

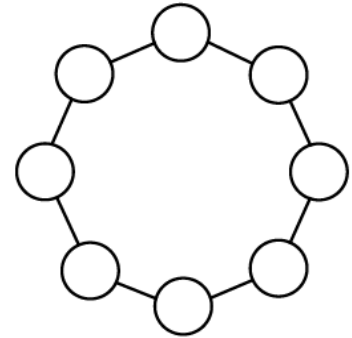
8. Raspodijeljeni sustavi

Komunikacija među čvorovima

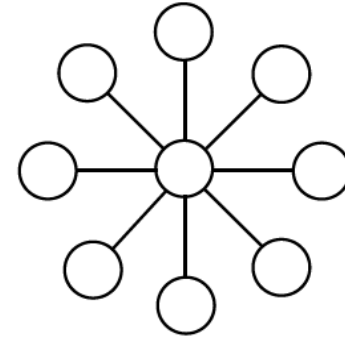
- ❑ svaki čvor radi nešto, upravlja nekim dijelom procesa, ima svoj „program”
- ❑ čvorovi zajedno sudjeluju u upravljanju sustavom
- ❑ čvorovi komuniciraju međusobno:
 - izravno – razmjenjuju informacije nekim medijem/protokolom
 - neizravno – djelovanjem u okolini (senzorima detektiraju promjene)
- ❑ mogući protokoli?
 - Internet (TCP/IP - *transmission control protocol/Internet protocol*)
 - prednost: „univerzalan”, svi ga „znaju”
 - problem: složen, slojevit, unosi dodatne odgode
 - ne može se izbjeći u većim raspodijeljenim sustavima
 - neki jednostavniji (npr. CAN)

Povezivanje računala u mrežu – fizičko povezivanje

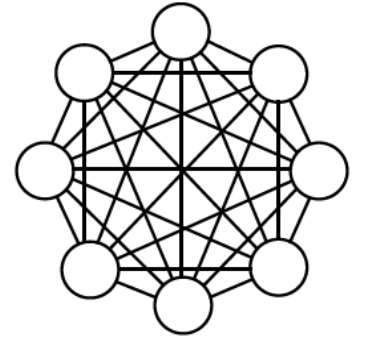
- ❑ mnoštvo mogućnosti
- ❑ svaka ima svoje prednosti, nedostatke, primjenu
- ❑ neka svojstva o kojima treba odlučiti (ili su već zadana):
 - zajednički medij
 - centralizirani čvor
 - lako dodavanje/micanje
 - cijena
 - fizička organizacija čvorova
 - tražena vremenska svojstva



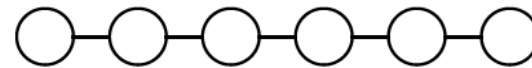
a. prsten



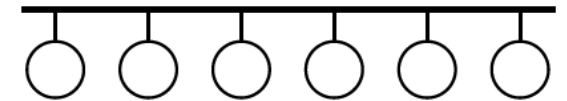
b. zvijezda



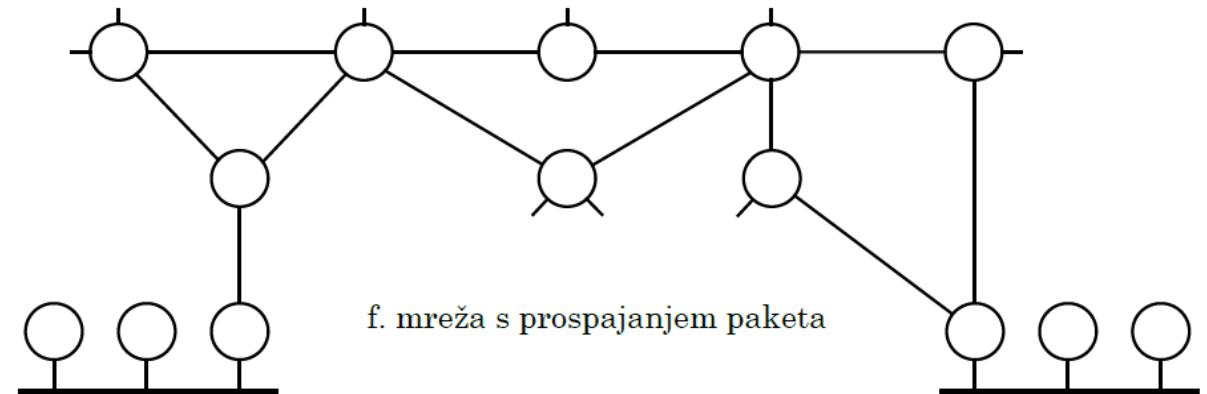
c. potpuna povezanost



d. lanac



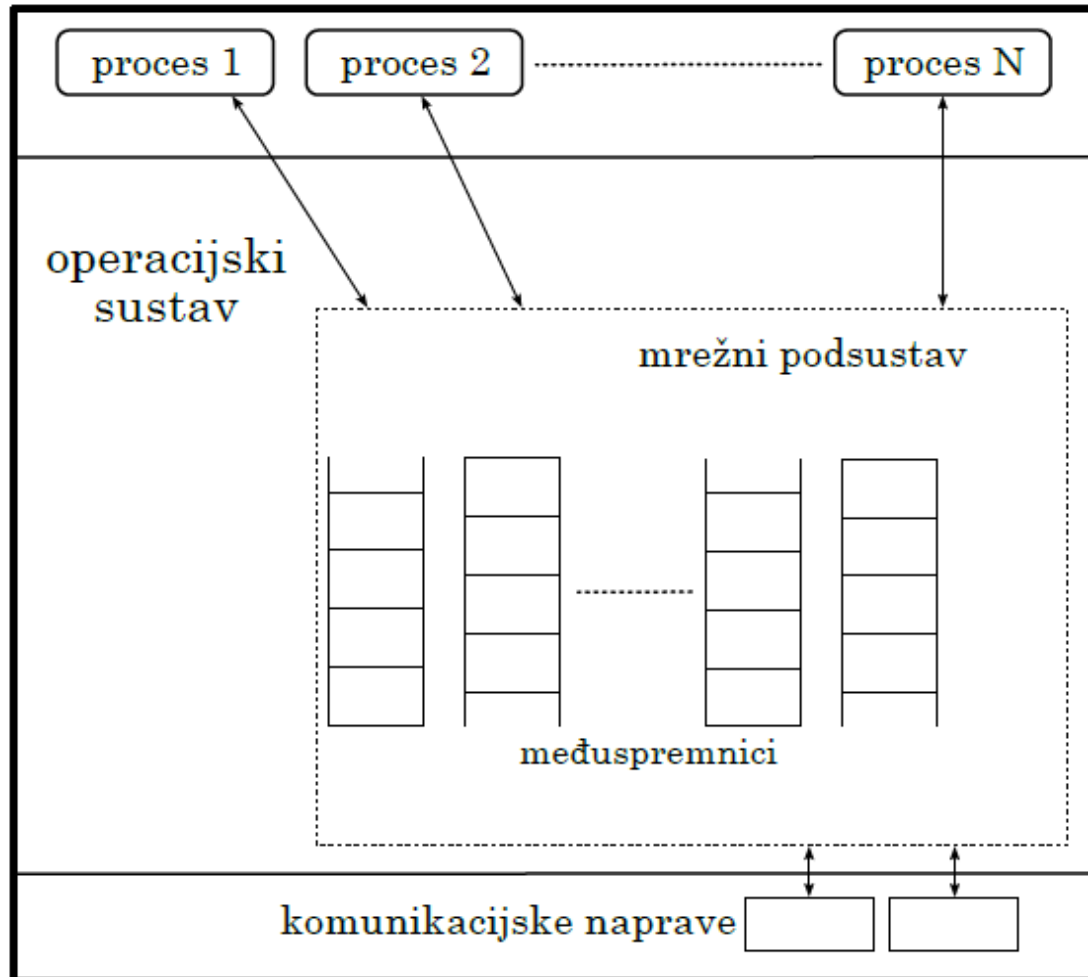
e. sabirnica



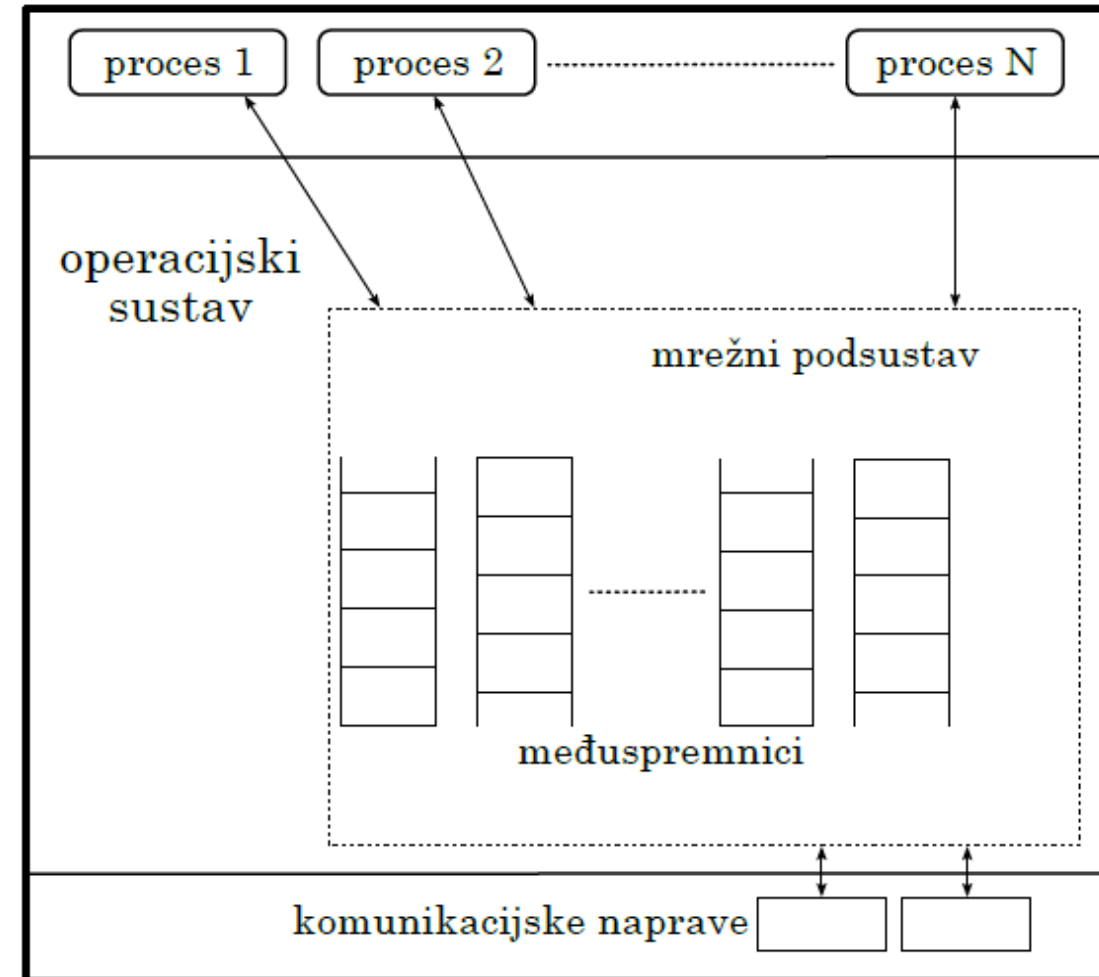
f. mreža s prosipanjem paketa

Model komunikacije (logički, stvarno, slojevitost)

računalo A



računalo B



komunikacijska infrastruktura (kabeli, usmjernici, ...)

Svojstva komunikacijskog sustava

- ☐ kašnjenje (engl. delay)
- ☐ brzina/propusnost mreže (engl. throughput, bandwidth)
- ☐ varijacije u kašnjenju (engl. delay jitter)
- ☐ postotak prekasno pristiglih paketa (engl. miss rate)
- ☐ postotak izgubljenih paketa (engl. loss rate)
- ☐ postotak pogrešaka pri prijenosu (engl. invalid rate)
- ☐ veličina paketa
- ☐ dimenzije mreže (udaljenosti).

Controller Area Network – CAN

- ❑ „industrijska mreža”
- ❑ koristi se često i u automobilima
- ❑ zajednička sabirnica
 - dvije žice – ista informacija ali u „obrnutim” napinima (CAN visoko/nisko)
 - do 50 m duljine
 - do 1 Mbit/s
 - CSMA/CD
- ❑ svaka poruka započinje identifikatorom
 - nema adrese „tko šalje, kome se šalje”, svi mogu primati
 - pri koliziji pri slanju koristi se svojstvo dominantnog bita za rješavanje

CAN – primjer kolizije

Primjer 8.1.

Pretpostavimo da dva čvora istovremeno započinju sa slanjem svojih poruka, prvi s identifikatorom 57, a drugi s 43. S obzirom na to da manji broj predstavlja veći prioritet, druga poruka treba biti prioritetnija. Binarno, identifikatori su:

$$57_{10} = 000001\underline{1}1001_2 \quad (8.1.)$$

$$43_{10} = 000001\underline{0}1011_2 \quad (8.2.)$$

Prvi čvor će tek pri slanju 7. bita primijetiti da poslano ne odgovara očitano na sabirnici te će on tada prestati slati, dok će drugi čvor svoju poruku poslati do kraja.

CAN (nastavak)

- ❑ CAN poruka: id + upravljačko zaglavlje + podatci
- ❑ tipovi poruka:
 - podatkovne
 - zahtjev za podacima
 - poruke o greškama
 - poruke preopterećenja

CAN (nastavak)

□ još neka svojstva:

- licencirana tehnologija, treba licenca za korištenje!
- prioritetna shema poruka
- jamstvo da kašnjenje neće biti iznad definiranog
- prilagodljivo podešavanje (čak i nije neophodno)
- slanje prema više čvorova (engl. multicast)
- osiguravanje konzistencije podataka
- ugrađeni postupci otkrivanja grešaka i njihovo signaliziranje
- automatizirana retransmisija kao opcija
- razlikovanje privremene greške od kvarova u čvorovima (koji se tada mogu i isključiti)

RTP/RTCP

- ❑ RTP: Real-Time Protocol
- ❑ RTCP: Real-Time Control Protocol
- ❑ osmišljeni su za
 - prijenos multimedije
 - videokonferencije
 - interaktivne multimedijalne aplikacije
 - za raspodijeljene simulacije
- ❑ RTP – prenosi podatke
- ❑ RTCP – upravljanje prijenosom, po jedan kanal za svaki RTP kanal
- ❑ koriste UDP za prijenos

TCP vs UDP

- ❑ TCP nudi „pouzdan prijenos podataka”
 - ali to uključuje i retransmisije neprimljenih podataka ili podatak s greškom
 - retransmisija uzrokuje dodatne odgode
- ❑ u SRSV-ima želimo veću kontrolu
 - ako treba sami ćemo napraviti retransmisiju
 - „zastarjeli” podatci (koje bi ponovno slali) nam možda nisu potrebni
 - TCP sam radi retransmisiju, nema „nemoj to raditi” opcije
 - zato se umjesto TCP-a koristi UDP
- ❑ UDP – User Datagram Protocol
 - svaka poruka ide zasebno
 - ne radi sam retransmisiju

RTP/RTCP (2)

- ❑ pri prijenosu multimedije svaki oblik ide zasebnim kanalima (npr. zvuk i slika)
 - u njih se ugrađuju vremenske oznake, redni brojevi i sl.
- ❑ u komunikaciji može sudjelovati puno čvorova
- ❑ neki posebni čvorovi:
 - mikseri, translatori
- ❑ RTCP kanal za kontrolu kvalitete
 - za moguću prilagodbu podataka, npr. promjenu kodiranja, bit rate i sl.
 - svaki čvor povremeno šalje svoja izvješća o kvaliteti
 - RTCP koristi samo mali dio propusnosti, da ne zakrči, do 5%

Sinkronizacija vremena u raspodijeljenim sustavima

- ❑ kako uskladiti aktivnosti koje pokreću različiti čvorovi?
- ❑ je li potrebno samo uskladiti redoslijed ili i vrijeme?
- ❑ za redoslijed se može koristiti npr. raspodijeljeni Lamportov algoritam
- ❑ za vrijeme?
 - svaki čvor ima svoj sat
 - ali jesu li ti satovi usklađeni?
 - kako ih uskladiti?
- ❑ format sata: UTC - Coordinated Universal Time
 - Hrvatska u zoni UTC+1
- ❑ kako dohvatiti točno vrijeme?
 - GPS sateliti su jedna mogućnost, poslužitelj točnog vremena druga

Sinkronizacija vremena između dva udaljena čvora (1)

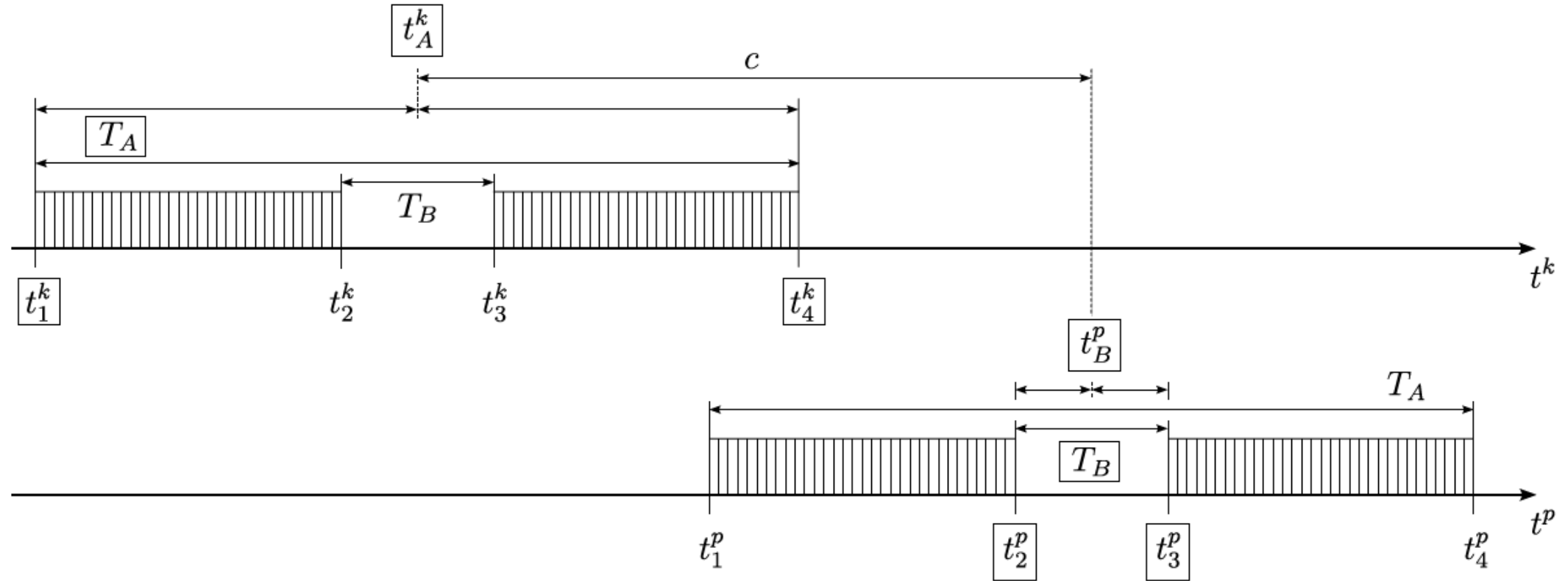
❑ kako uzeti u obzir kašnjenje mreže?

❑ ideja:

- poslati paket s jednog čvora na drugi i obratno
- pritom za svaki događaj zapisati vrijeme:

1. t_1^k – vrijeme slanja zahtjeva, po satu klijenta (satu kojeg treba uskladiti)
2. t_2^p – vrijeme primitka zahtjeva, po satu poslužitelja (satu s kojim se treba uskladiti)
3. t_3^p – vrijeme slanja odgovora, po satu poslužitelja
4. t_4^k – vrijeme primitka odgovora, po satu klijenta.

Sinkronizacija vremena između dva udaljena čvora (2)



- **kašnjenje mreže** (engl. *round trip delay*):
- **razliku u satovima** (engl. *offset*):

$$d = T_A - T_B = (t_4^k - t_1^k) - (t_3^p - t_2^p)$$

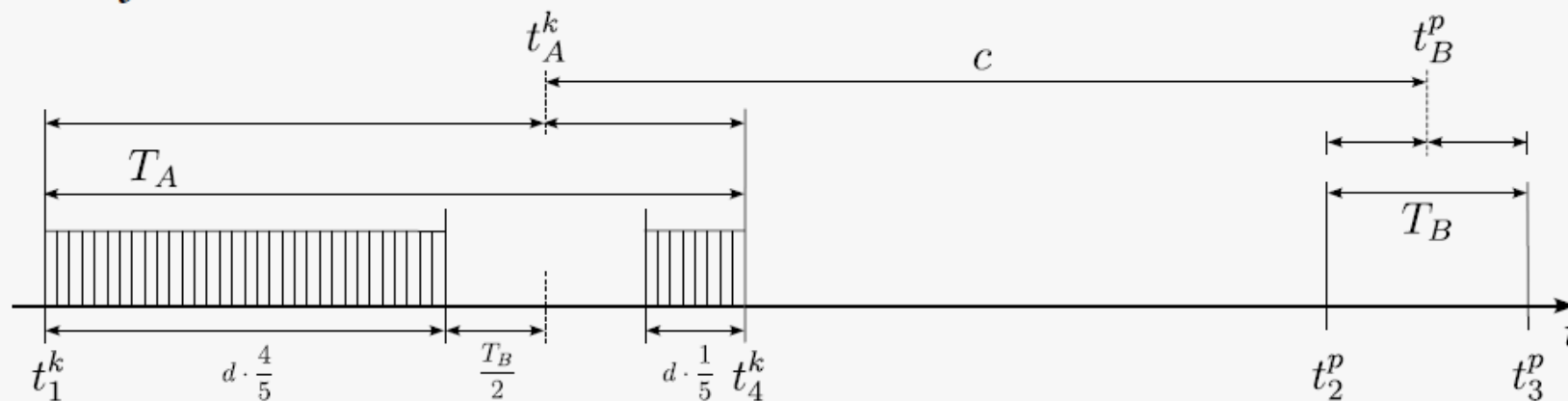
$$c = t_B - t_A = (t_2^p + t_3^p)/2 - (t_1^k + t_4^k)/2$$

Primjer 8.2. Nesimetrična veza

Neka je slanje zahtjeva četverostruko sporije od primanja odgovora (puta paketa u tom smjeru), tada bi se t_A kojim se koristimo u prethodnom primjeru u formuli 8.4. za računanje razlike u satovima sada računao prema:

$$t_A = t_1^k + 4/5 \cdot d + T_B/2 \quad (8.5.)$$

Trajanje prijenosa dijeli se u omjeru 4:1 te je prvi dio 4/5 te vrijednosti, na što se još zbraja pola intervala T_B da bismo došli do iste točke koja odgovara t_B kod poslužitelja. Navedeni primjer prikazuje slika 8.4.



$$d = T_A - T_B \Rightarrow t_A = t_1^k + d \cdot 4/5 + T_B/2 \Rightarrow c = t_B - t_A \quad (8.6.)$$

Primjeri iz skripte

9. U nekom sustavu čvor I pristupa čvoru J putem asimetrične veze: slanje poruka je pet puta sporije od primanja. Ako je lokalno vrijeme slanja poruke u čvoru I 10 te primanja odgovora 50, dok je vrijeme primanja poruke u čvoru J 30 i vrijeme slanja 40 (u lokalnom vremenu čvora J) odredite vrijeme prijenosa podataka od čvora I do J i obratno te razlika među satovima ($t_I = t_J + \text{razlika}$).
10. Satelit A treba uskladiti svoj sat sa satelitom B. Udaljenost među satelitima jest 300 000 km. Opišite postupak sinkronizacije i pokažite njegov rad na primjeru ako je u tom početnom trenutku sat na A imao vrijednost 13:13:13.111111 a sat na B 13:13:13.111222. Obrada poruka traje 0,0001 s.