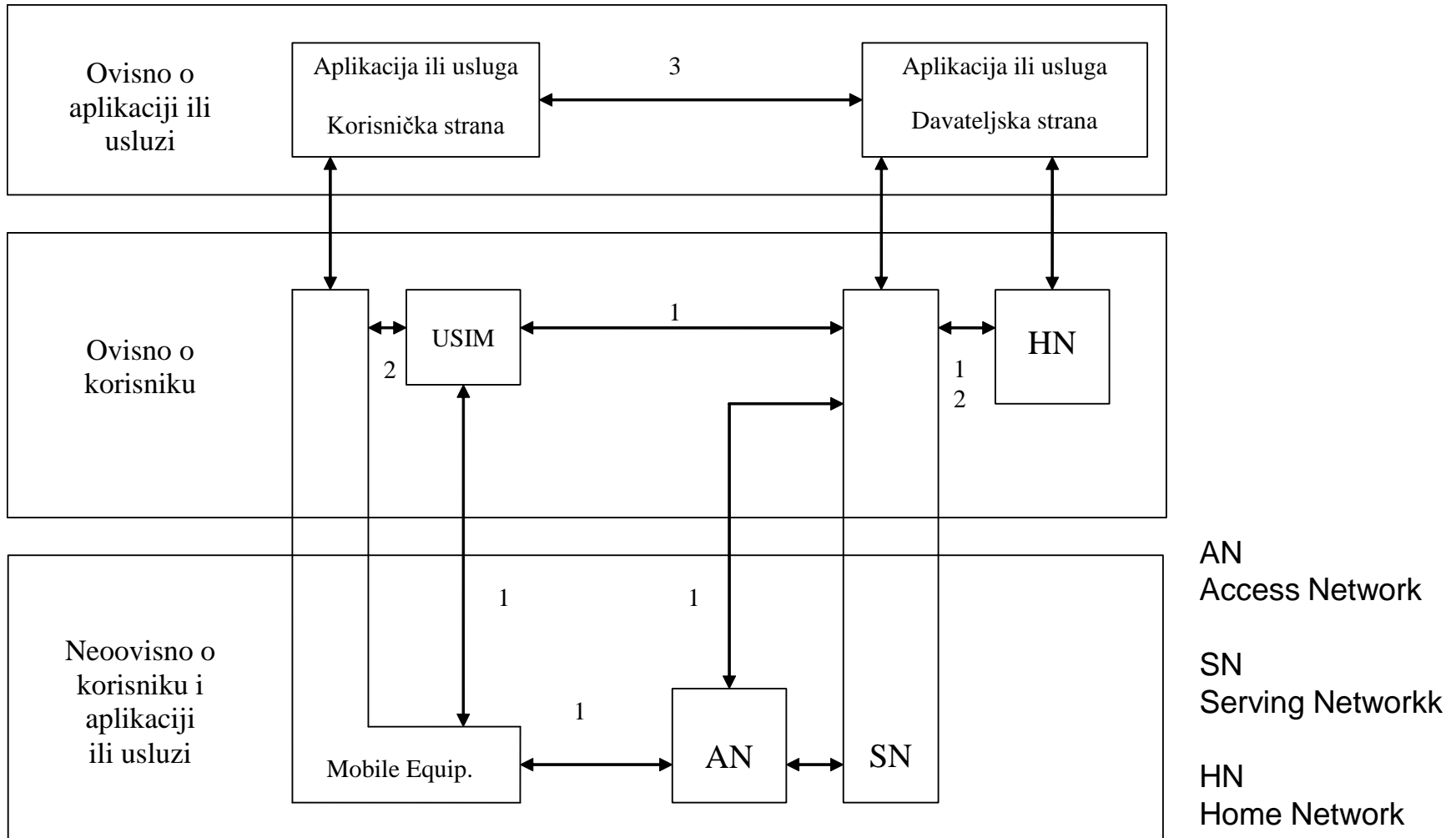
## ◆ Preuzimanje dobrih rješenja GSM-a:

- ključ Ki se ne razmjenjuje kroz mrežu,
- mreža provjerava autentičnost USIM-a

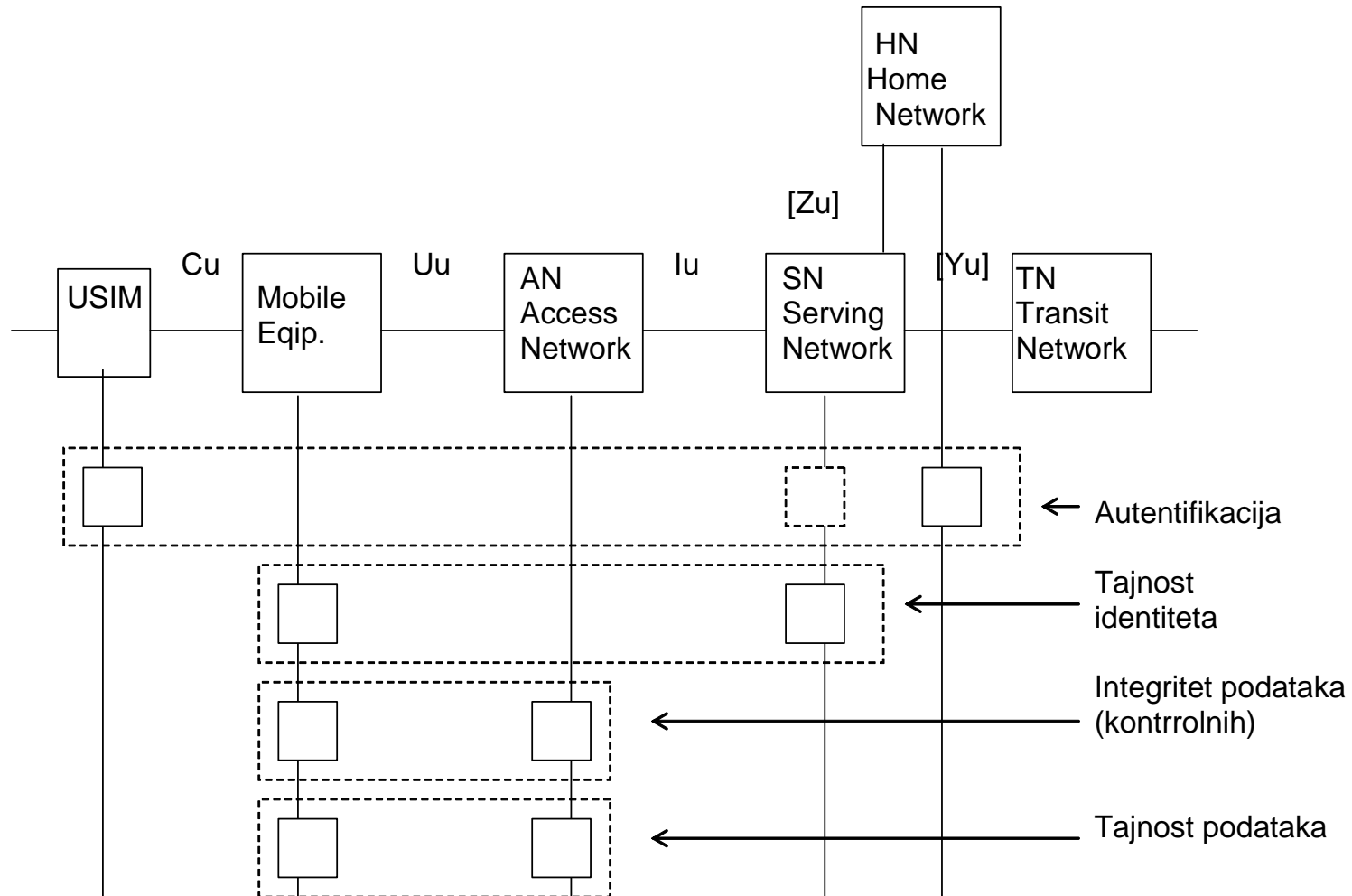
## ◆ Promjena slabijih rješenja GSM-a:

- šifriranje signalizacijske informacije koja omogućuje otkrivanje identiteta pretplatnika,
- provjera autentičnosti mreže
- tajnost: algoritam KASUMI, sjednički ključ CK
- integritet signalizacijske informacije: ključ IK

# Sigurnosni model UMTS-a



# Sigurnosna rješenja UMTS-a (1)



## Autentifikacija:

- ◆ provodi se na temelju ključa zapisanog u USIM-u i HN-u. HN može prepustiti autentičnosti SN-u;

## Tajnost korisnikovog identiteta:

- ◆ korisniku se dodjeljuje privremeni tajni identitet koji upotrebljavaju ME i SN;

## Integritet podataka:

- ◆ čuva se samo integritet kontrolnih podataka, a za korisničke podatke se smatra da su zaštita od pogrešaka i tajnost dovoljni;

## Tajnost podataka:

- ◆ korisnički podaci.

## USIM (*UMTS Subscriber Identity Module*)

- ◆ MSISDN, IMSI
- ◆  $K_i$ , algoritmi  $f_1(\text{AUTN})$ ,  $f_2(\text{XRES})$ ,  $f_3(\text{CK})$  i  $f_4(\text{IK})$

## HLR (*Home Location Register*)

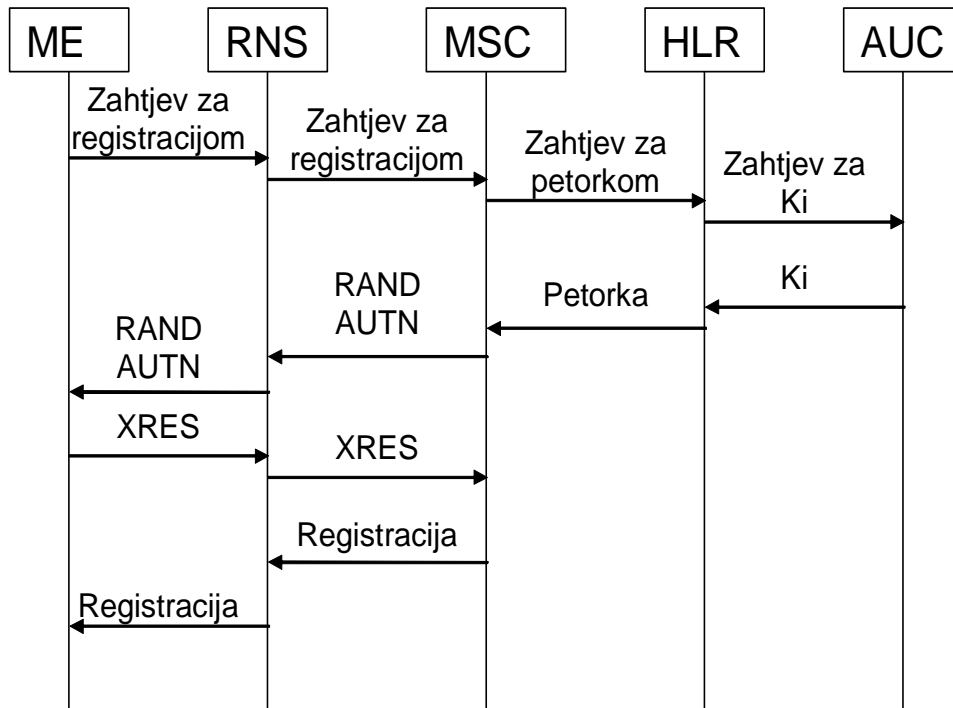
- ◆ MSISDN, IMSI

## AUC (*Authentication Centre*)

- ◆  $K_i$

## EIR (*Equipment Identity Register*)

- ◆ IMEI



**(RAND, XRES, CK, IK, AUTN)**

**RAND – slučajni broj, 128 bita**

**XRES – odgovor na RAND  
generiran s Ki, 32 bita**

**CK – sjednički ključ (tajnost),  
128 bita**

**IK – integritetski ključ, 128 bita**

**AUTN – oznaka autentičnosti  
mreže, 128 bita**

**ME provjerava AUTN**

## Pri oblikovanju raspodijeljenog sustava i njegovoj primjeni

- ◆ evaluirati temeljna obilježja:
  - Otvorenost, transparentnost i skalabilnost
- ◆ odabrati/ustanoviti teorijske postavke
  - Interakcijski model, model kvara i model sigurnosti
- ◆ predložiti/ispitati:
  - Odvijanje operacija u vremenu, konzistentnost podataka, oporavak nakon kvara i sigurnosne postavke

- ◆ A. S. Tanenbaum, M. Van Steen: Distributed Systems: Principles and Paradigms, Second Edition, Prentice Hall, 2007.
- ◆ G. Coulouris, J. Dollimore, T. Kindberg: Distributed Systems: Concepts and Design, 4th edition, Addison-Wesley, 2005
- ◆ Ajay D. Kshemkalyani, Mukesh Singhal, Distributed Computing: Principles, Algorithms, and Systems, Cambridge University Press, 2008.
- ◆ N. Lynch: Distributed Algorithms, Morgan Kaufmann Publishers Inc., 1996