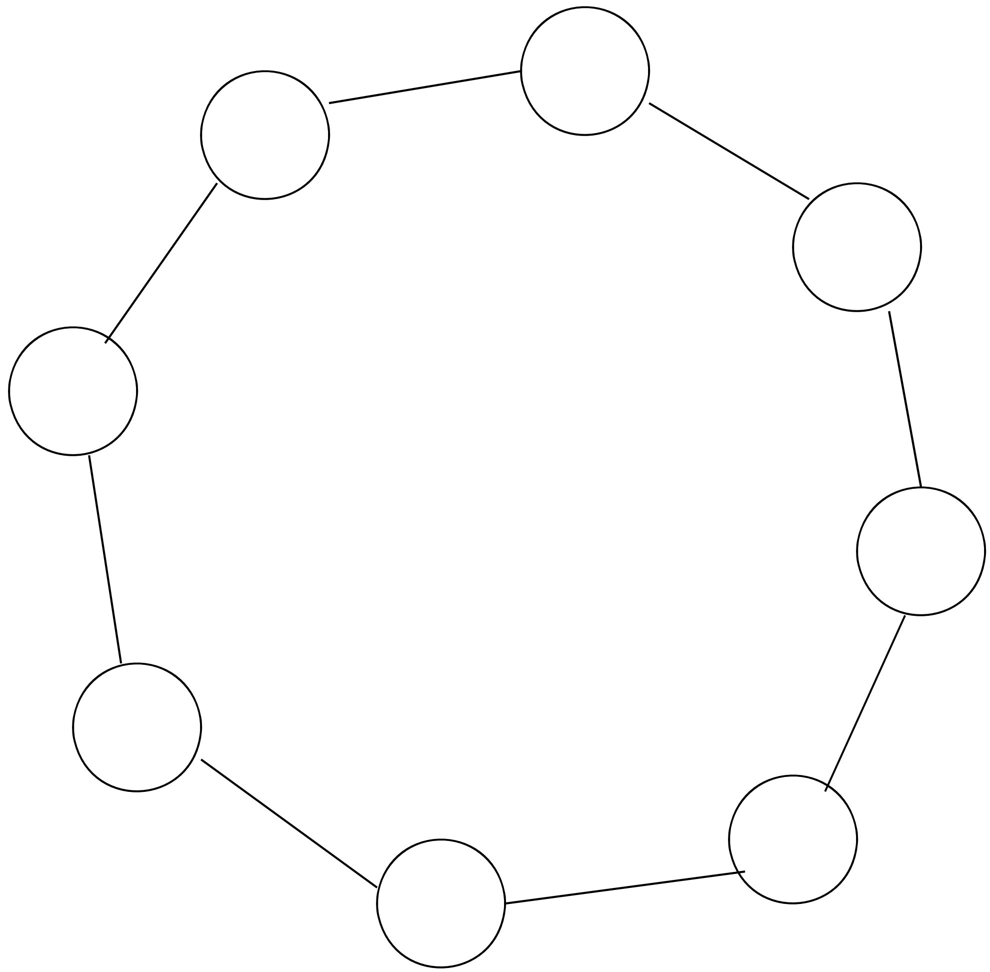
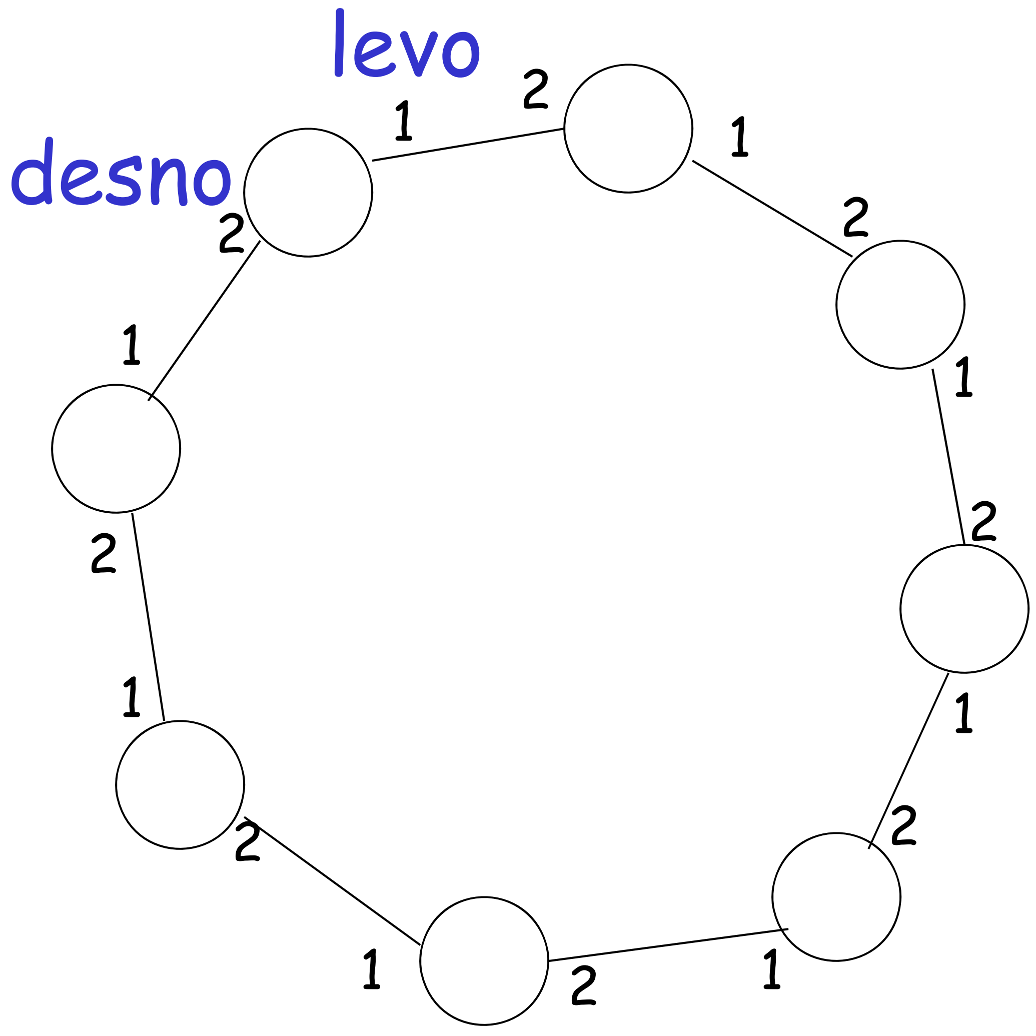
Izbor lidera u prstenovima

Prstenasta mreža

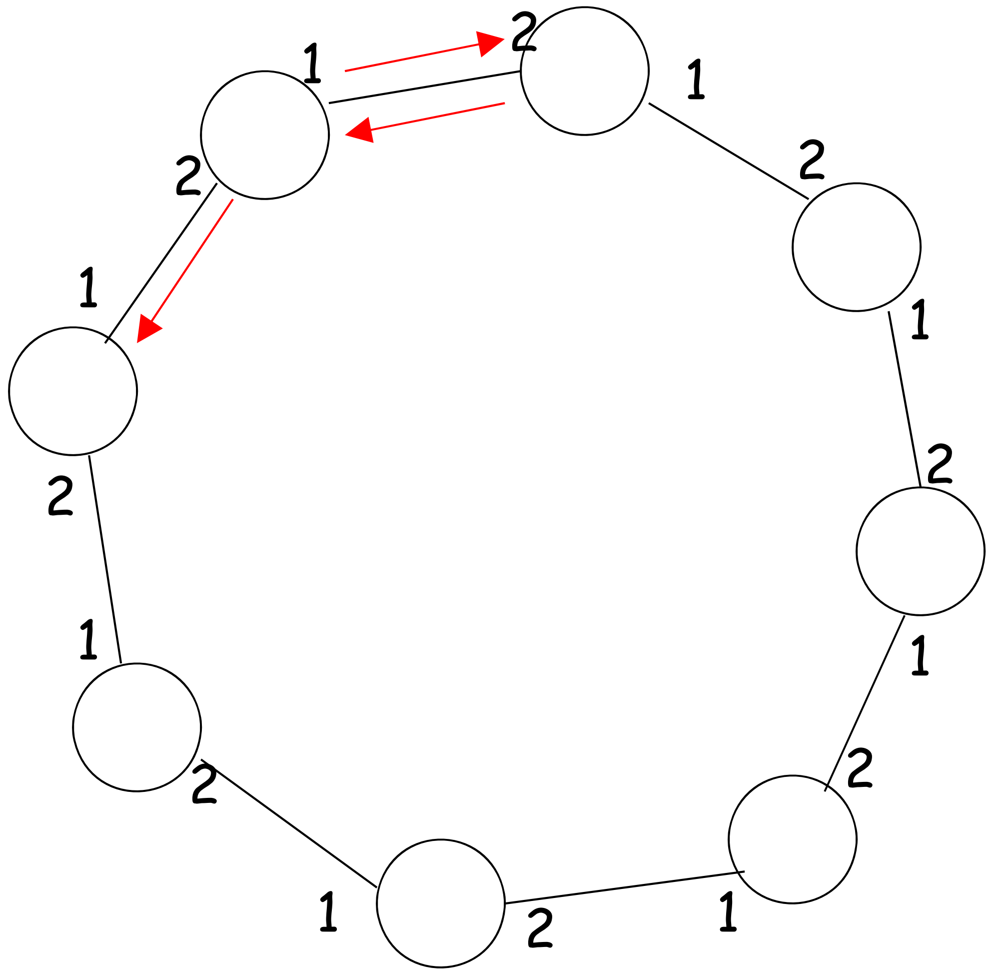


Osećaj pravca

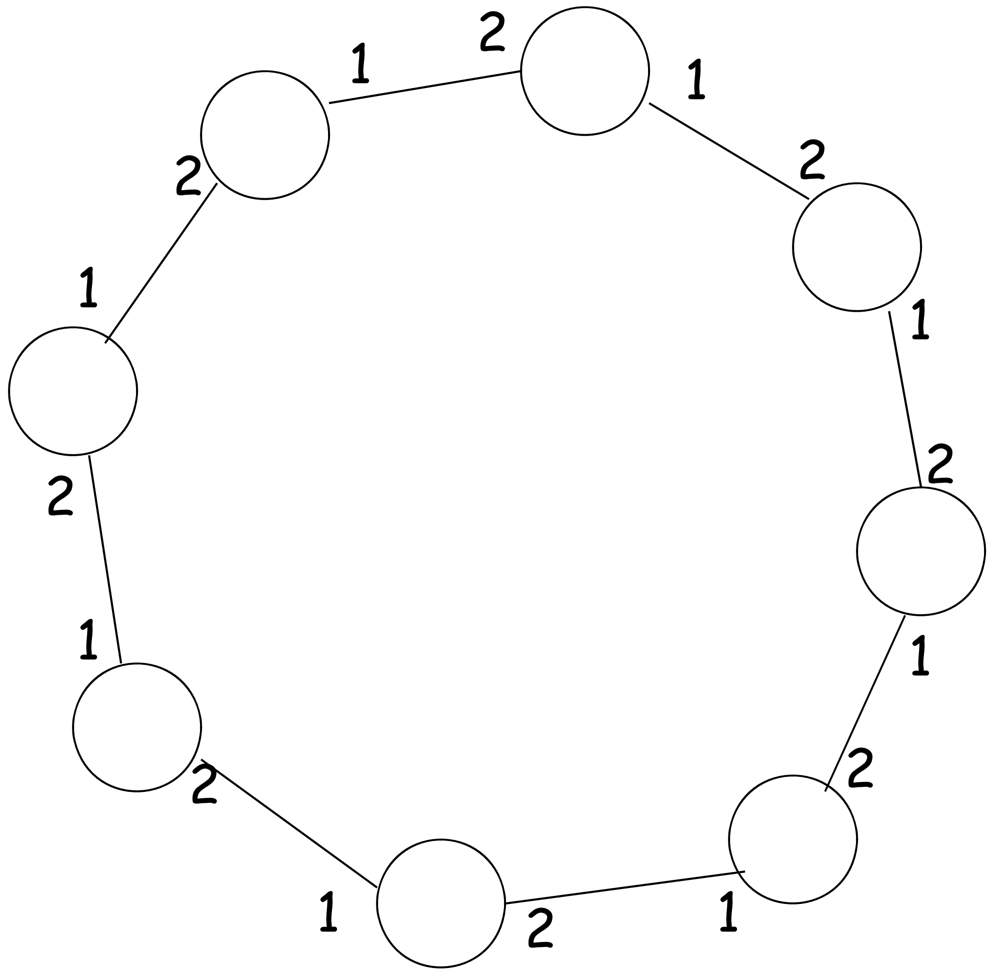


Linkovi su dvosmerni za poruke

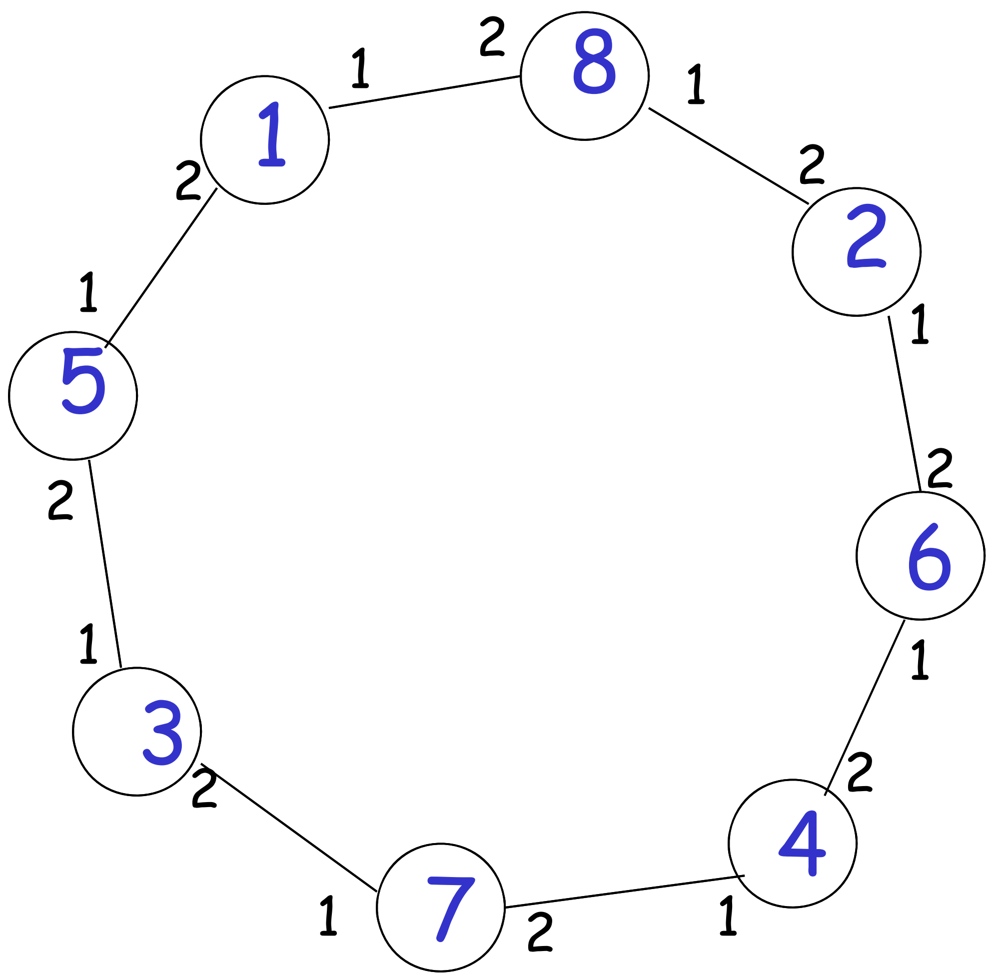
Najviše jedna poruka u svakom smeru



Anonimni prsten



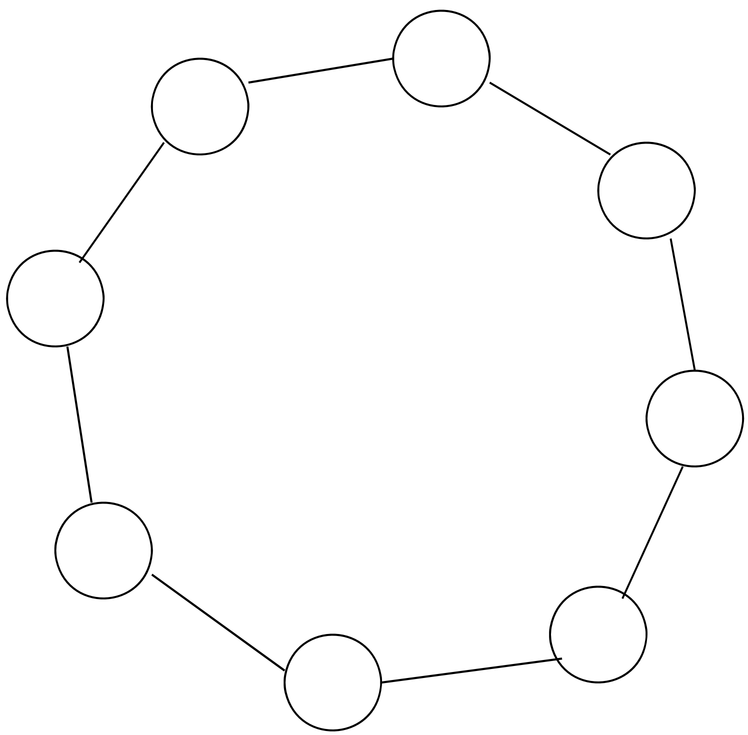
Eponimni (ne-anonimni) prsten



Izbor lidera

Početno stanje

Konačno stanje



Algoritmi izbora lidera zavise od sl. faktora:

Anonimni prsten

Eponimni prsten

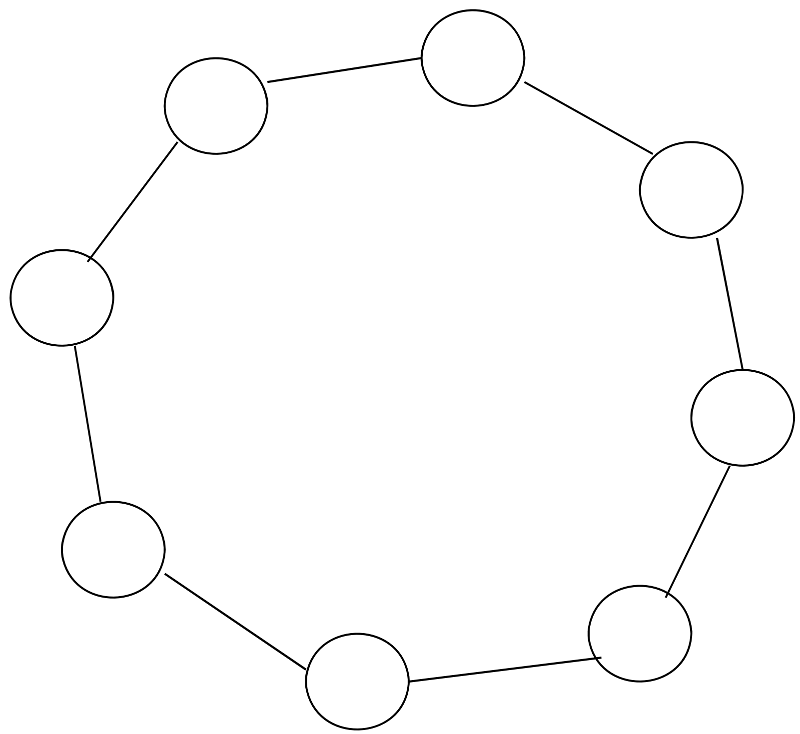
Veličina mreže n je poznata

Veličina mreže n je nepoznata

Sinhroni algoritam

Asinhroni algoritam

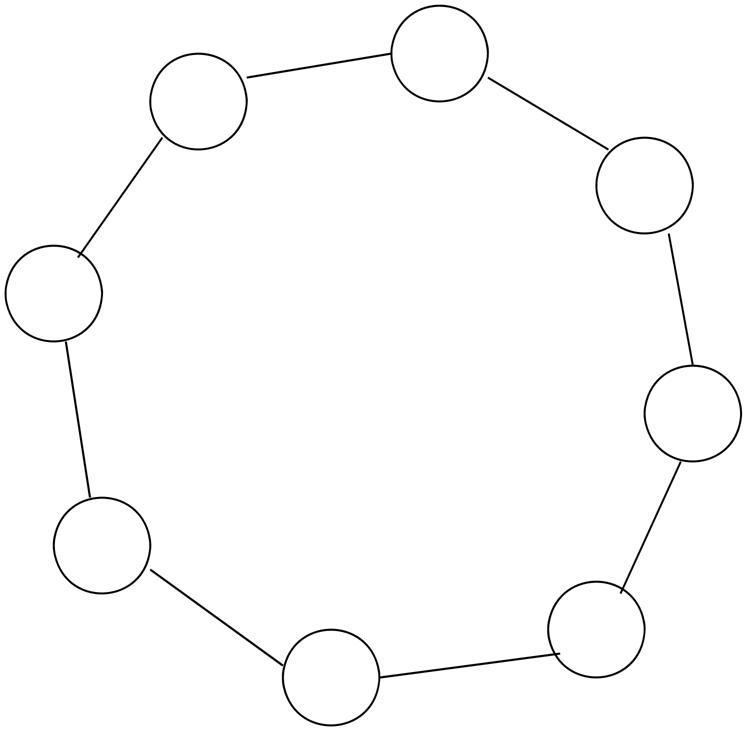
Sinhroni anonimni prsteni



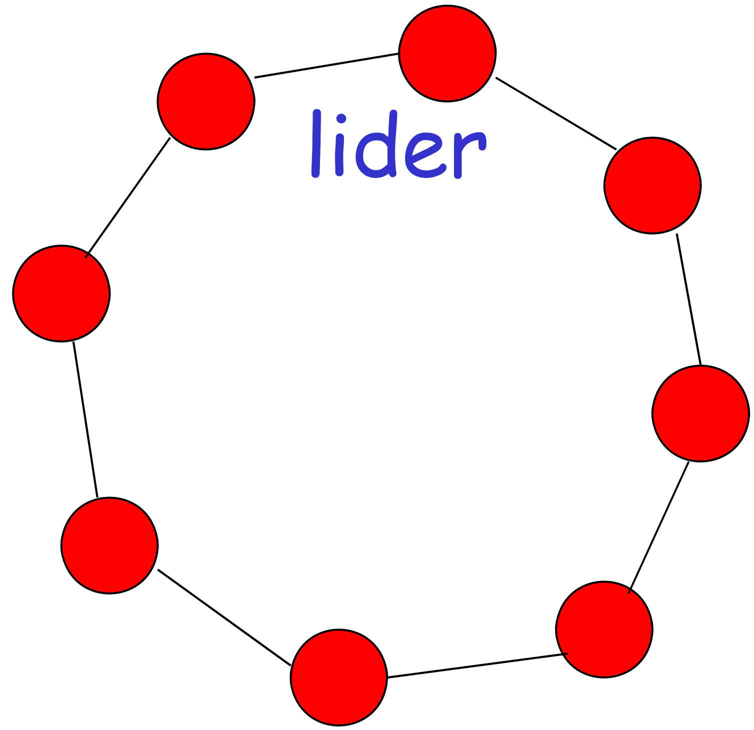
Svaki procesor izvršava isti algoritam

Svaki procesor obavlja potpuno istu obradu

Početno stanje



Konačno stanje

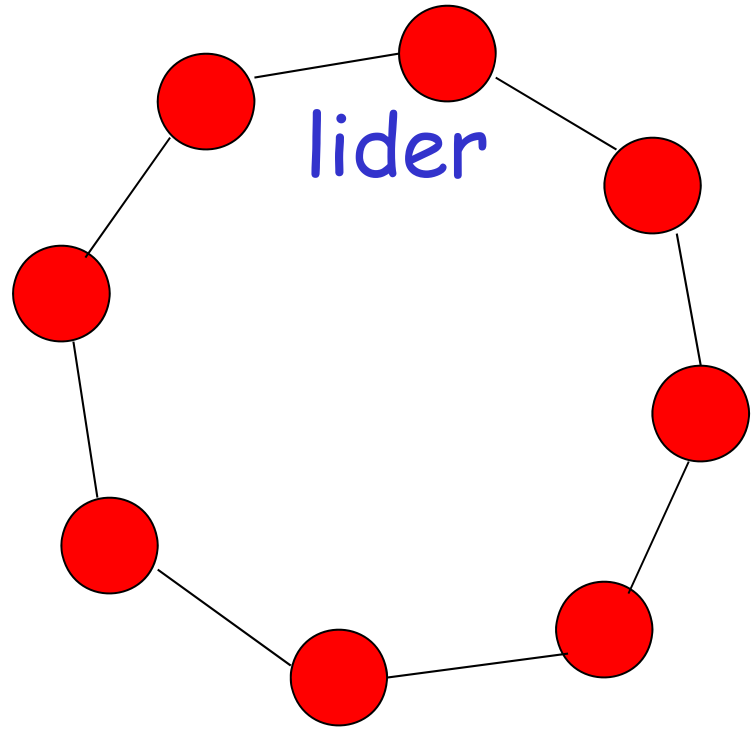


Ako je jedan čvor izabran za lidera,
onda je svaki čvor izabran za lidera

Zaključak 1:

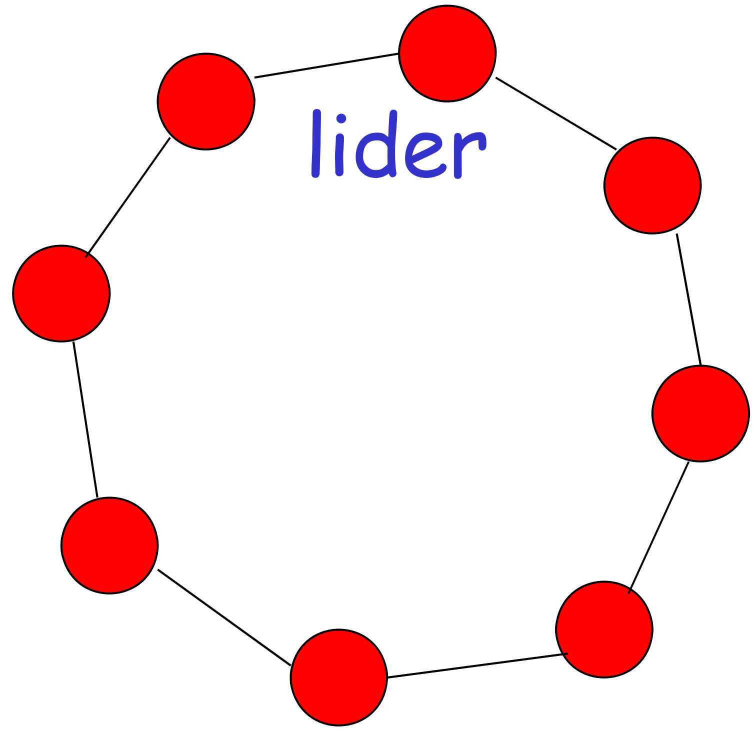
Izbor lidera
se ne može rešiti
u sinhronim
anonimnim prstenima

Konačno stanje



Zaključak 2:

Konačno stanje



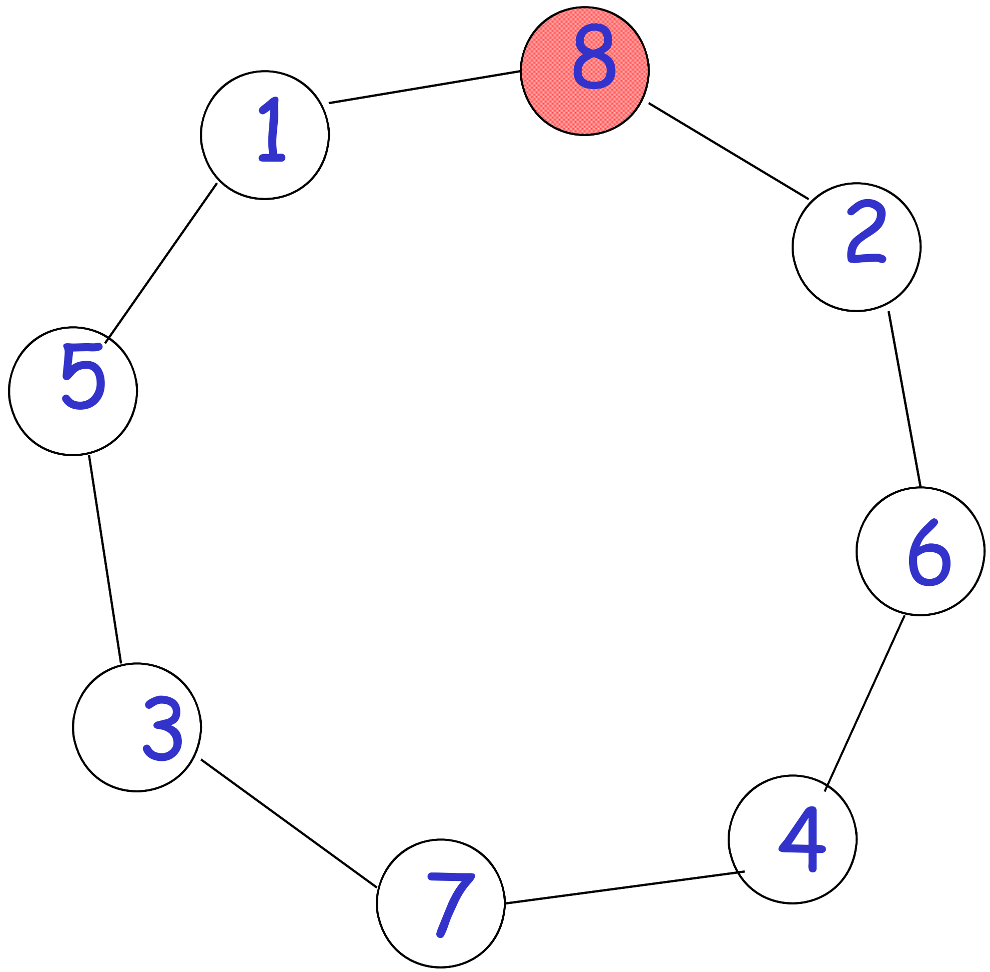
Izbor lidera
ne može se rešiti
ni u <u>asinhronim</u>
anonimnim prstenima

Zašto?

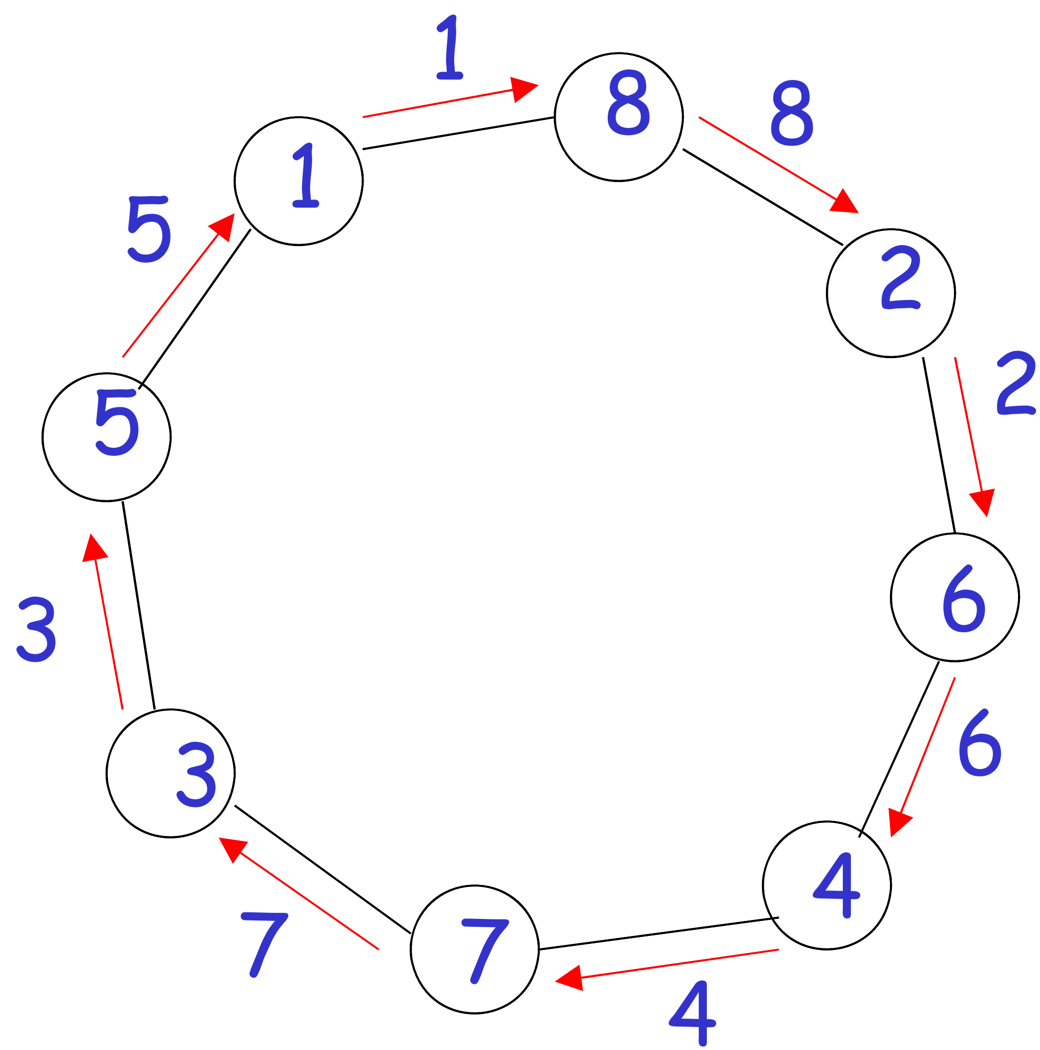
Asinhroni prsten se može ponašati
kao sinhroni prsten

Asinhroni eponimni prsteni

Čvor sa max id se izabira za lidera

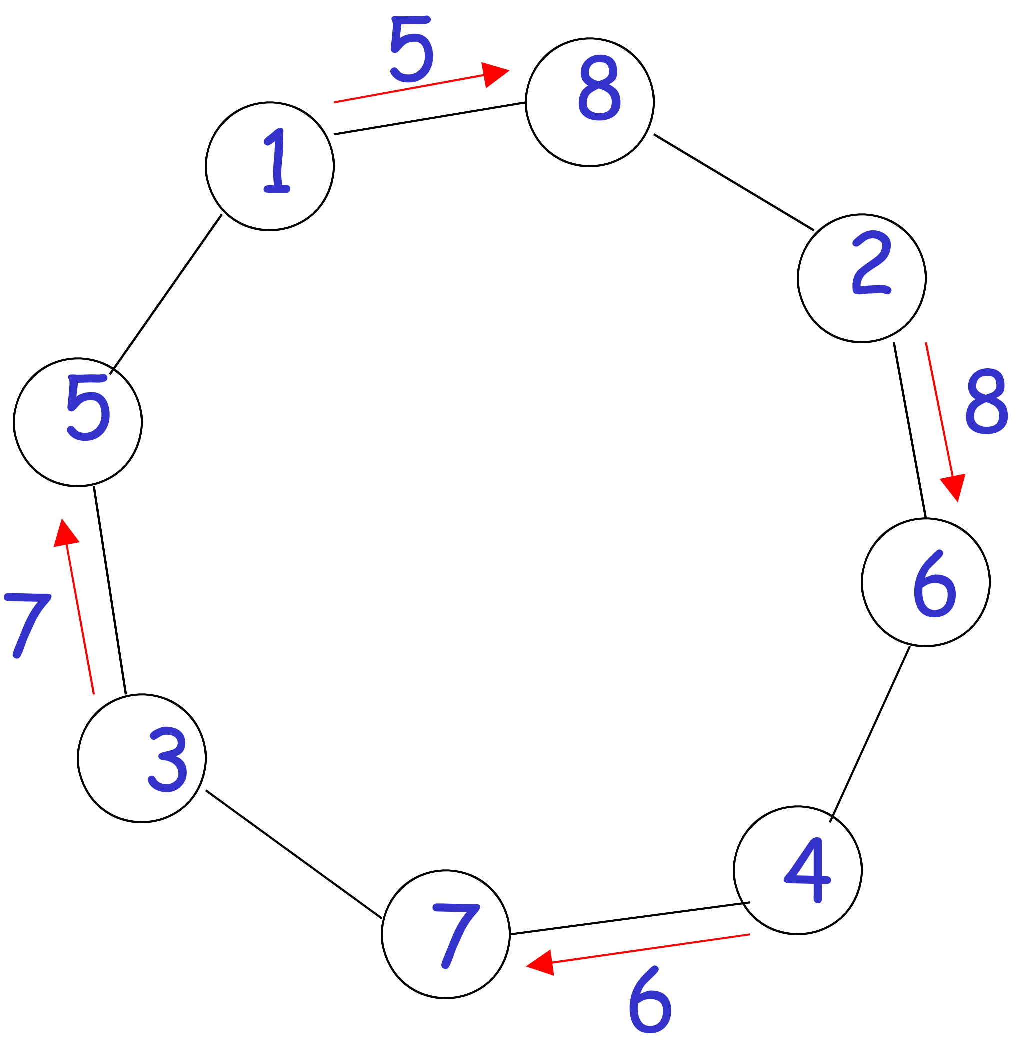


Svaki čvor šalje poruku sa svojim id svom levom susedu



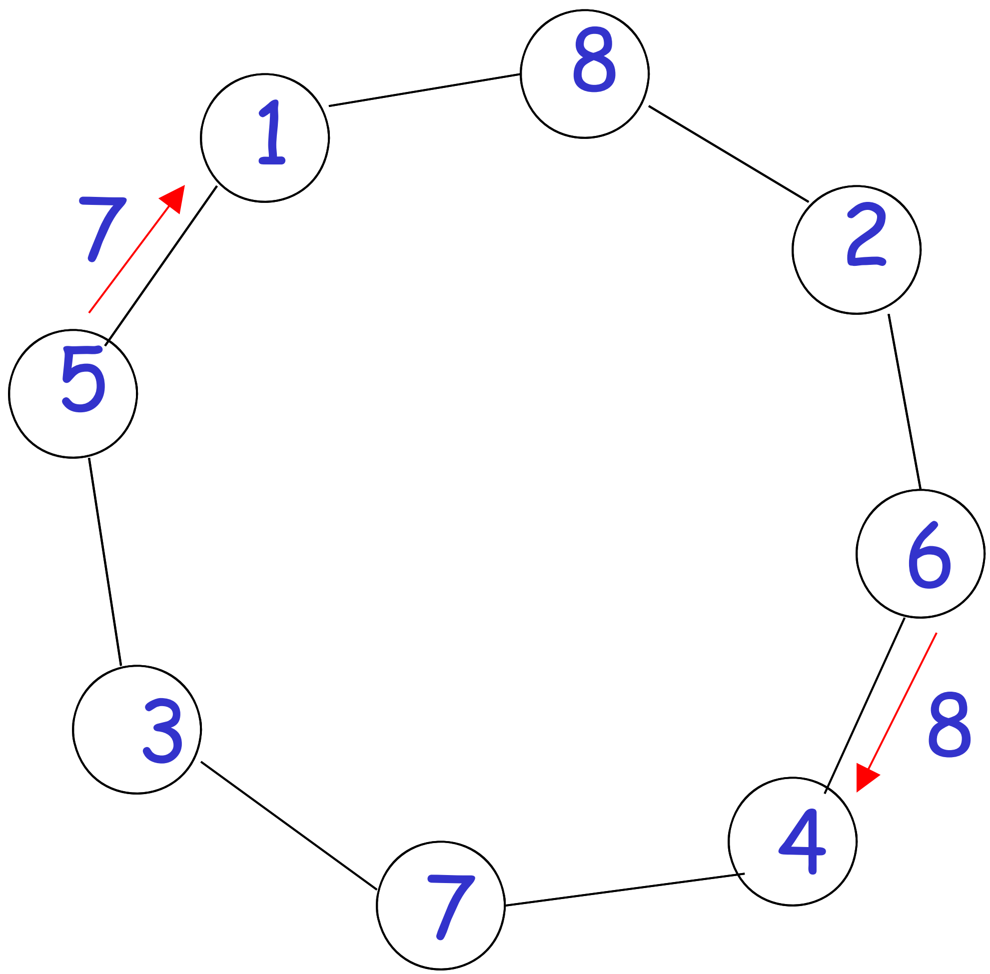
Ako: id u primljenoj poruci > id tekućeg čvora

Onda: prosledi poruku



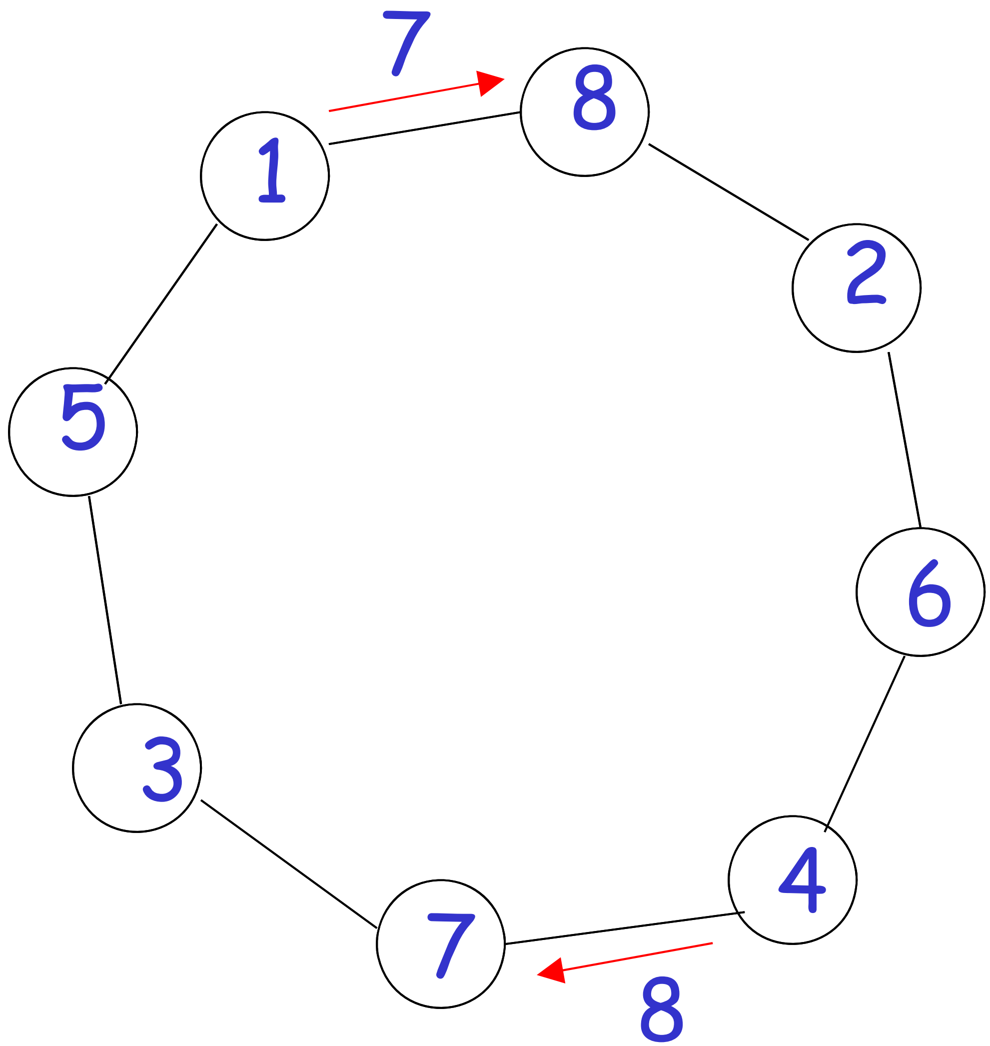
Ako: id u primljenoj poruci > id tekućeg čvora

Onda: prosledi poruku



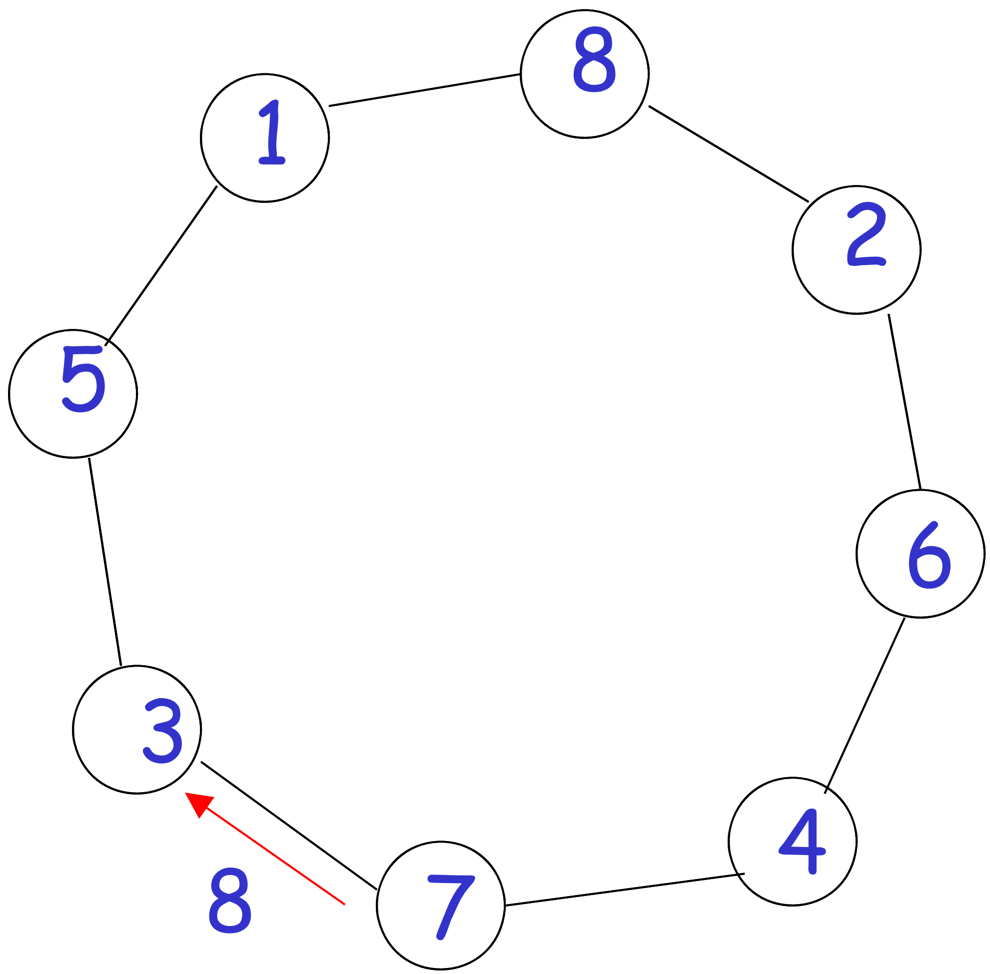
Ako: id u primljenoj poruci > id tekućeg čvora

Onda: prosledi poruku



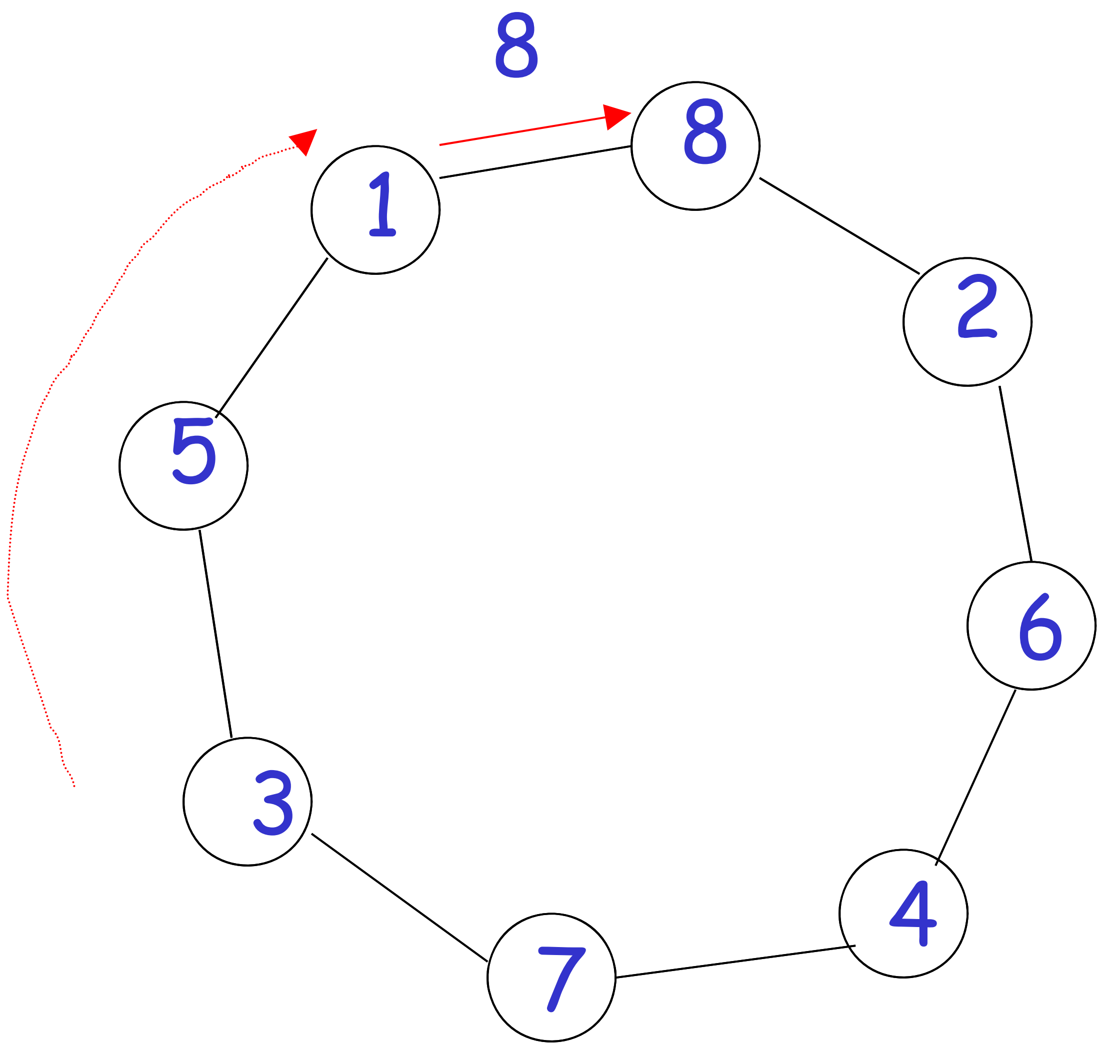
Ako: id u primljenoj poruci > id tekućeg čvora

Onda: prosledi poruku



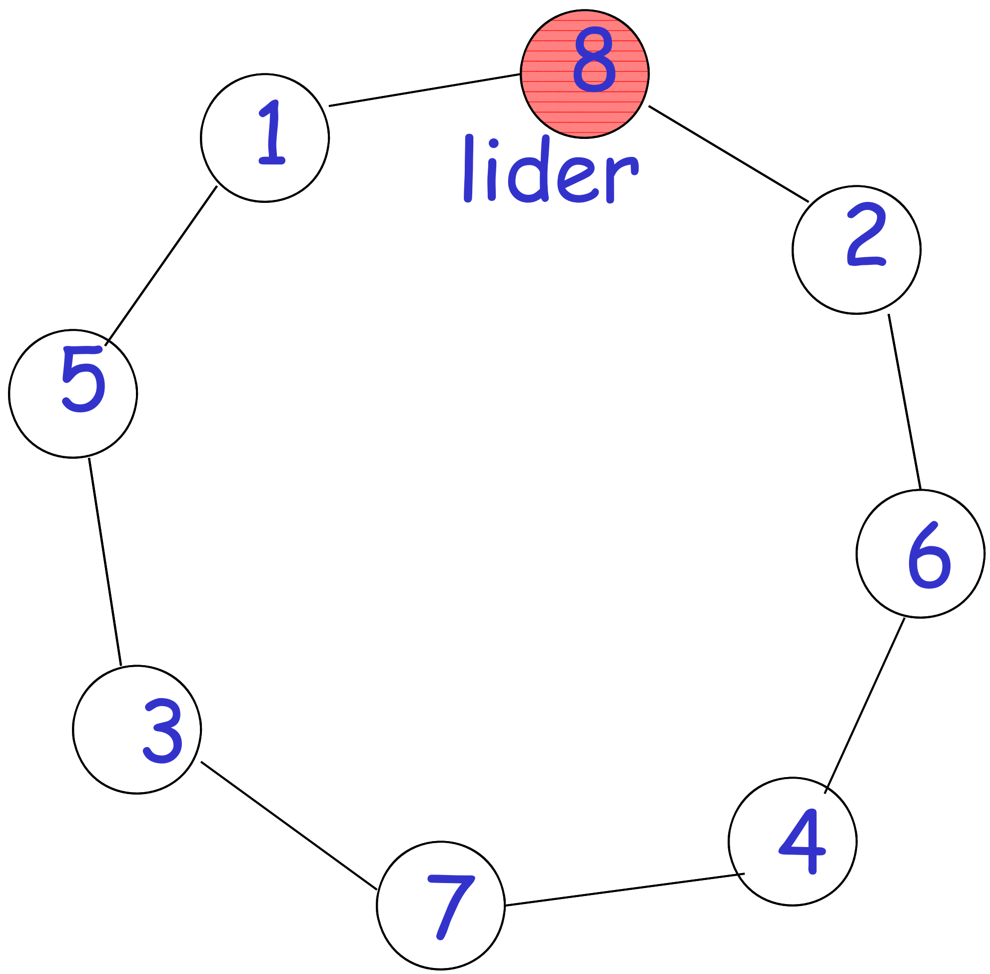
Ako: čvor primi svoju spostvenu poruku

Onda: on izabira sebe za lidera

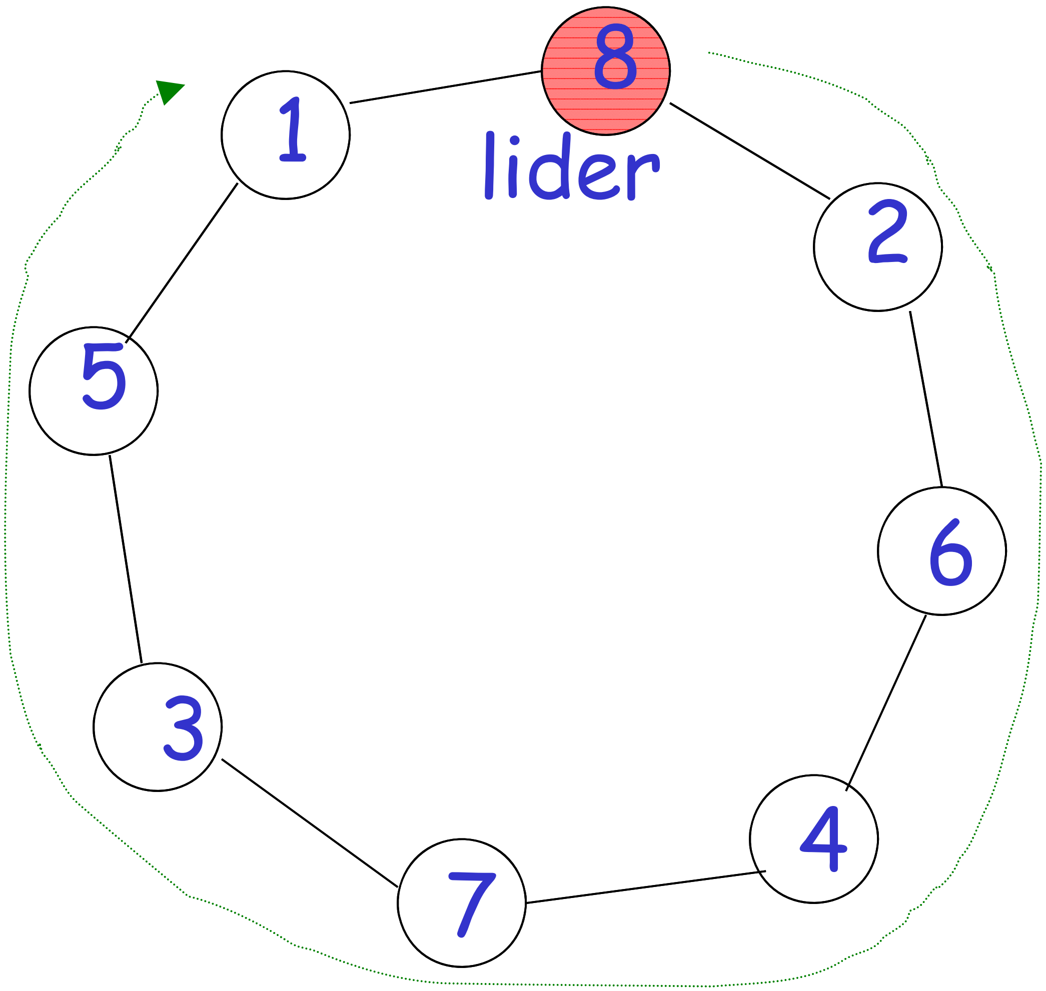


Ako: čvor primi svoju spostvenu poruku

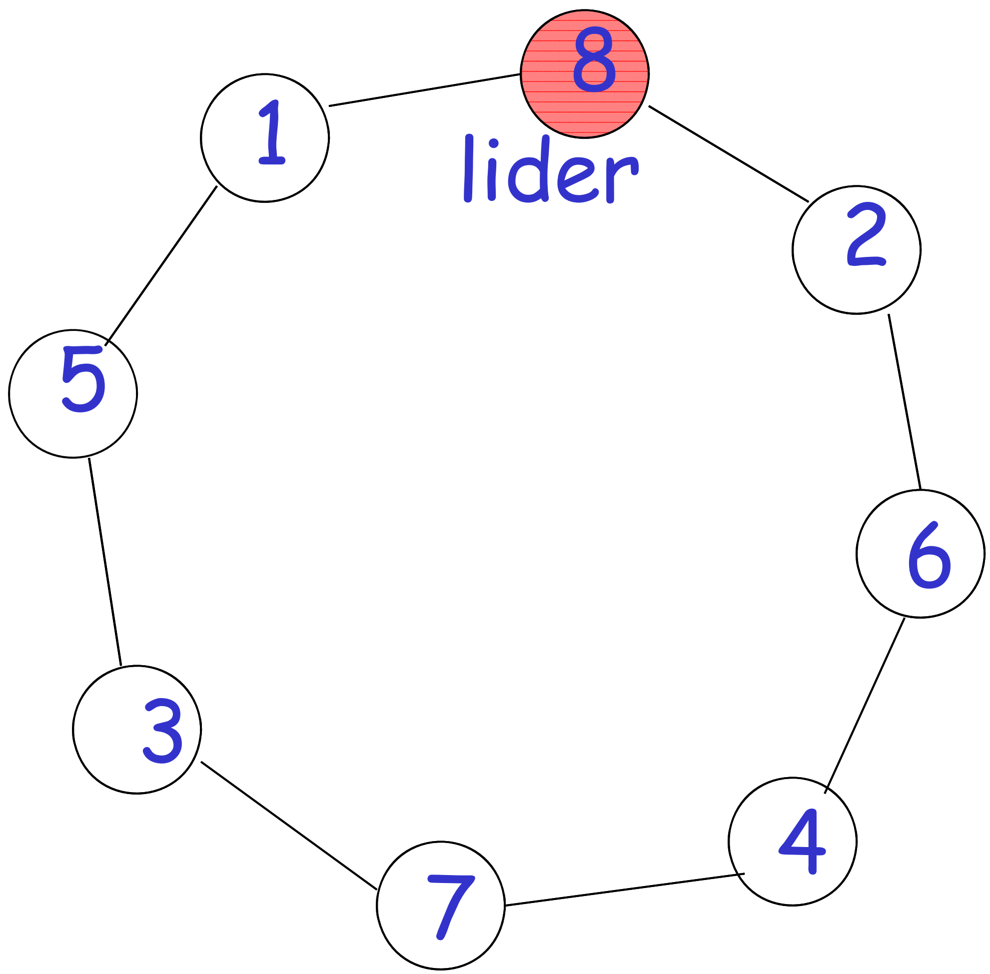
Onda: on izabira sebe za lidera



Lider šalje poruku kroz mrežu deklarišući sebe za "lidera prstena"

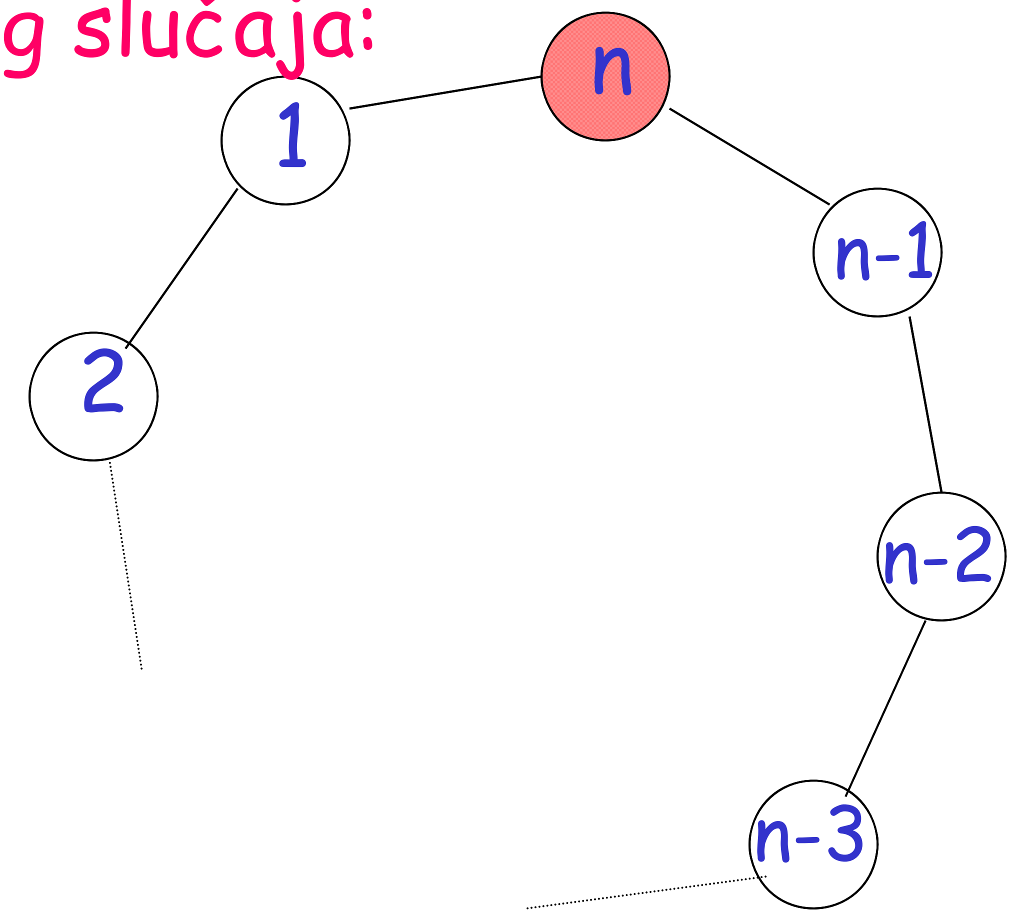


Vreme izvršenja: $O(n)$ n čvorova



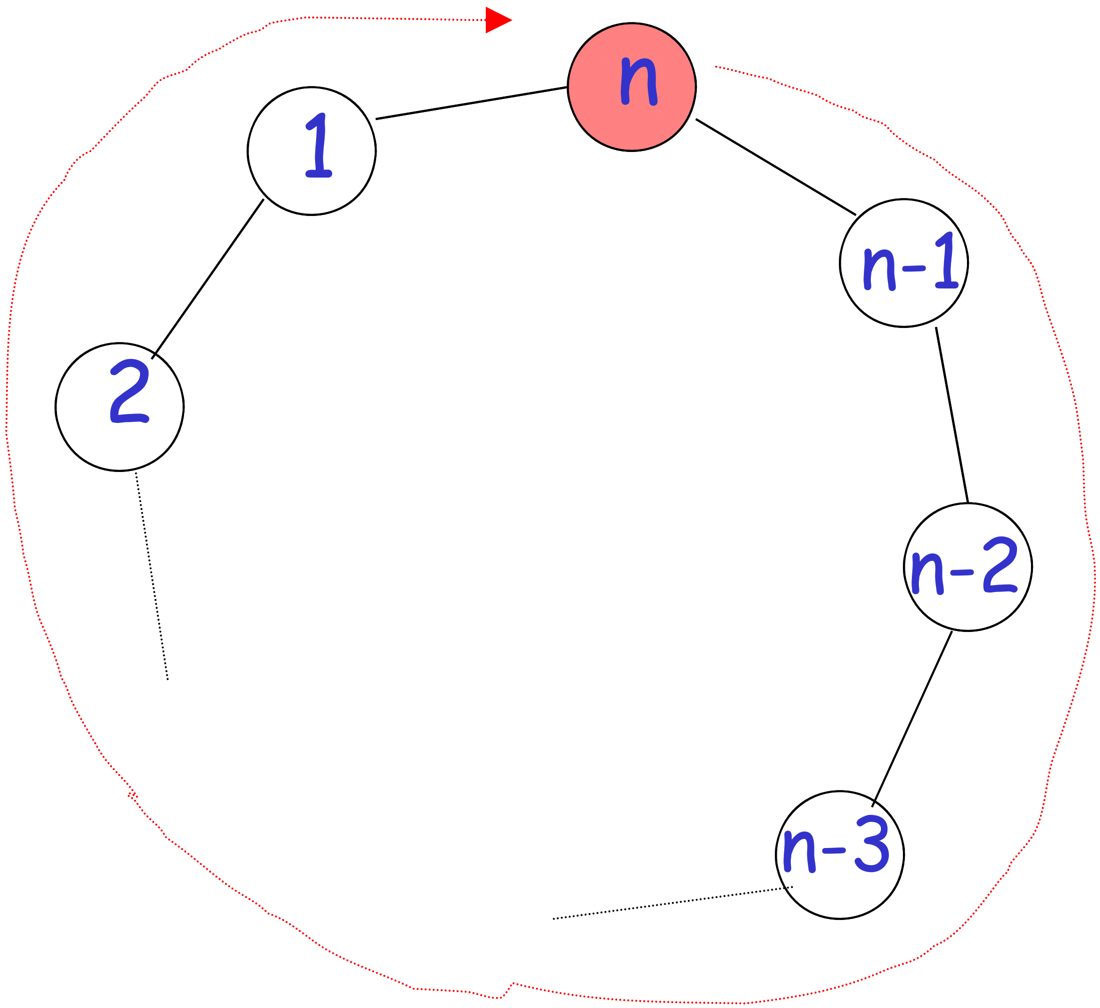
Broj poruka: $O(n^2)$ n čvorova

scenario najgoreg slučaja:



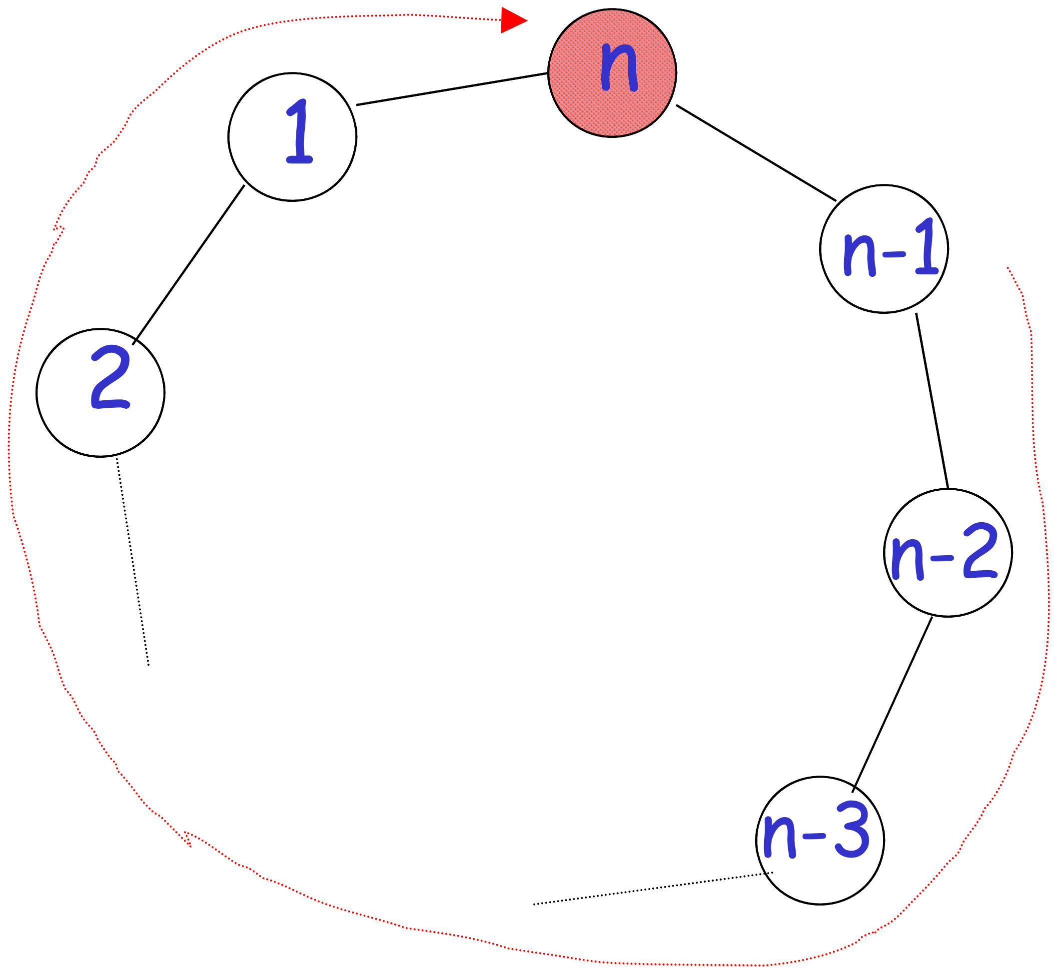
Broj poruka: $O(n^2)$ n čvorova

n poruka



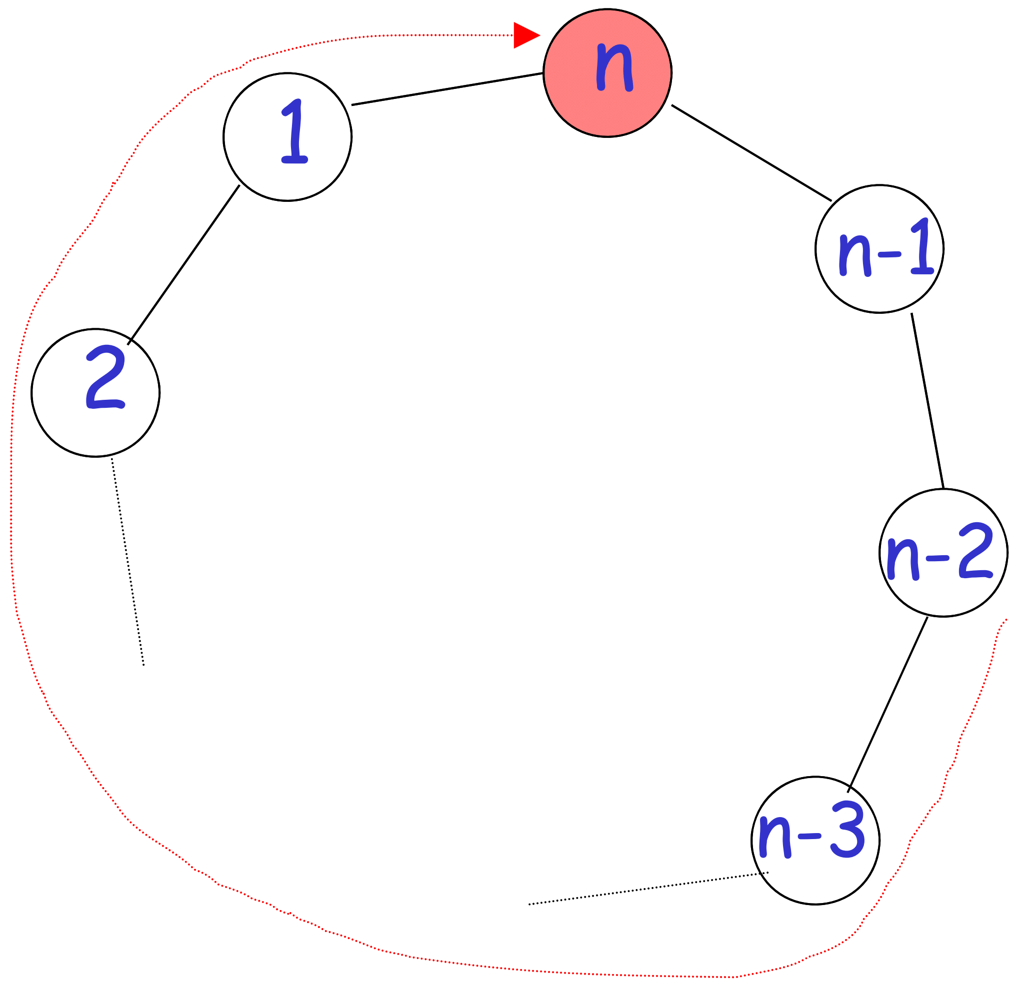
Broj poruka: $O(n^2)$ n čvorova

$n-1$ poruka



Broj poruka: $O(n^2)$ n čvorova

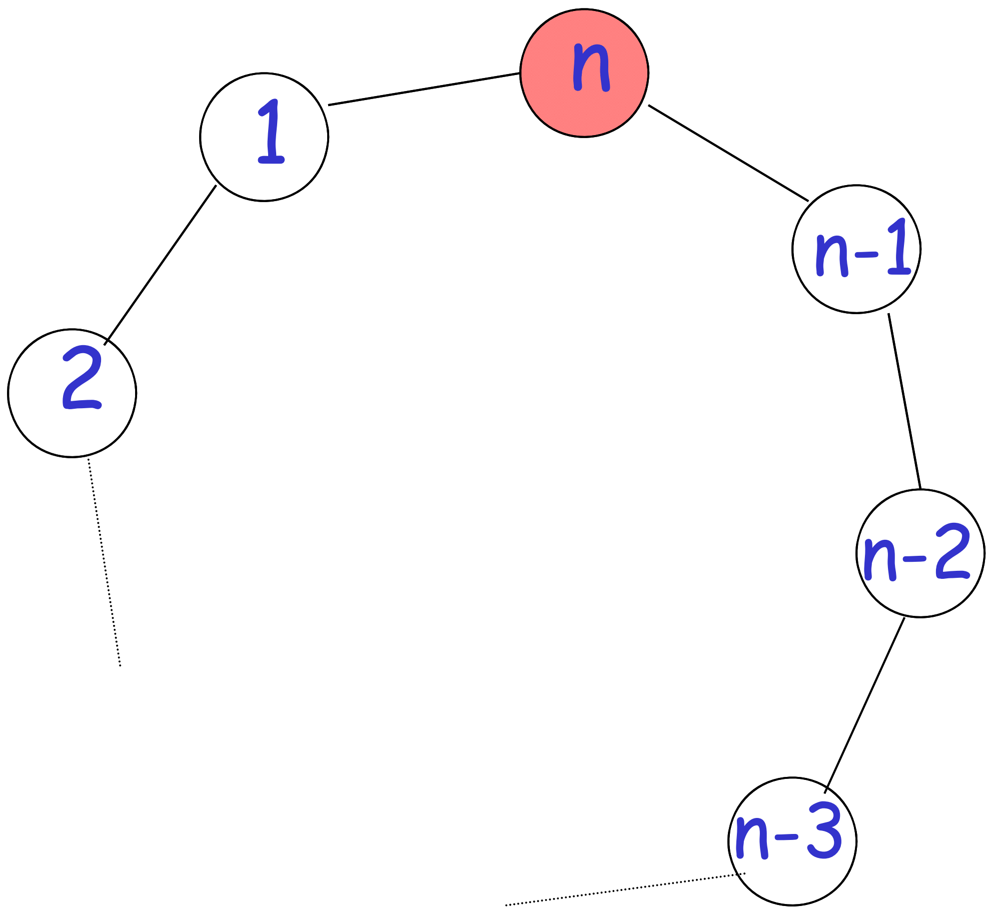
$n-2$ poruka



Broj poruka: $O(n^2)$ n čvorova

Ukupno poruka:

$$n+$$
$$n-1+$$
$$n-2+$$
$$\cdots$$
$$2+$$
$$1=\Omega(n^2)$$



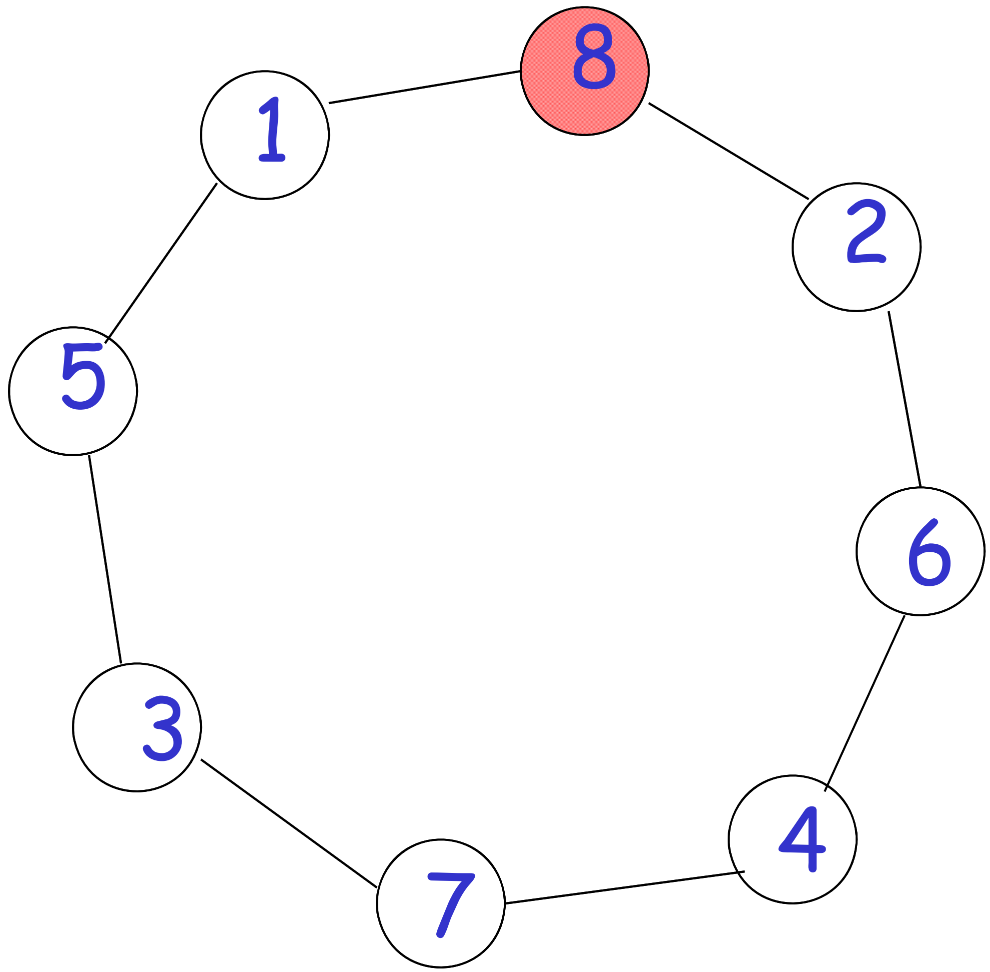
Napomene:

n ne mora biti poznato algoritmu

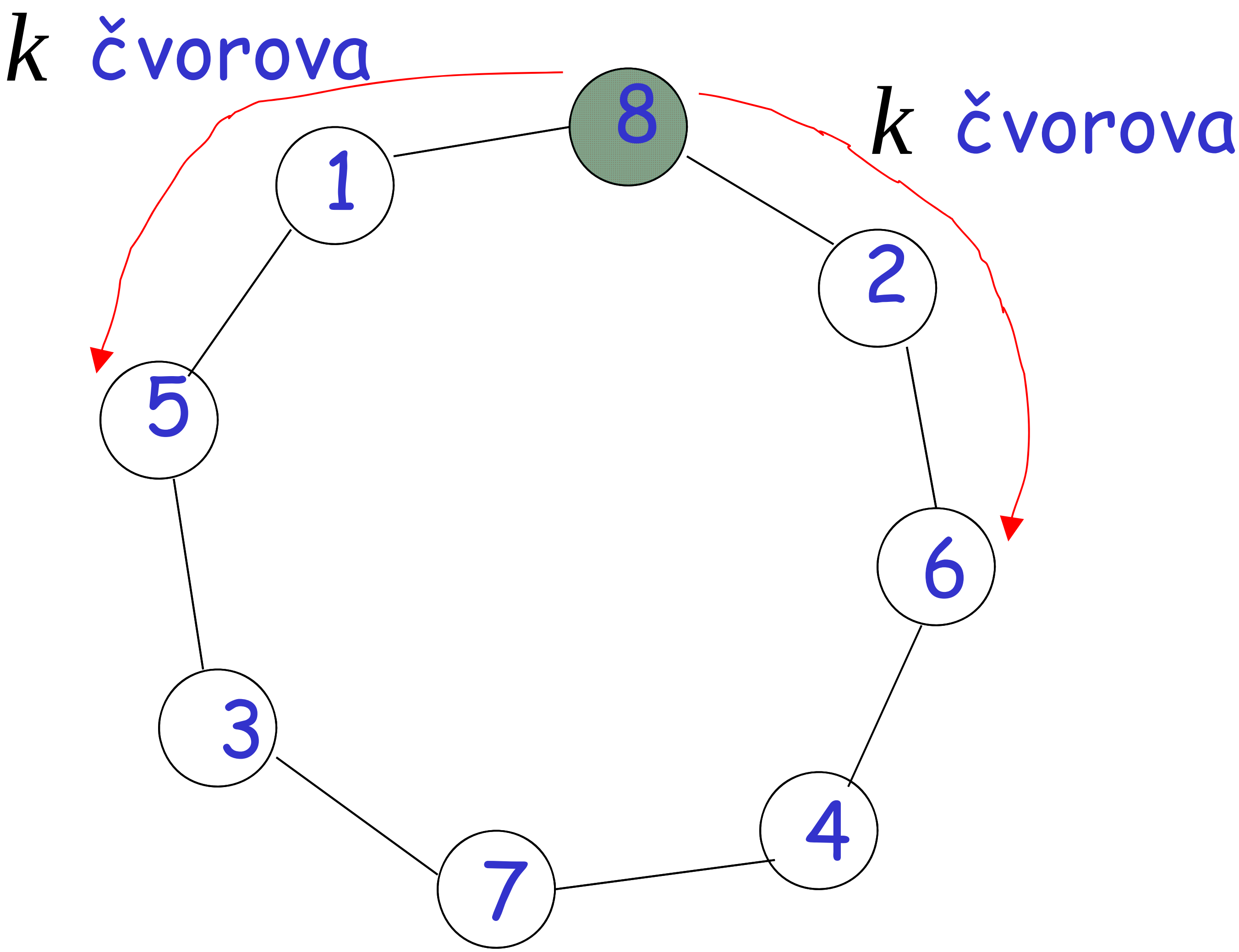
Algoritam se može pretvoriti u asinhroni

Algoritam sa O(n log n) poruka

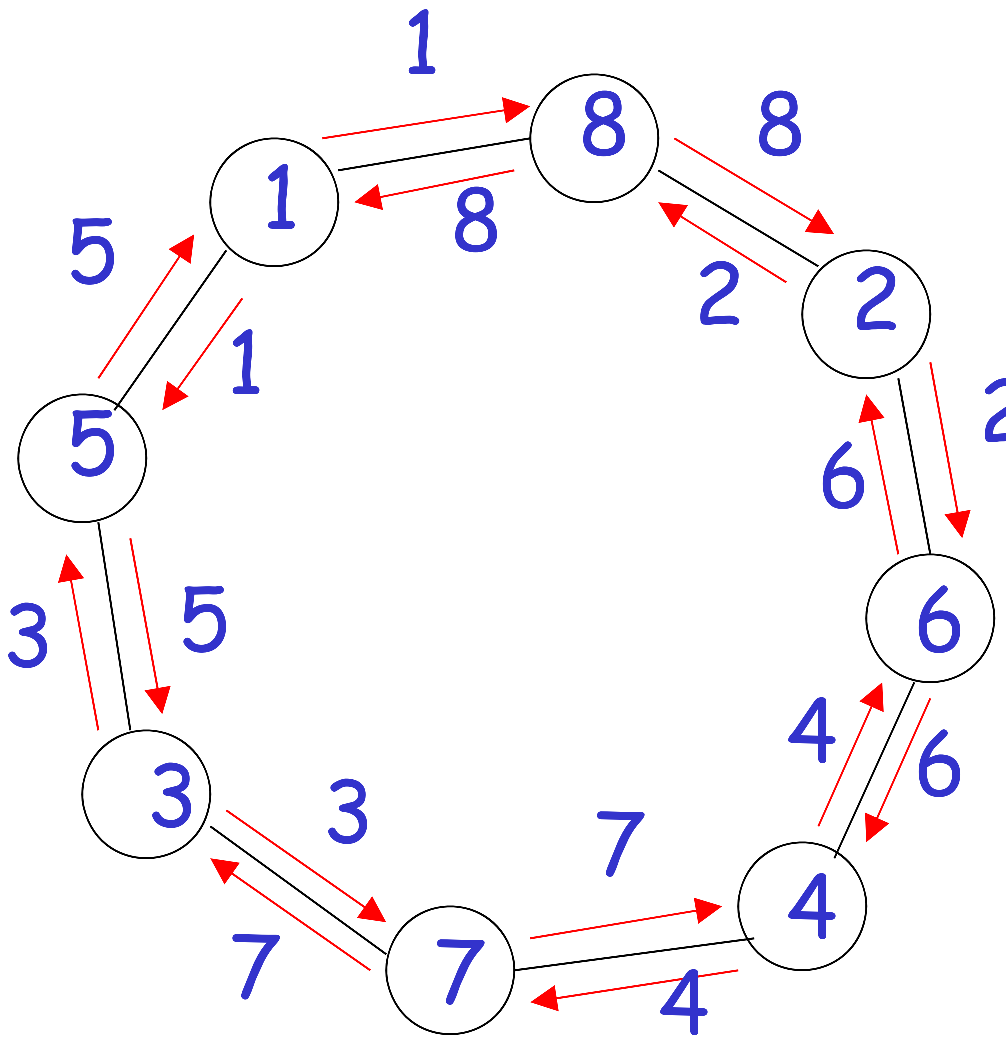
Ponovo se izabira za lidera čvor sa max id



k-komšiluk

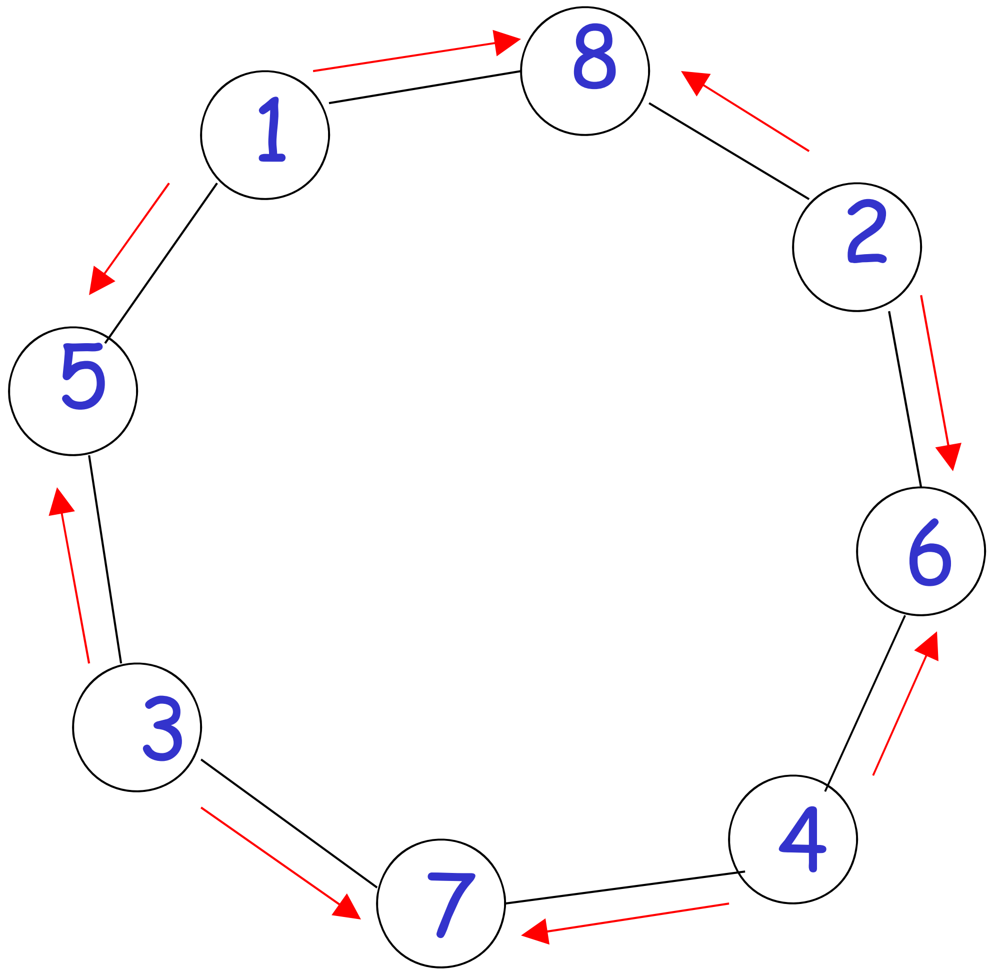


Faza 1: pošalji id u 1-komšiluk



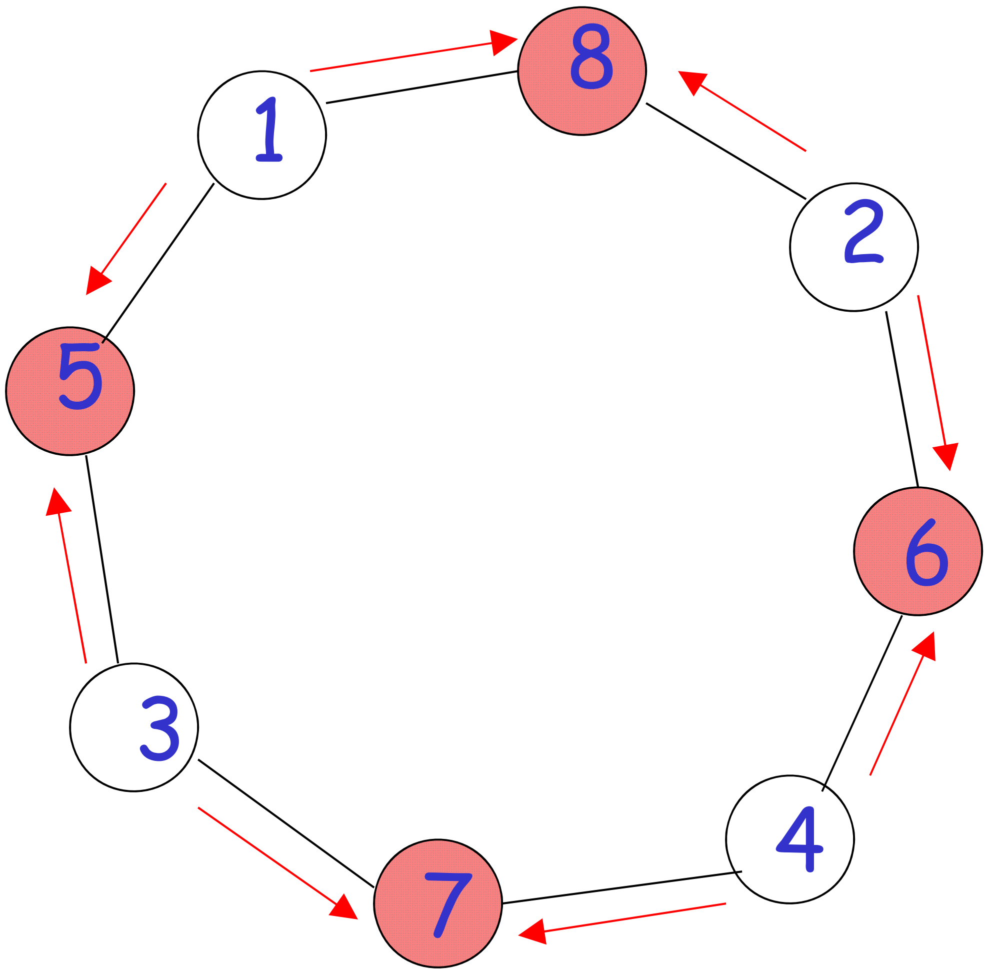
Ako: primljen id > tekući id

Onda: pošalji odgovor

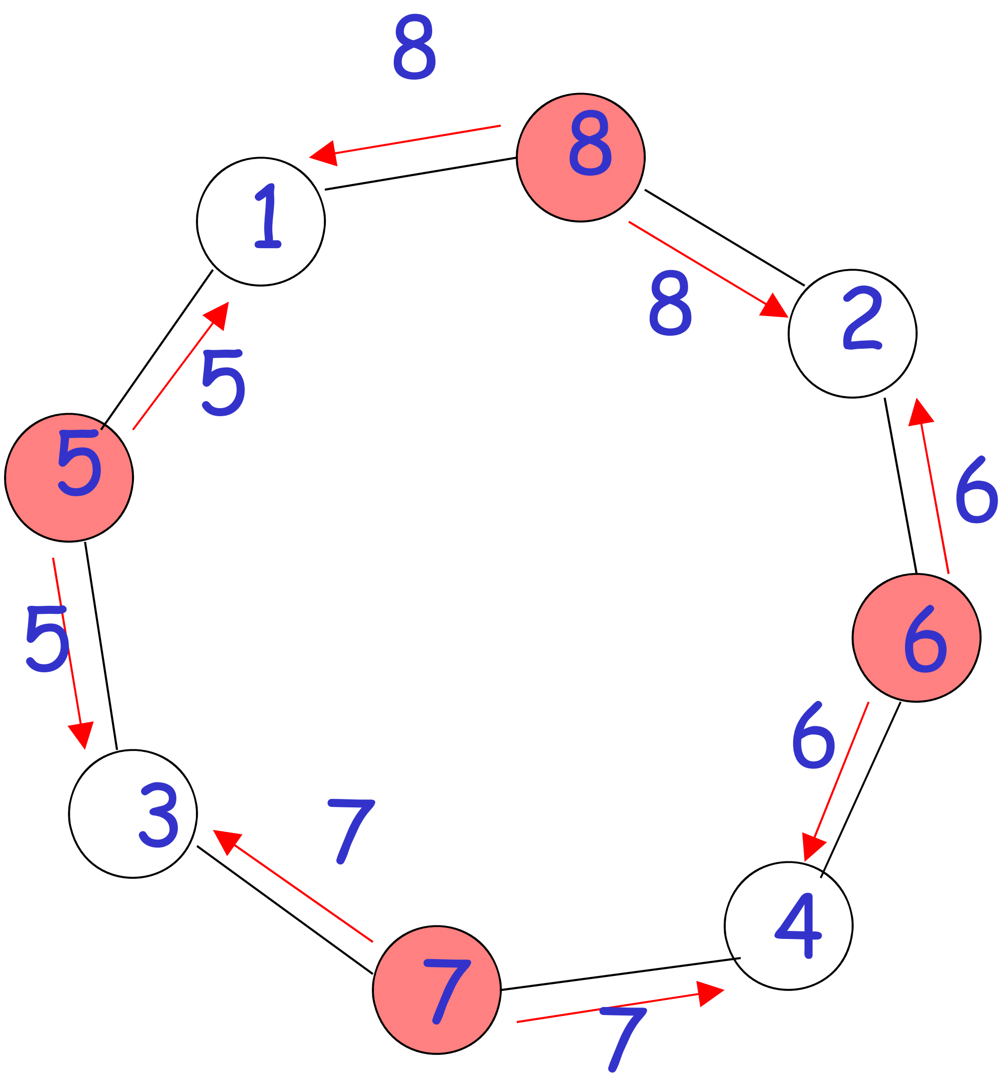


Ako: čvor primi oba odgovora

Onda: on postaje privremeni lider

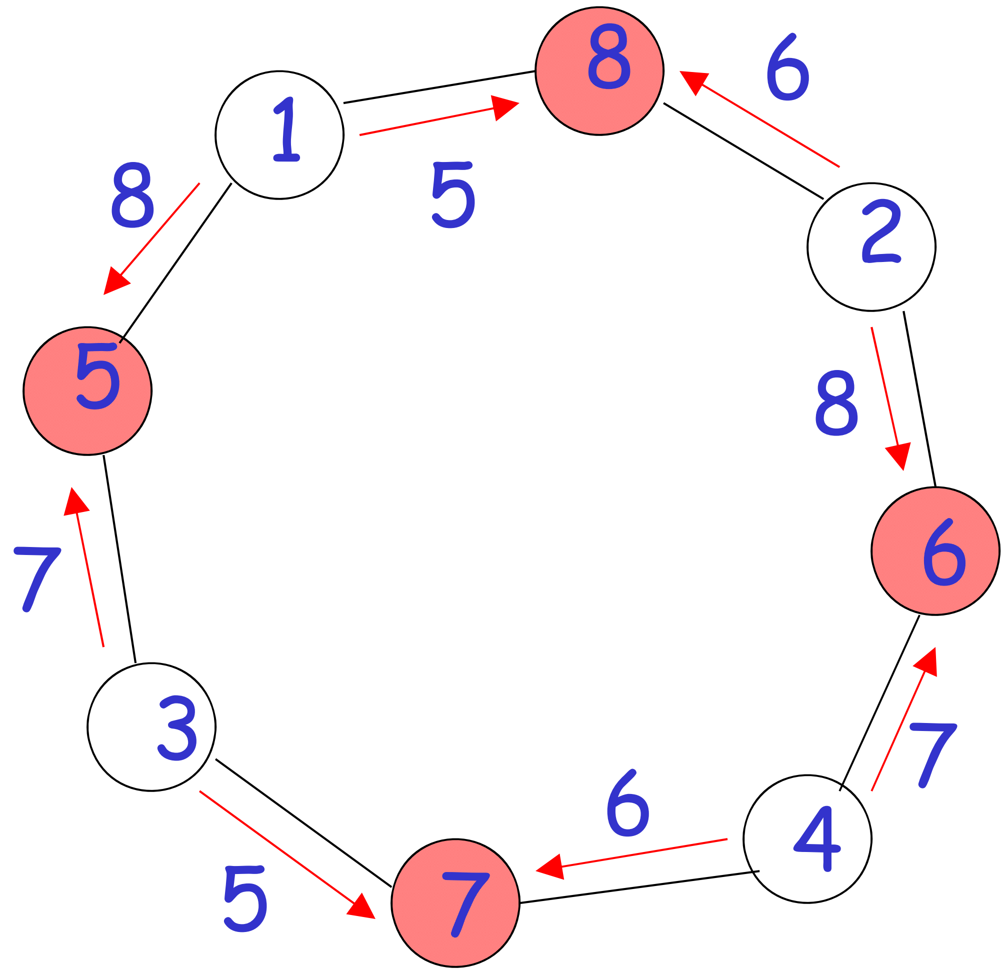


Faza 2: pošalji id u 2-komšiluk



Ako: primljen id > tekući id

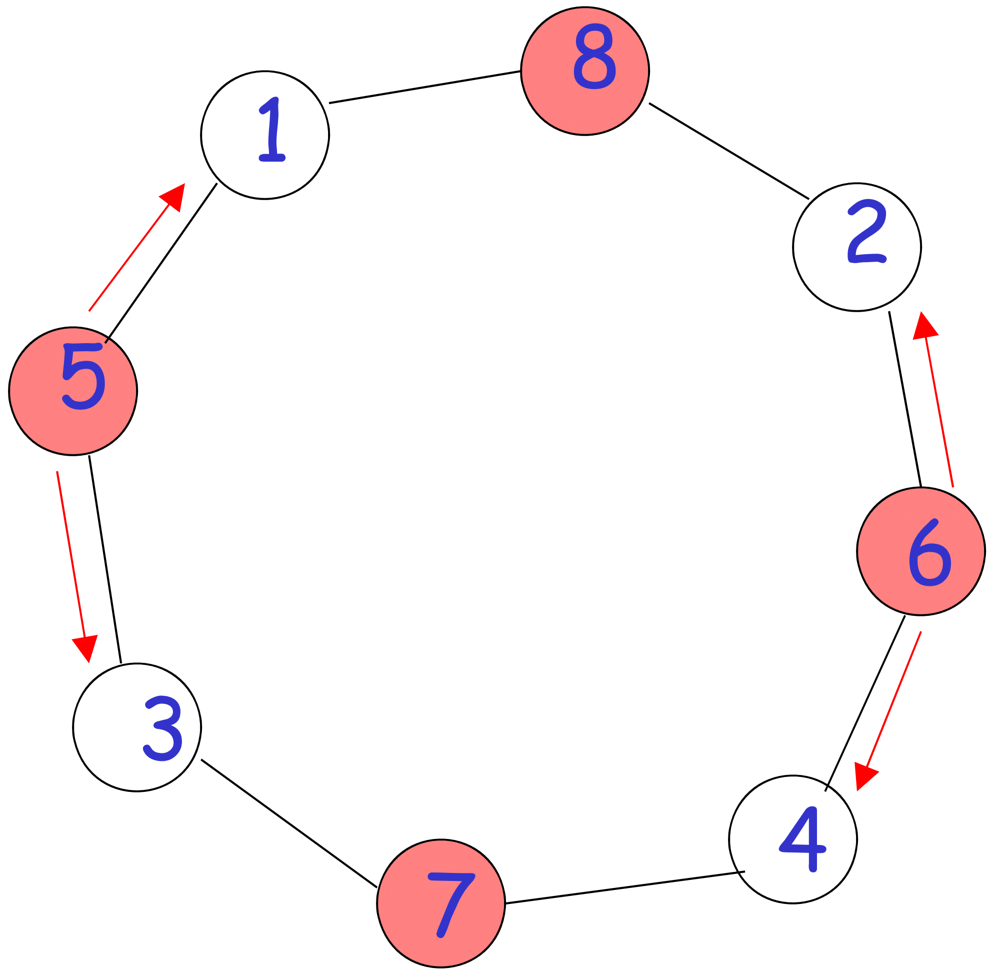
Onda: prosledi poruku



U drugom koraku:

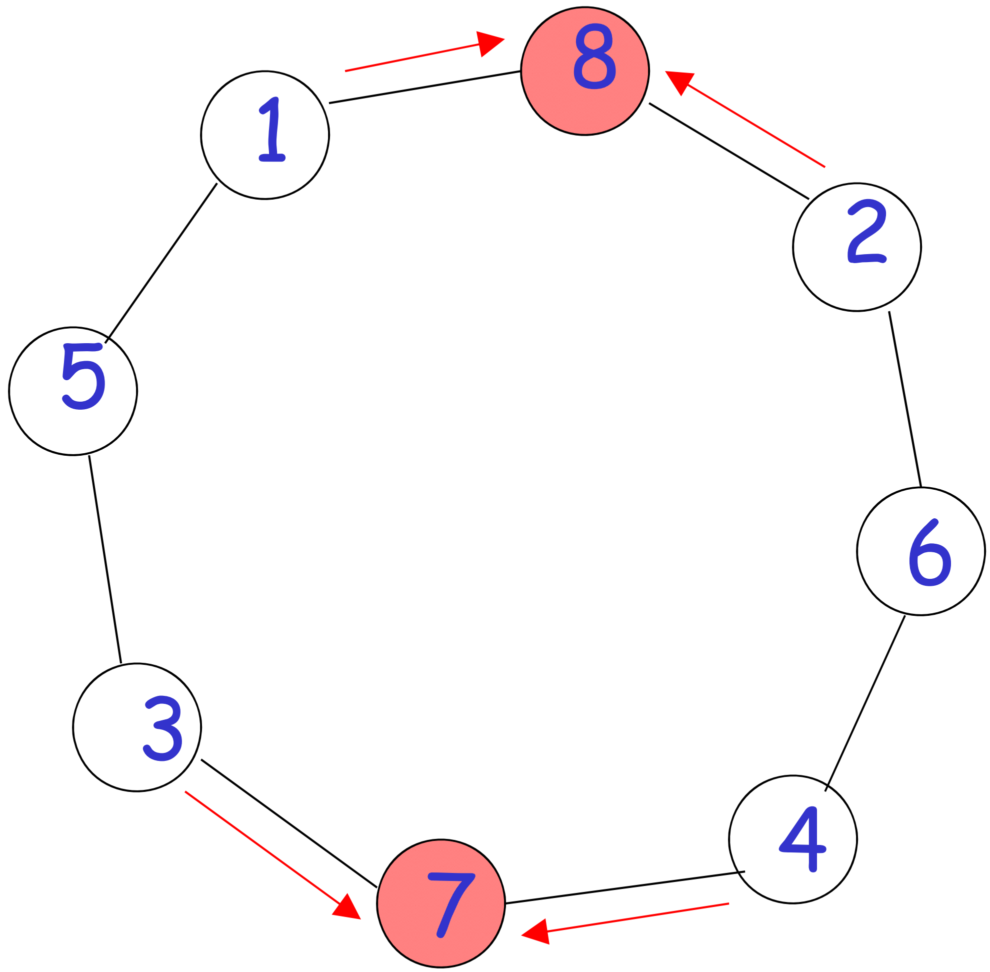
Ako: primljen id > tekući id

Onda: pošalji odgovor

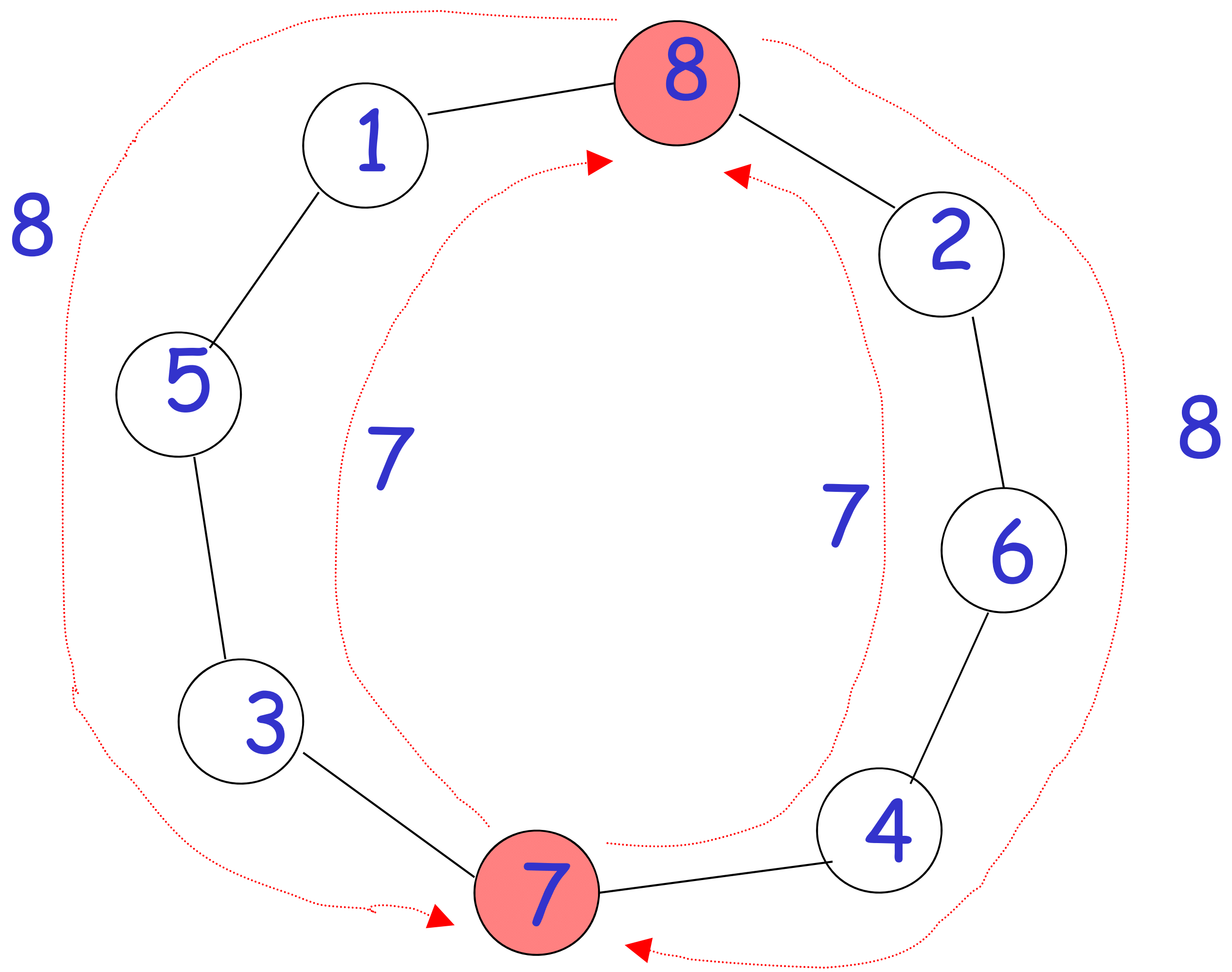


Ako: čvor primi oba odgovora

Onda: on postaje privremeni lider

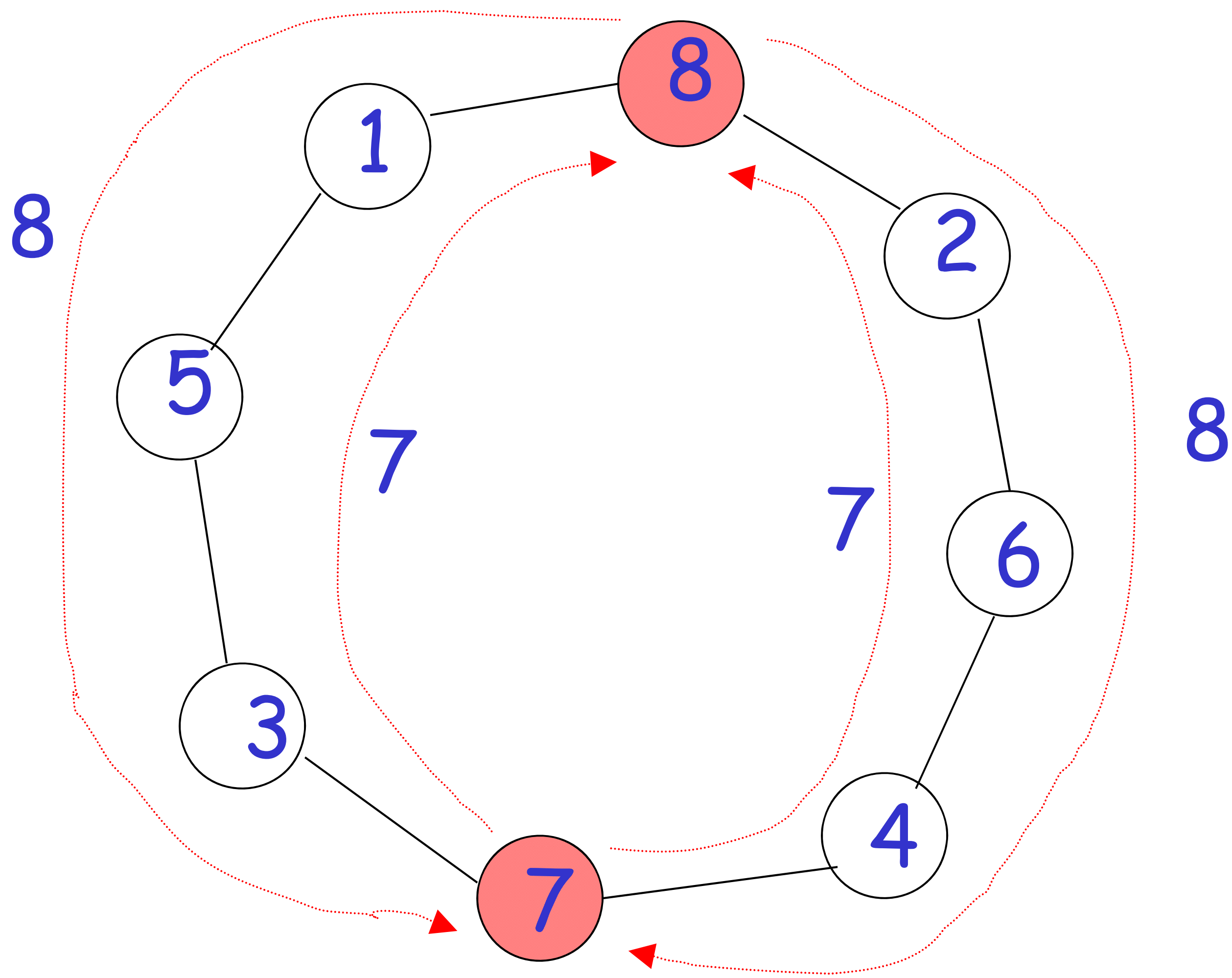


Faza 3: pošalji id u 2^2-komšiluk



Ako: primljen id > tekući id

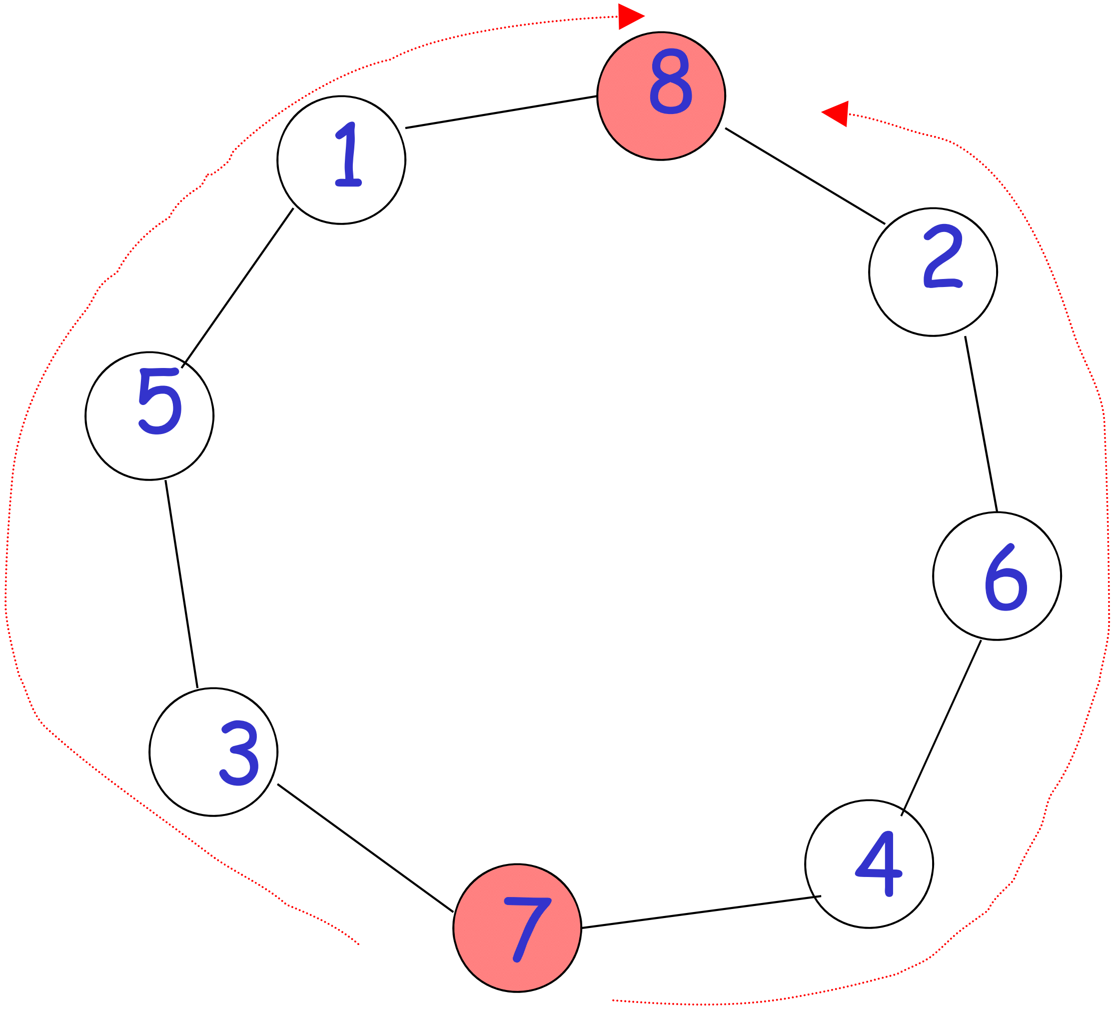
Onda: prosledi poruku



U 2^2 koraku:

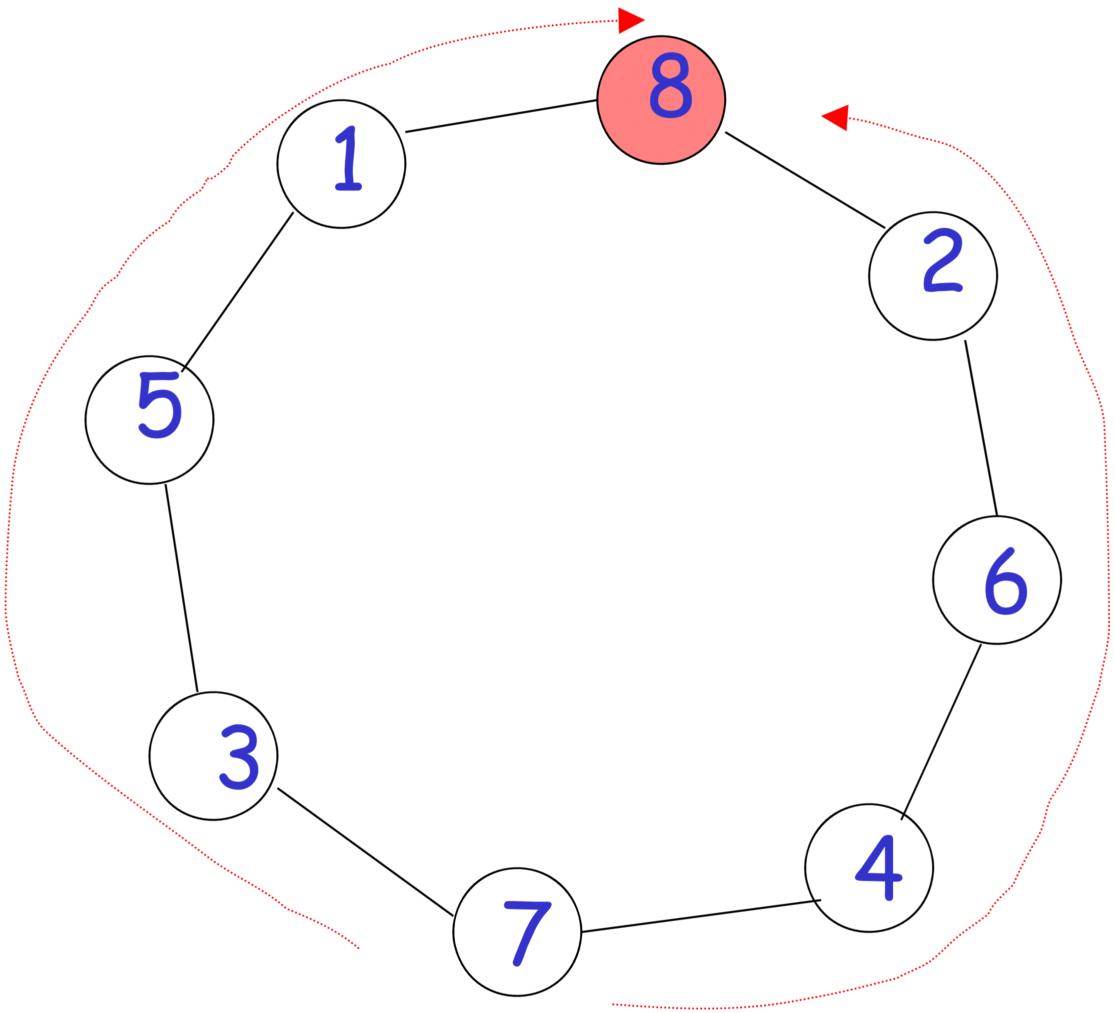
Ako: primljen id > tekući id

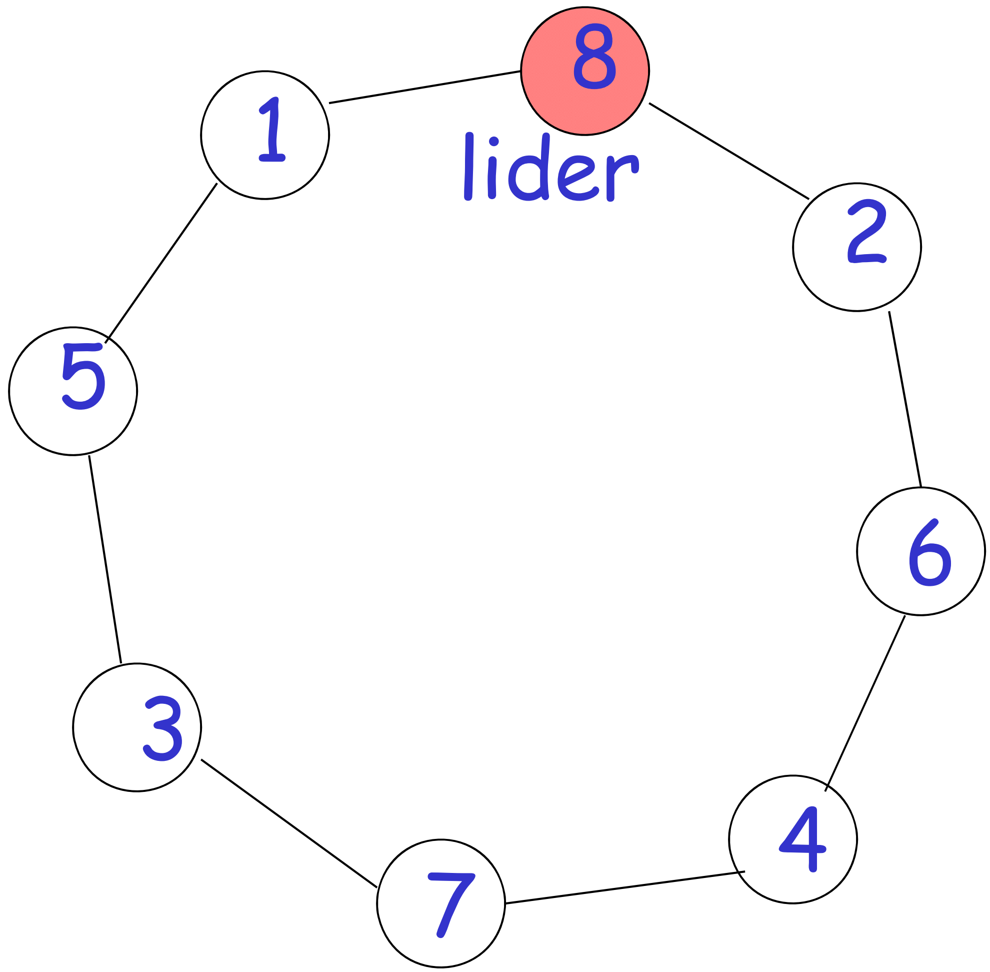
Onda: pošalji odgovor



Ako: čvor primi oba odgovora

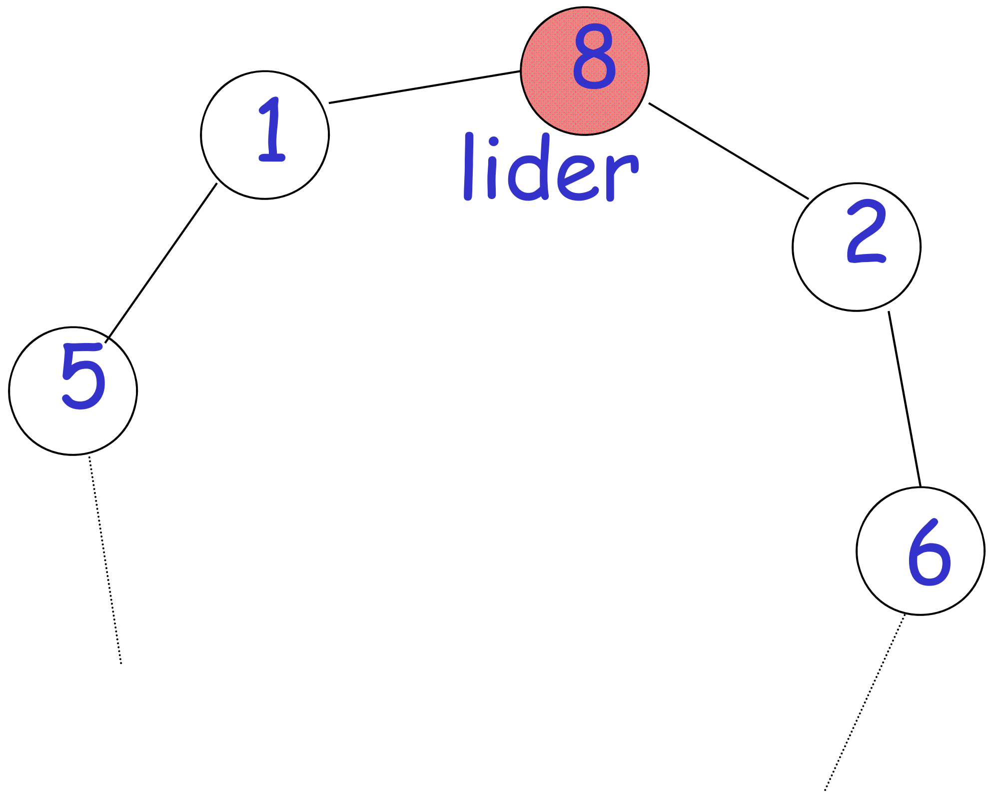
Onda: on postaje lider



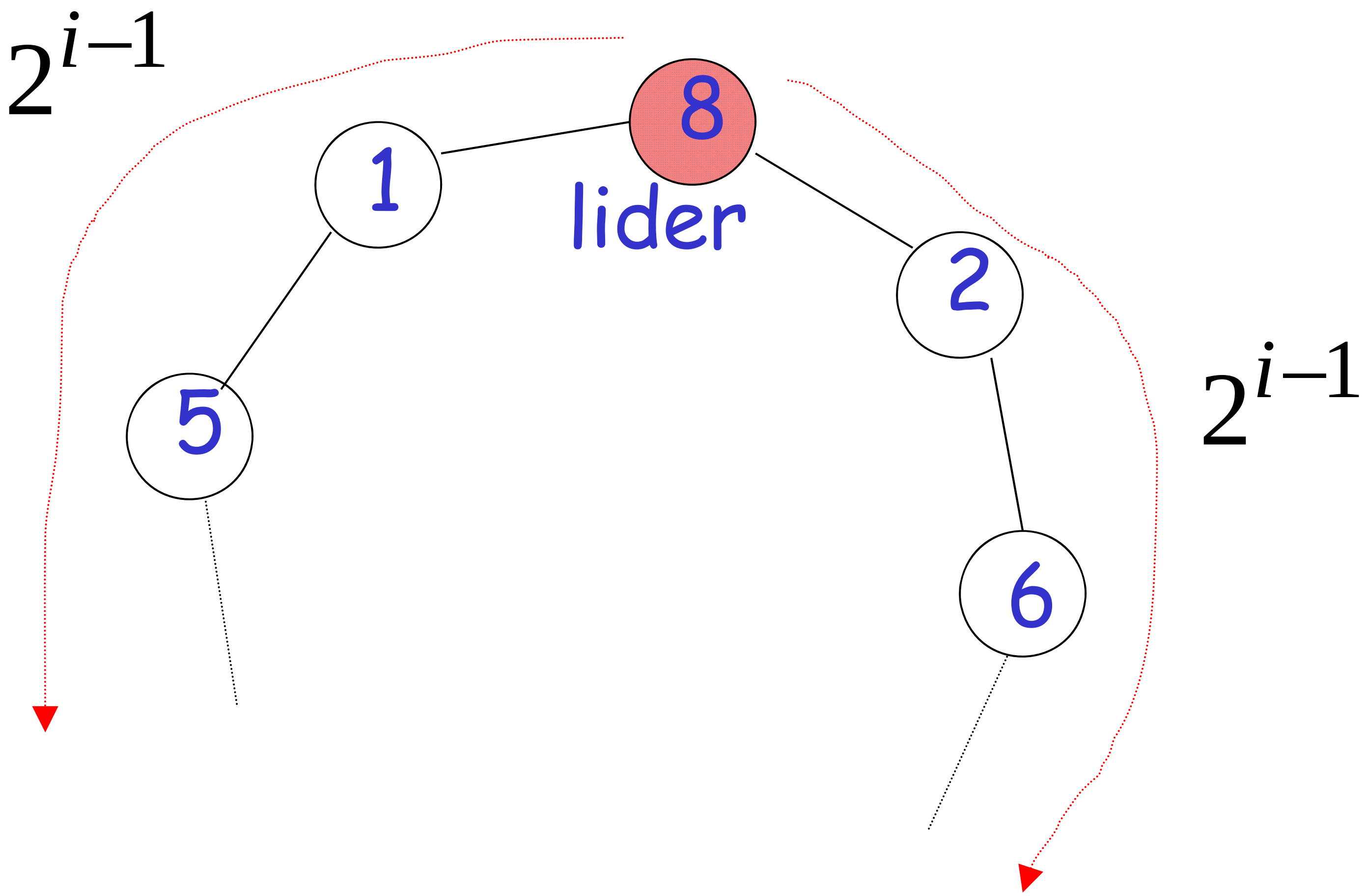

8
lider
1
2
5
6
3
4
7

U opštem slučaju:

n čvorova $\Rightarrow$ $\log n$ faza



Faza i: pošalji id u 2^{i-1}-komšiluk



Vreme izvršenja

Lider troši vreme u

Fazi 1: 2

Fazi 2: 4

...

Fazi i: 2^i

...

Fazi log n: $2^{\log n}$

Uku. vreme: $2^{\log n+1} - 2 = O(2^{\log n}) = O(n)$

Broj poruka

Poruka po lideru	Max #lidera
Faza 1: 4	n
Faza 2: 8	$n/2$
…	
Faza i: 2^{i+1}	$n/2^{i-1}$
…	
Faza log n: $2^{\log n+1}$	$n/2^{\log n-1}$

Poruka po lideru		Max #lidera	
Faza 1: 4	$\times$	n	$= 4n$
Faza 2: 8	$\times$	$n/2$	$= 4n$
… $= 4n$			
Faza i: 2^{i+1}	$\times$	$n/2^{i-1}$	$= 4n$
…			
Faza log n: $2^{\log n+1}$	$\times$	$n/2^{\log n-1}$	$= 4n$

Ukupno poruka: $O(n \cdot \log n)$

Napomene:

Algoritam ne mora da poznaje n

Može se pretvoriti u asinhroni algoritam

Sinhroni algoritam sa $O(n)$ poruka

n je poznato

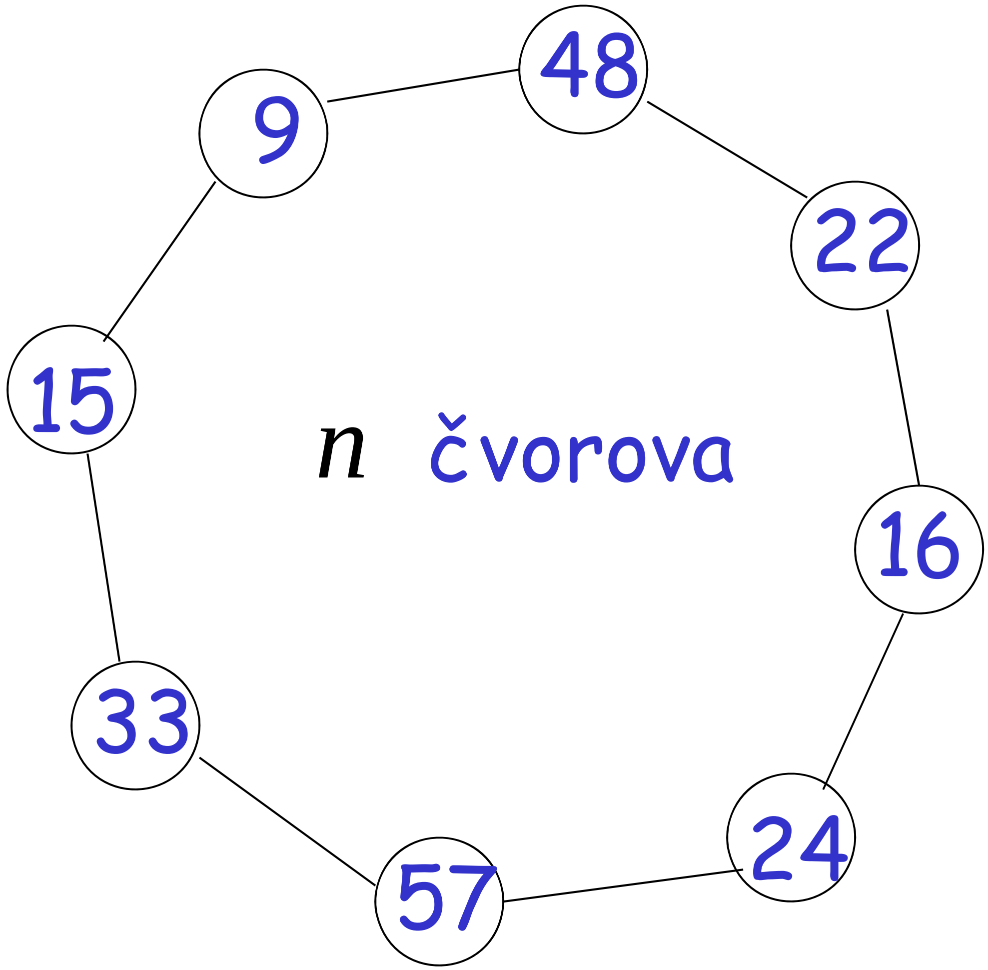
Čvor sa najmanjim id se izabira za lidera

Postoje runde:

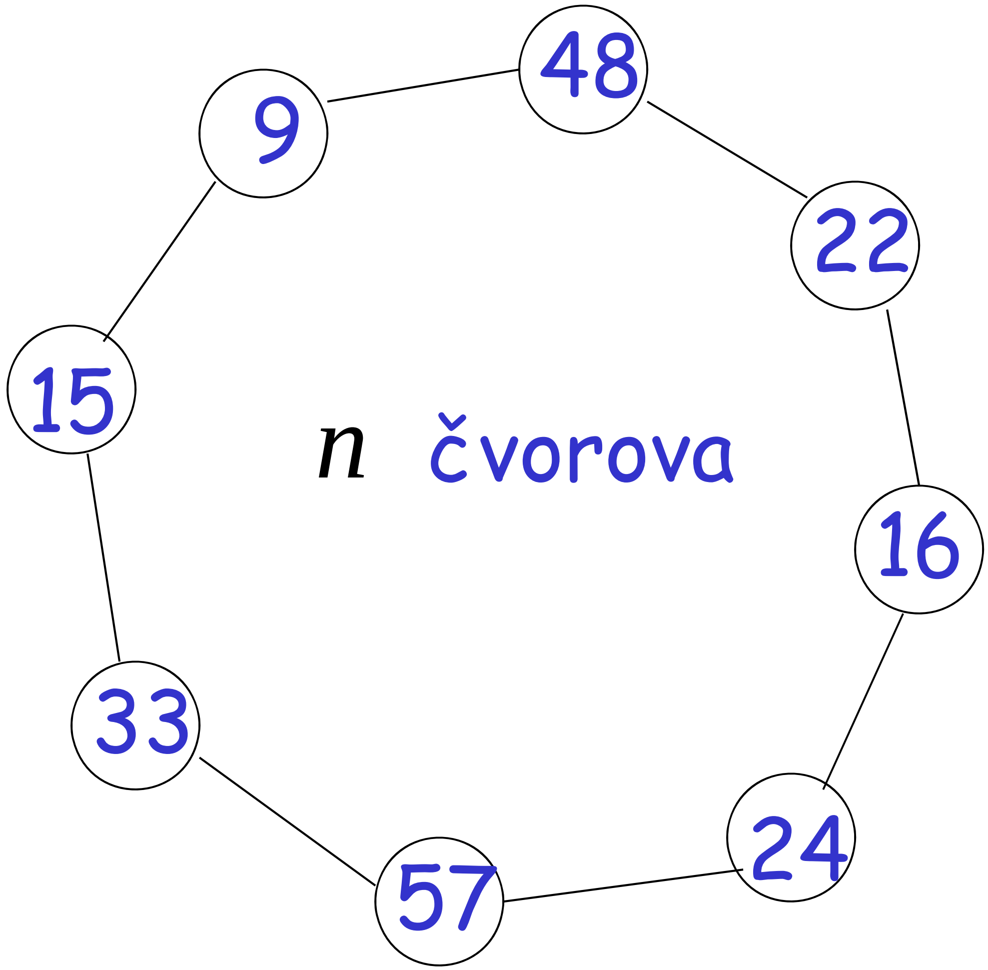
Ako u rundi i postoji čvor sa id i

- on je novi lider
- algoritam se završava

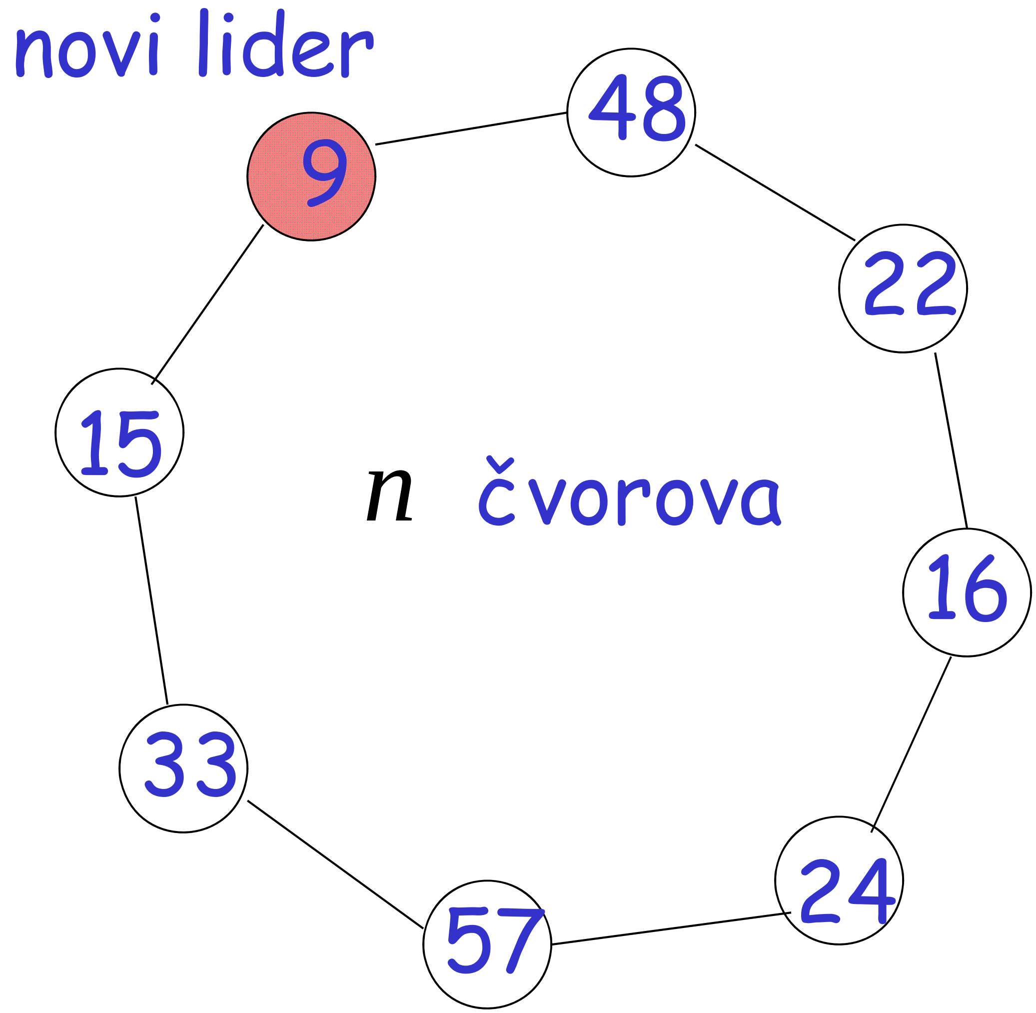
Runda 1 (n vrem. koraka): nešalju se poruke



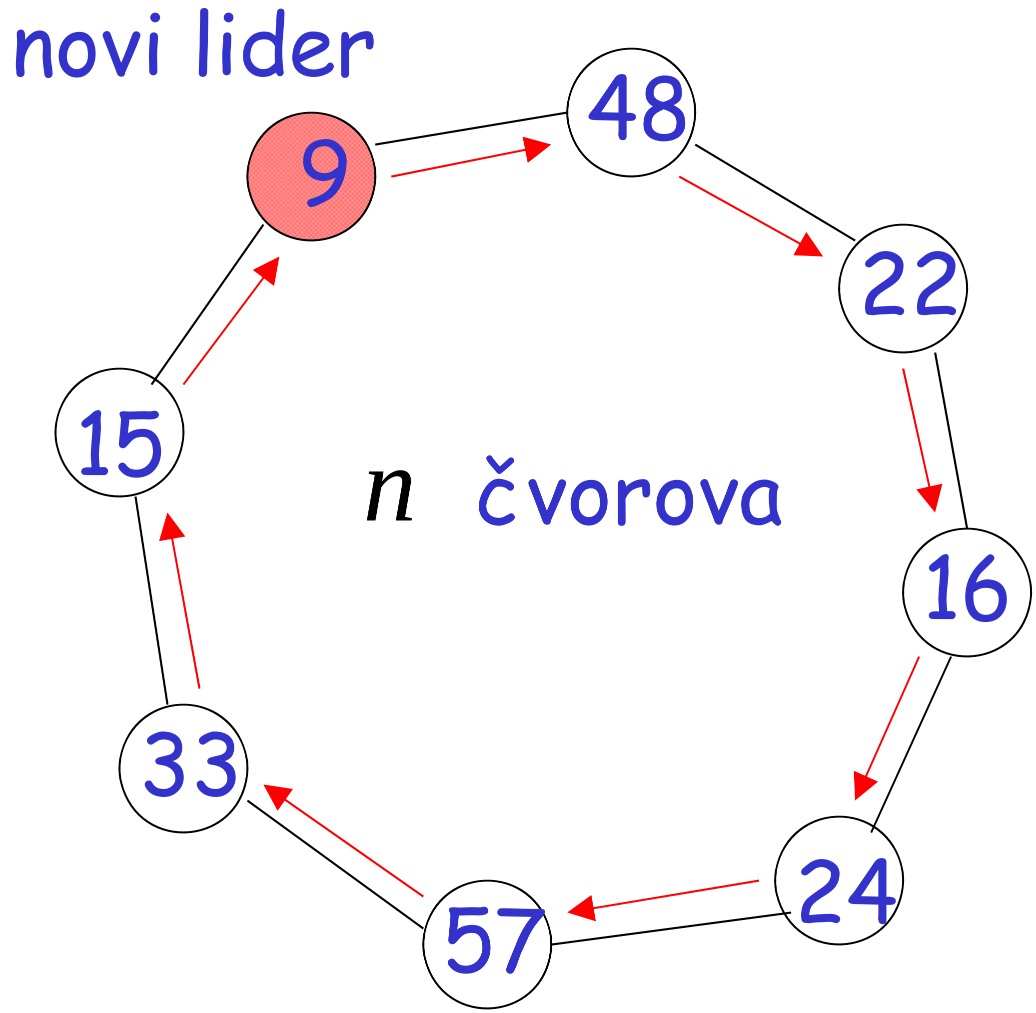
Runda 2 (n vrem. koraka): nešalju se poruke



Runda 9

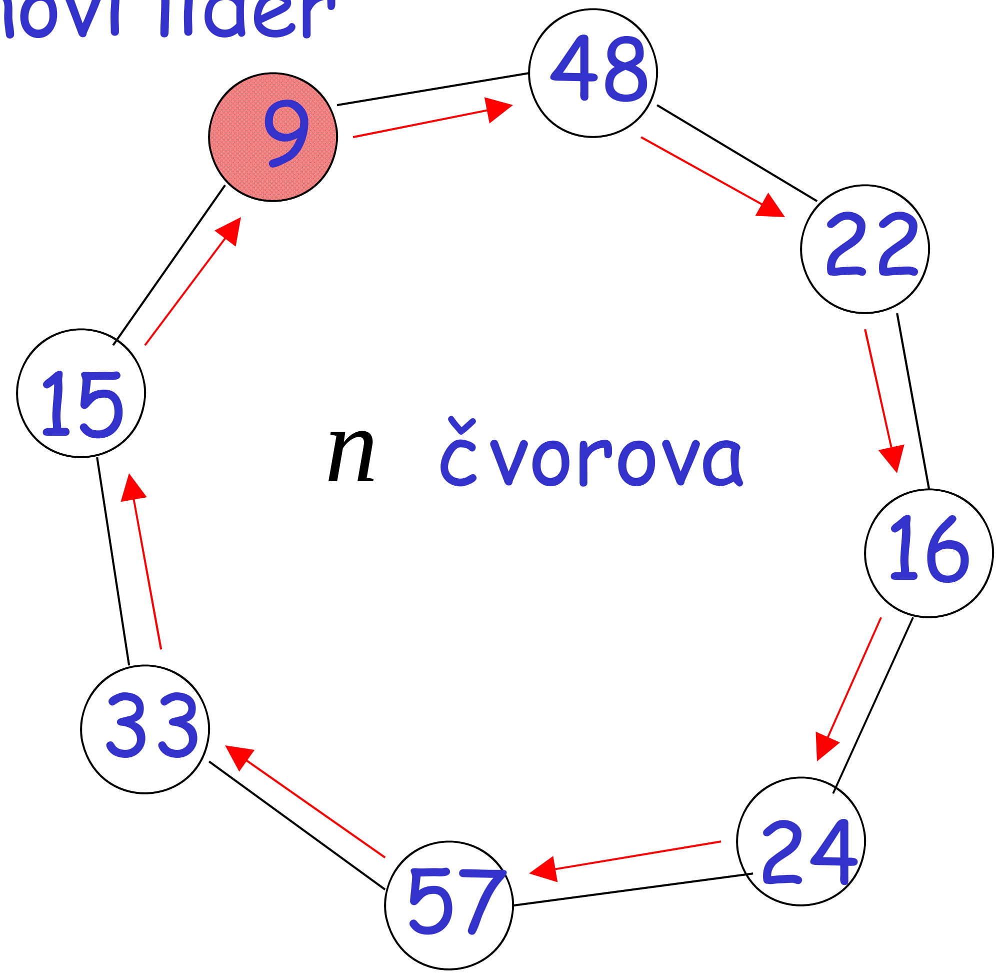


Runda 9 (n vrem. koraka): pošalje se n poruka



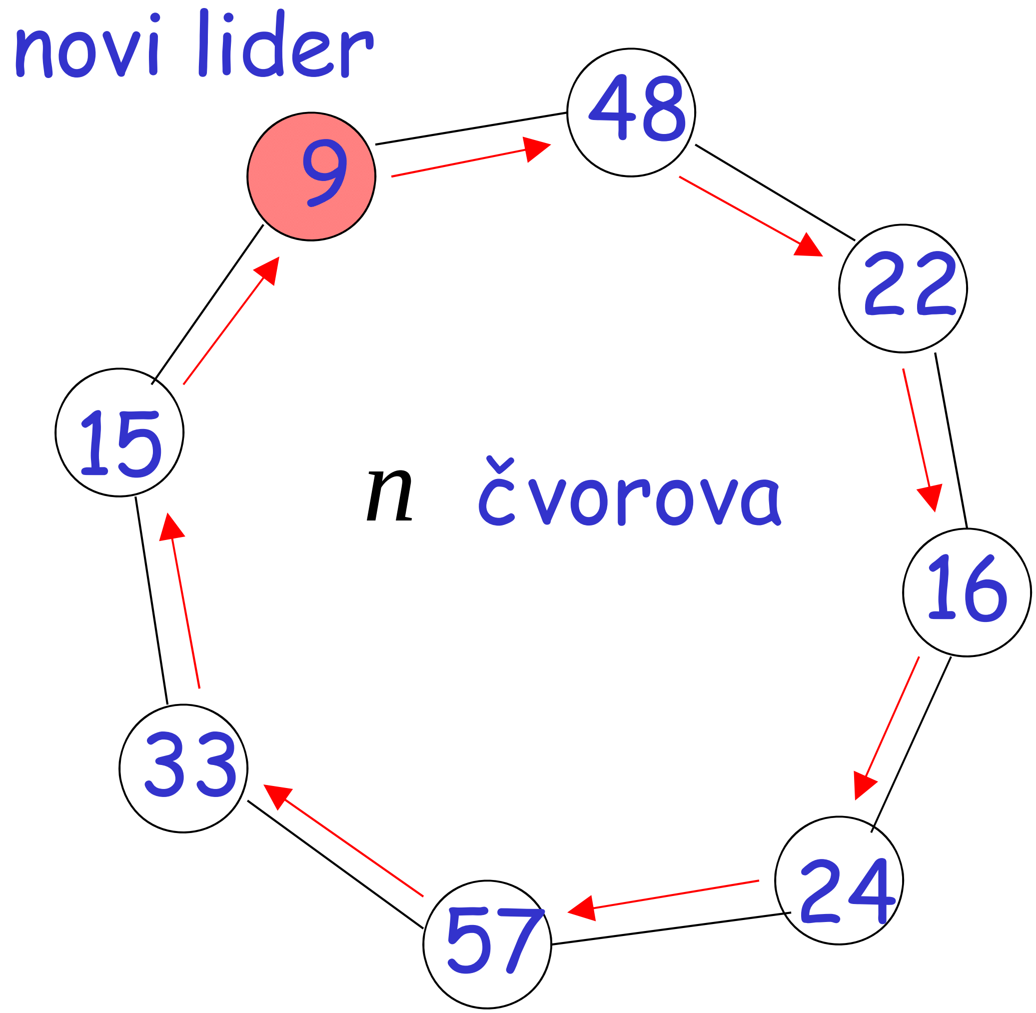
Runda 9 (n vrem. koraka): pošalje se n poruka

novi lider



Algoritam se završava

Runda 9 (n vrem. koraka): pošalje se n poruka



Ukupan broj poruka: n

Drugi $O(n)$ sinhroni algoritam

n <u>nije poznato</u>

Čvor sa <u>najmanjim</u> id se izabira za lidera

Algoritam:

- Svaki čvor ubrizgava poruku sa svojim id

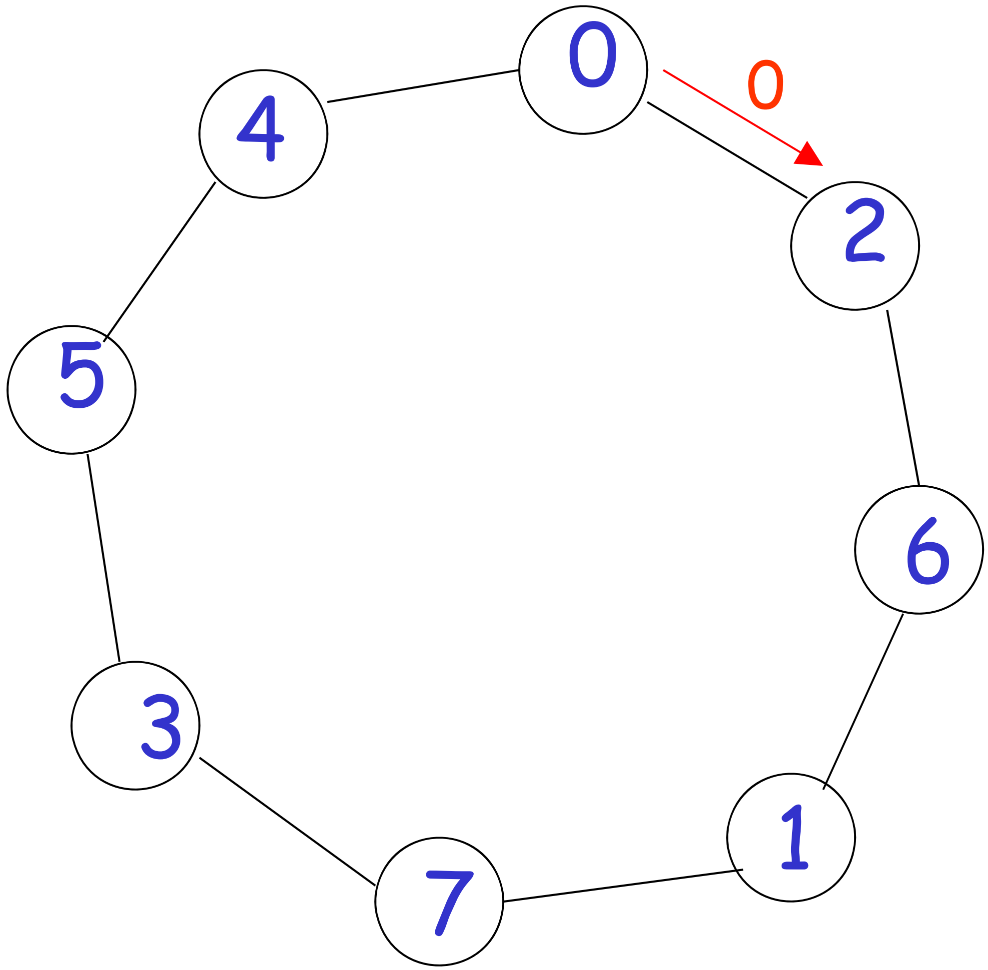
- Poruka sa id i se ubrizgava i prenosi sa učestanošću $\frac{1}{2^i}$

- Čvorovi koji su videli manje id apsorbuju poruke sa većim id

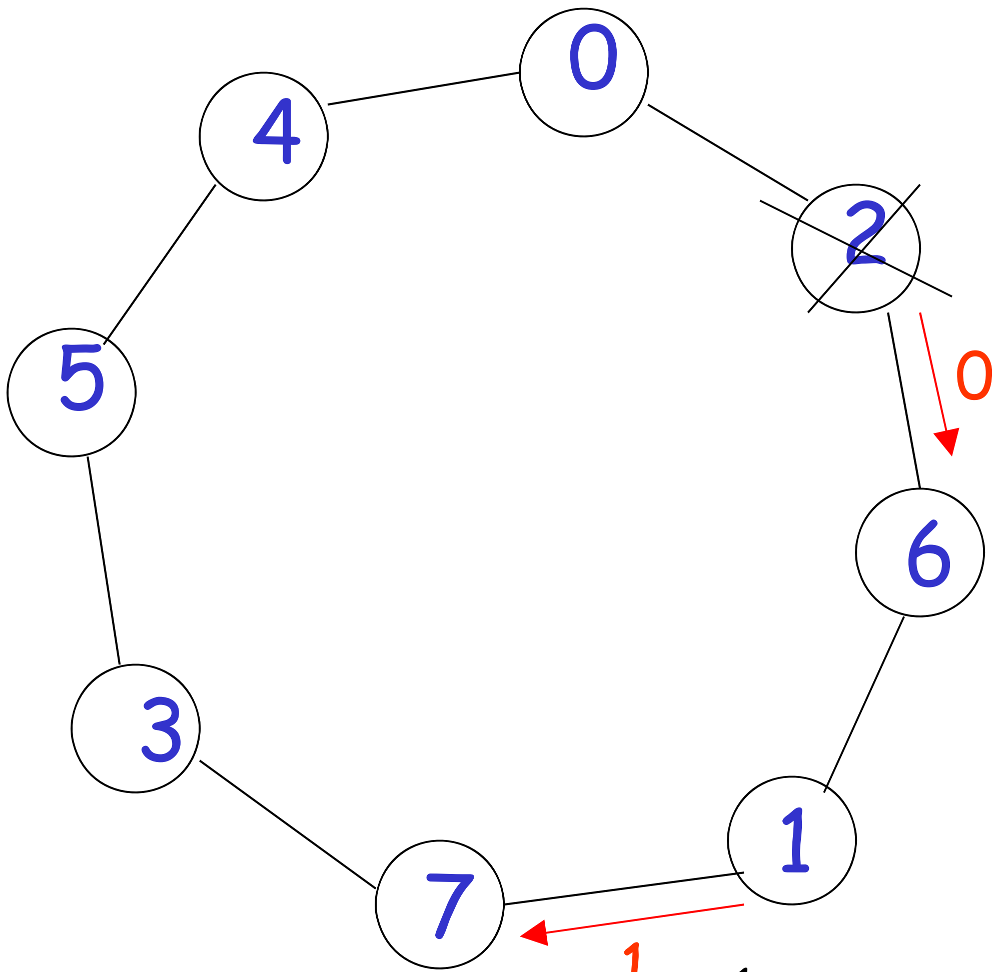
Vreme 1

Perioda prenosa

$2^0 = 1$



Vreme 2

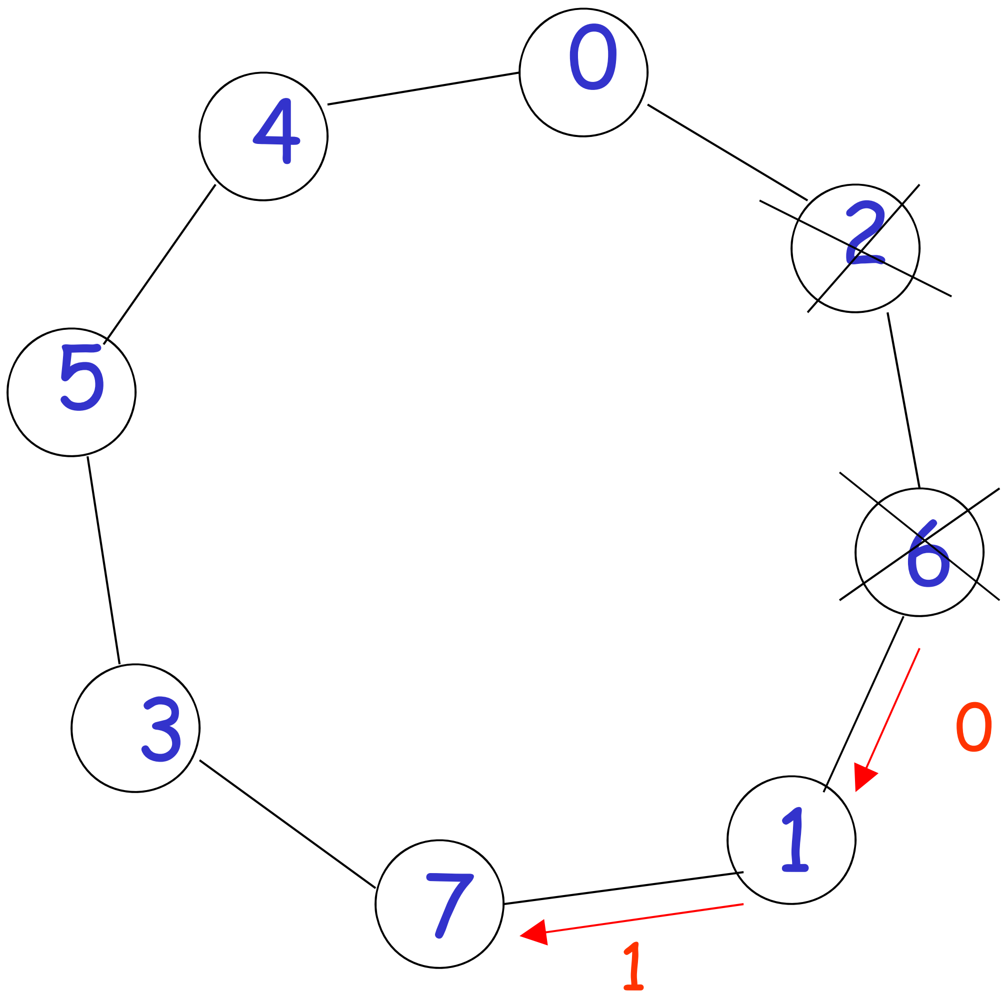


perioda

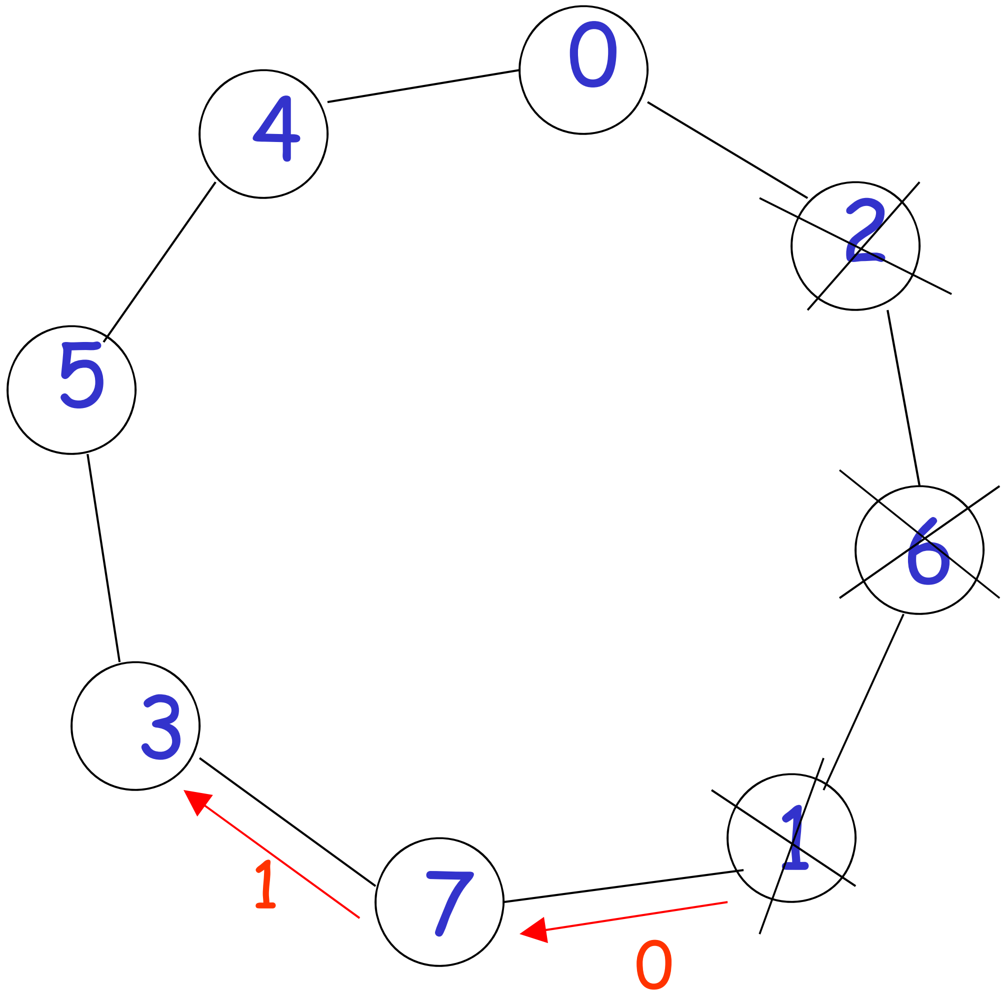
$2^0 = 1$

$2^1 = 2$

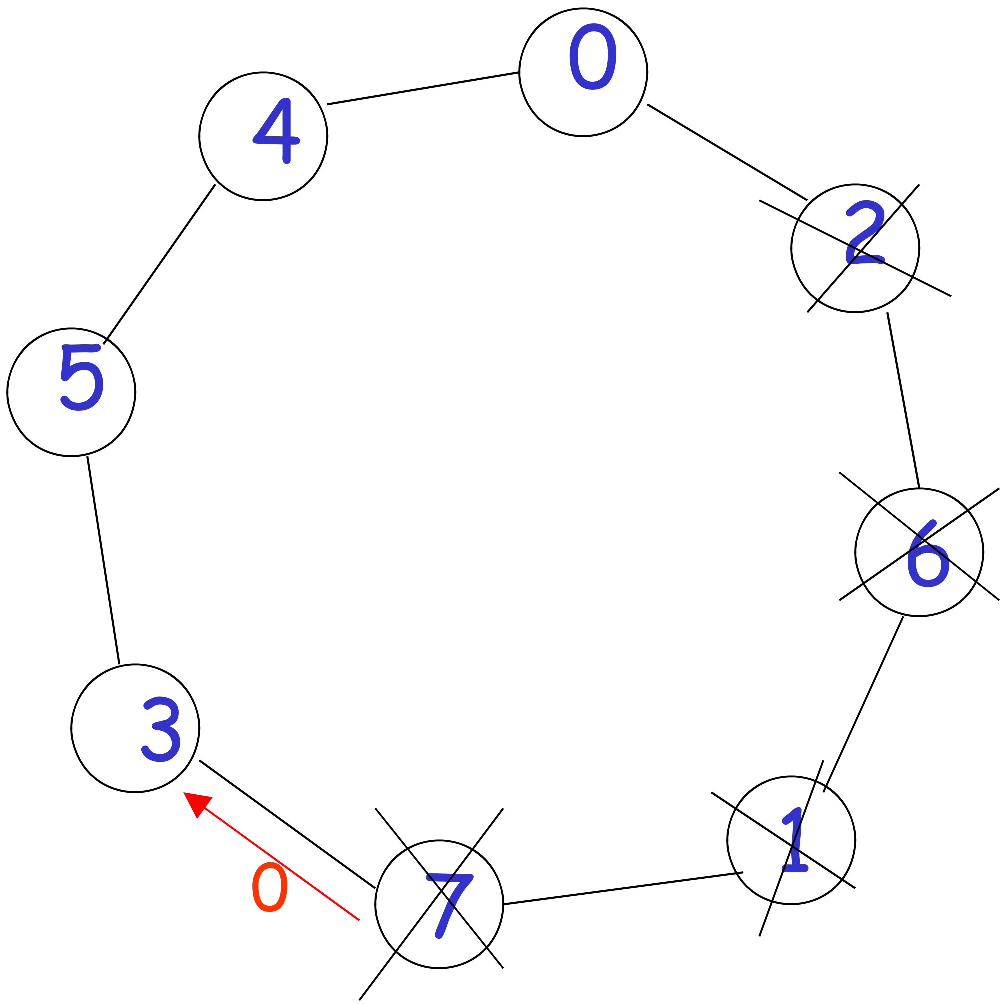
Vreme 3



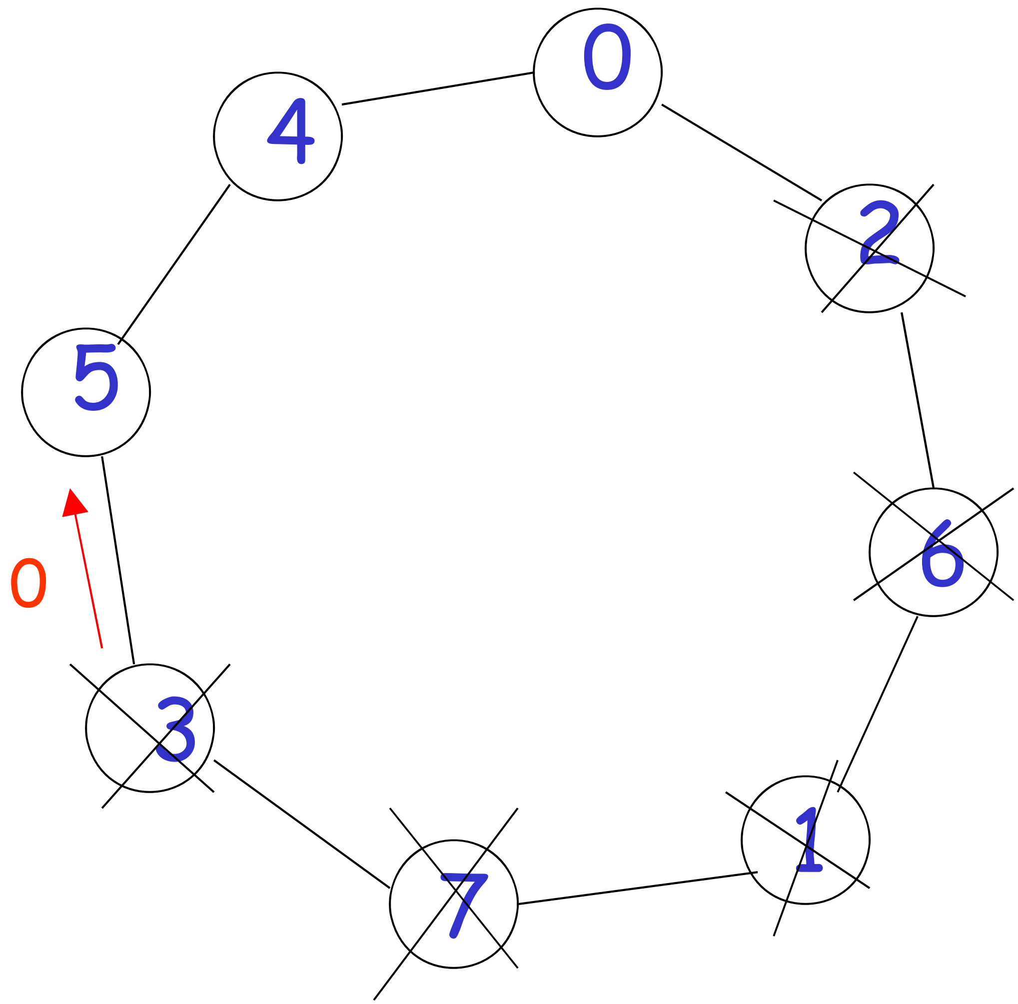
Vreme 4



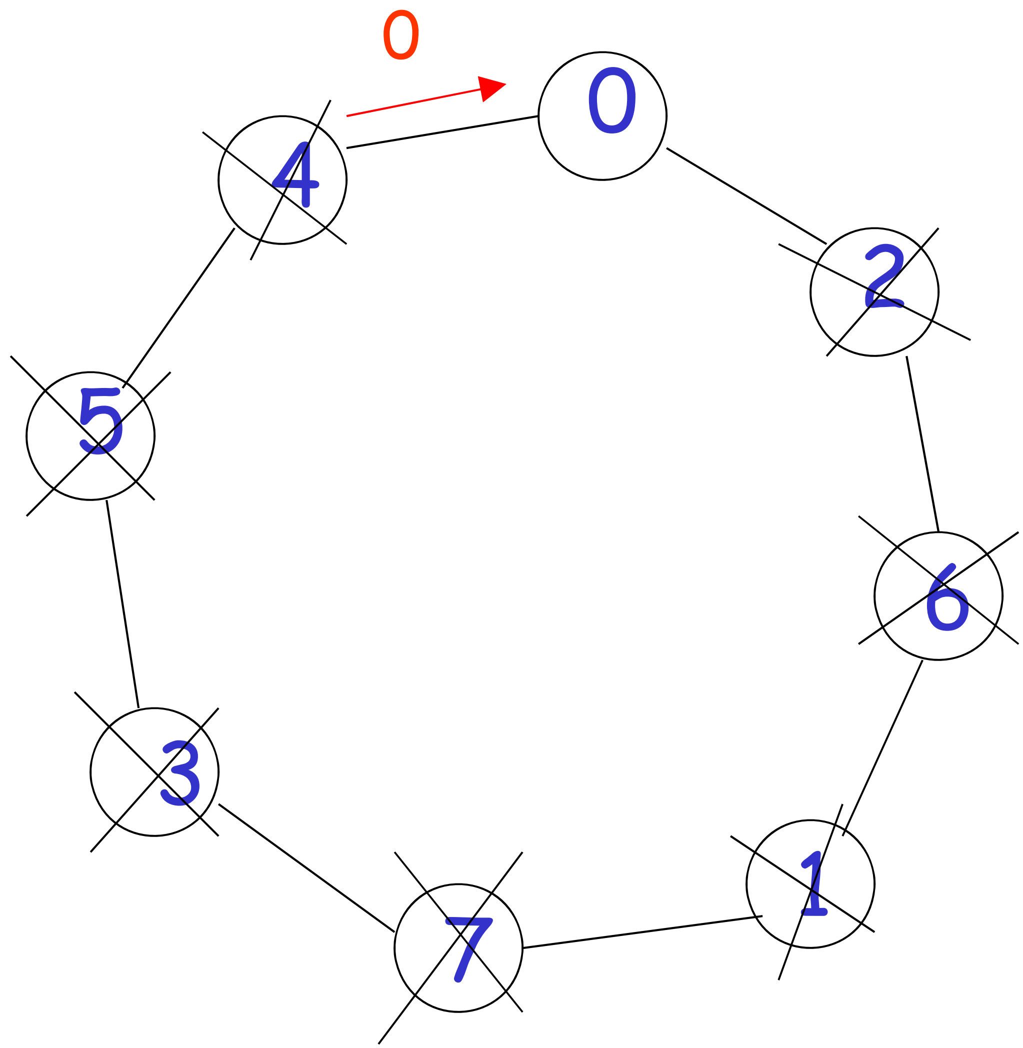
Vreme 5



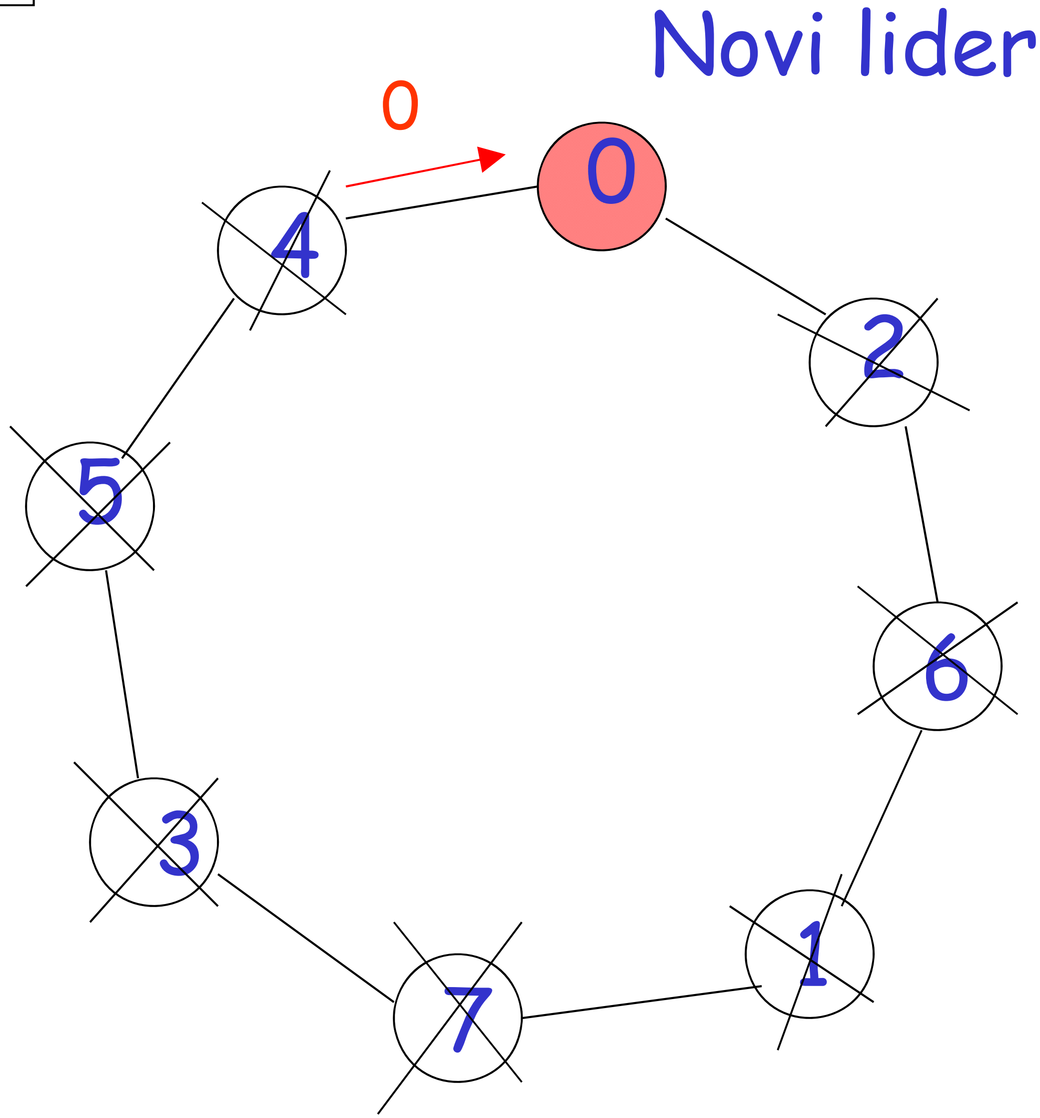
Vreme 6



Vreme 8



Vreme 8



Broj poruka

Predpost. lider ima (najmanji) id i

Ukupno vreme za algoritam: $n \cdot 2^{i}$

U slučaju da je $i = \Omega(n)$, algoritam je eksponencijalno spor

Uzmimo čvor sa prvim većim id

$$k > i$$

Ukupan broj poruka:

$$\frac{\text{ukupno vreme algoritma}}{\text{kašnjenje poruke na svakom luku}} = \frac{n \cdot 2^i}{2^k} \leq \frac{n}{2}$$

Uzmimo čvor sa prvim većim id

$$l > k > i$$

Ukupan broj poruka:

$$\frac{n \cdot 2^i}{2^l} \leq \frac{n}{4}$$

	id	poruke
niži	i	n
	k	$n/2$
viši	l	$n/4$

$\cdots\cdots$

Ukupno poruka: $n + \dfrac{n}{2} + \dfrac{n}{4} + \cdots \leq 2n$

Donja granica $\Omega(n \log n)$

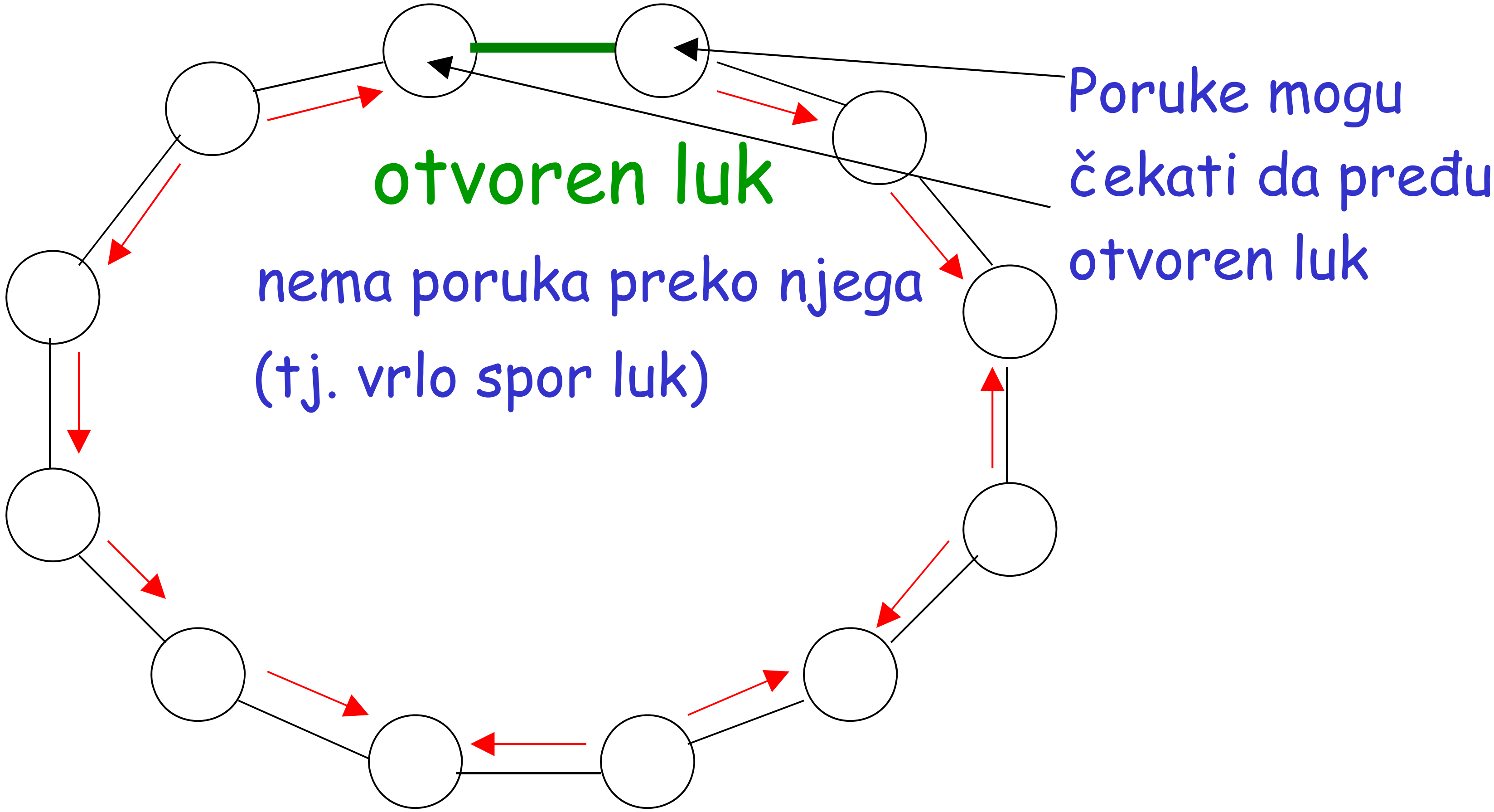
Predpostavimo da imamo algoritme u kojima:

- mreža je asinhrona
- za lidera se izabira čvor sa max id
- svi čvorovi moraju poznavati lidera
- veličina mreže n <u>nije</u> poznata

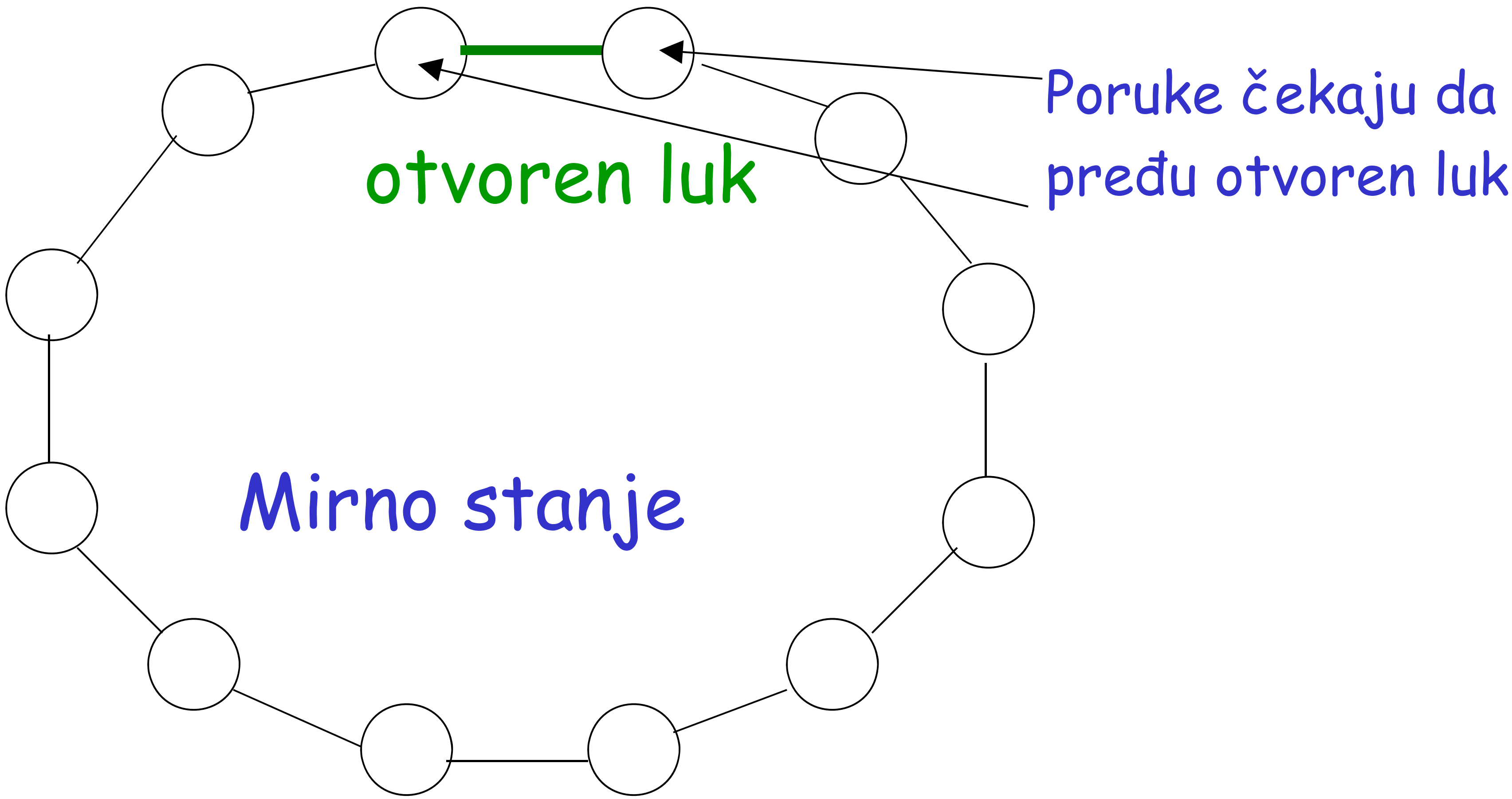
Dokazaćemo:

da je potrebno barem $\Omega(n \log n)$ poruka

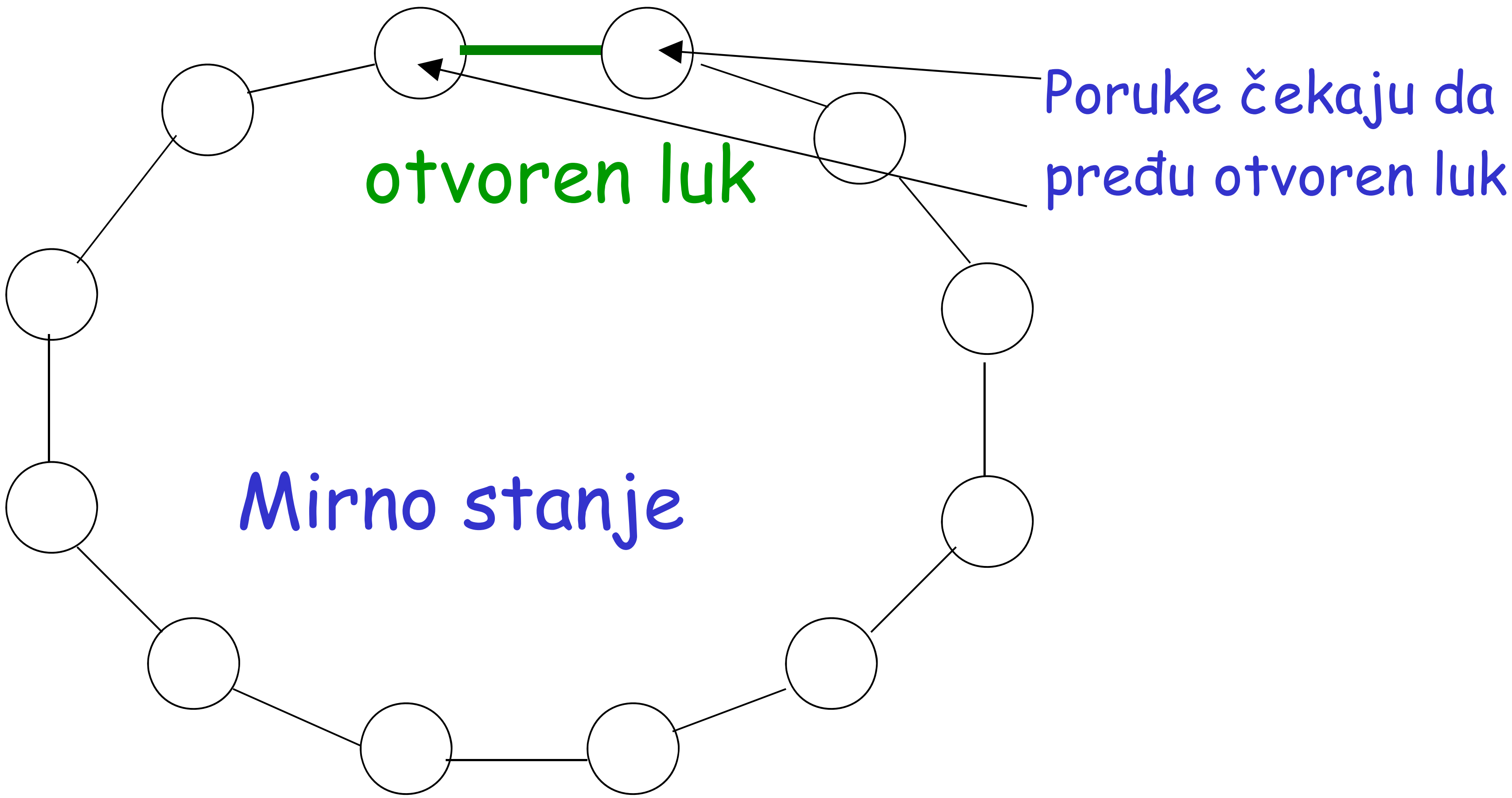
da bi se izabrao lider



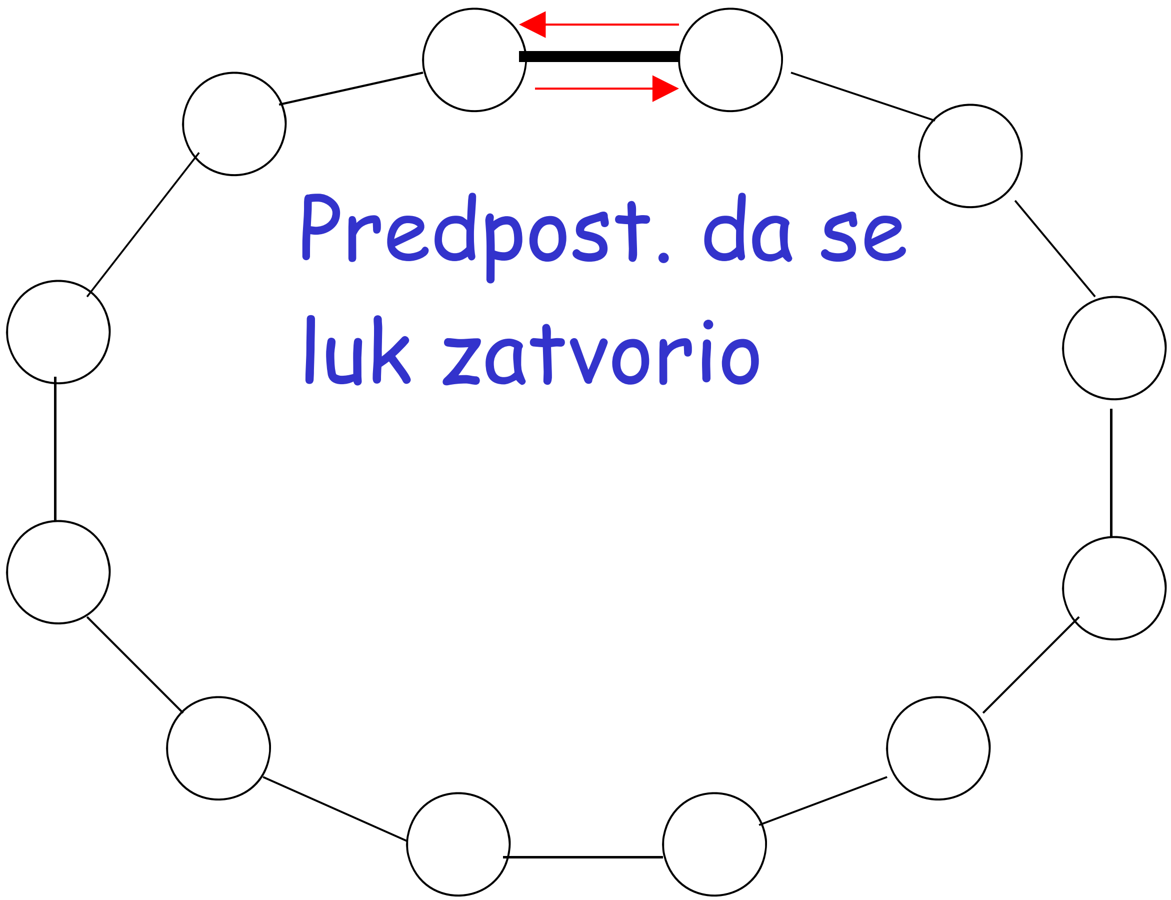
Postoji moguće asinhrono izvršenje sa otvorenim lukom



Ako luk ostane otvoren, onda će izvršenje na kraju doći do mirnog stanja gde više nema poruka koje se prenose u prstenu

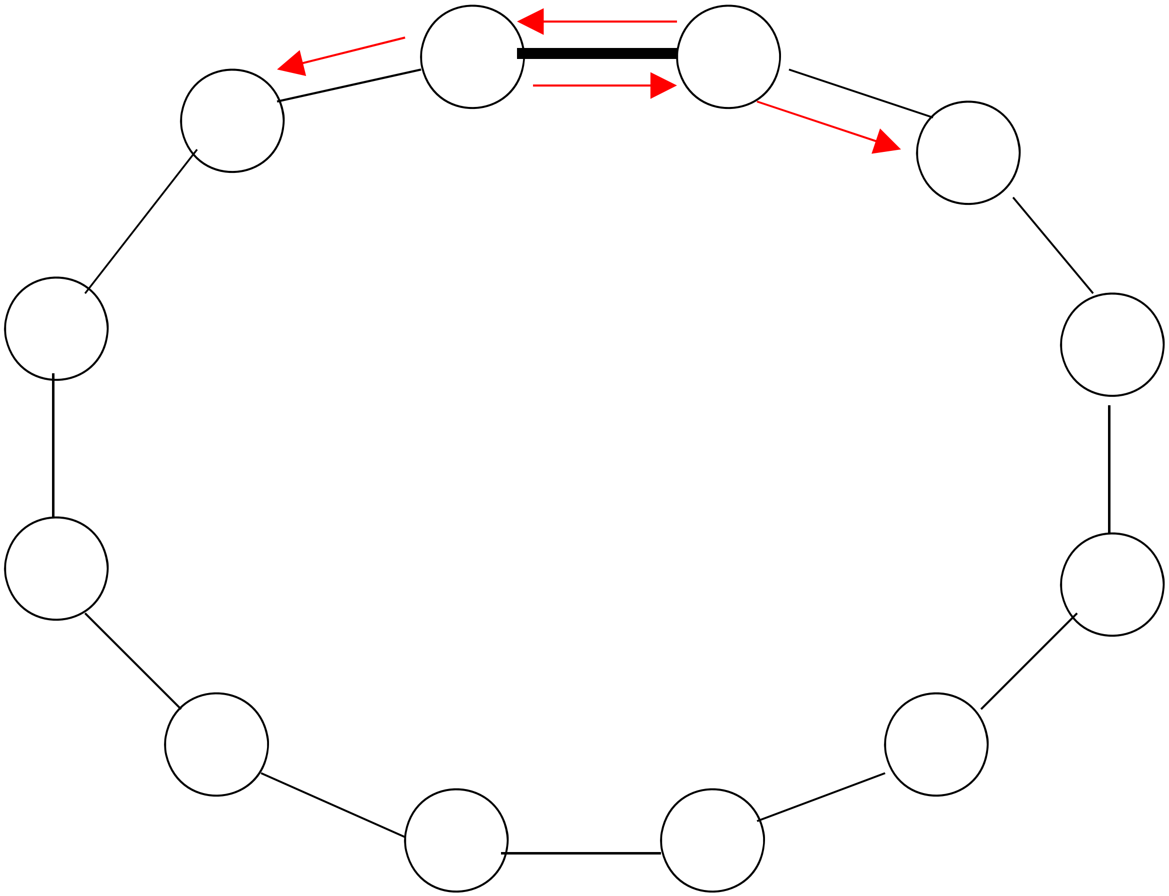


U mirnom stanju, čvor može poslati poruku samo nakon što primi poruku (zato nema poruka na prstenu)



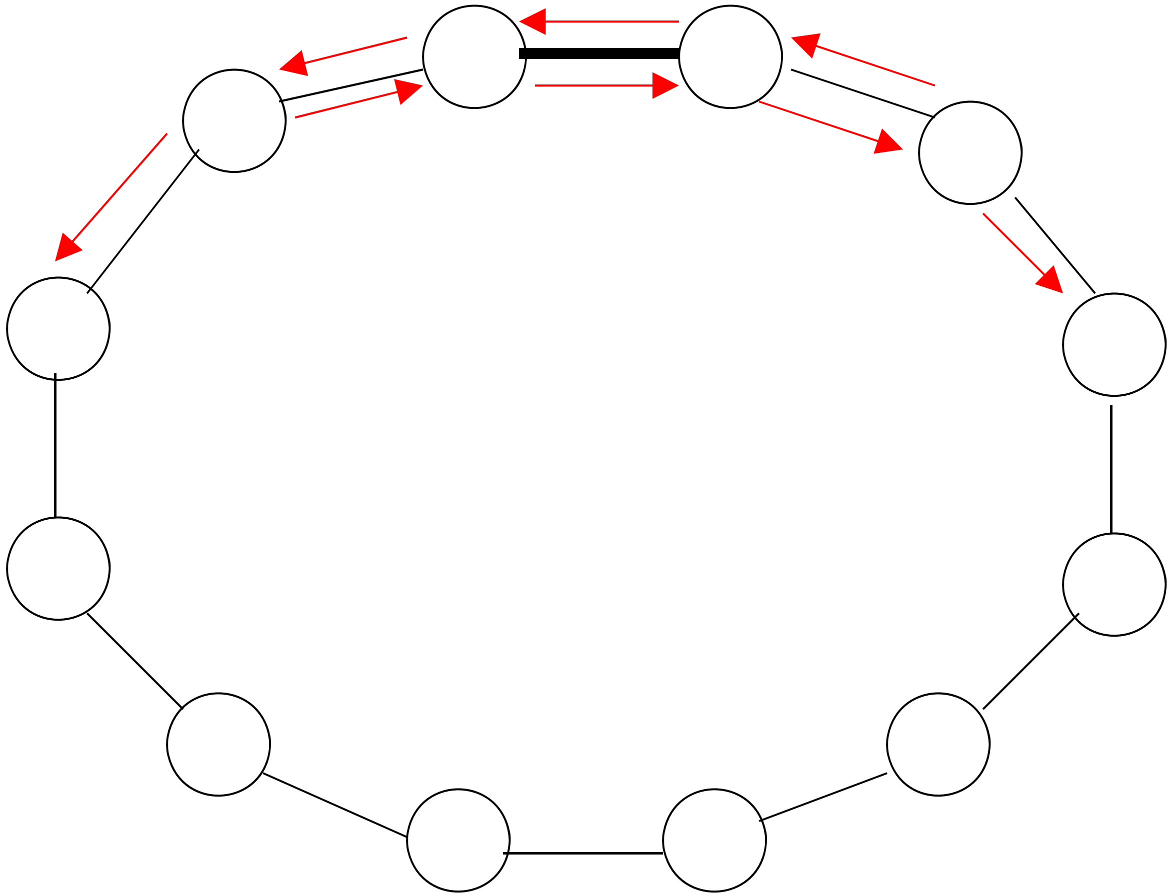
Vreme t

Ovo može izazvati propagaciju poruka do završetka algoritma



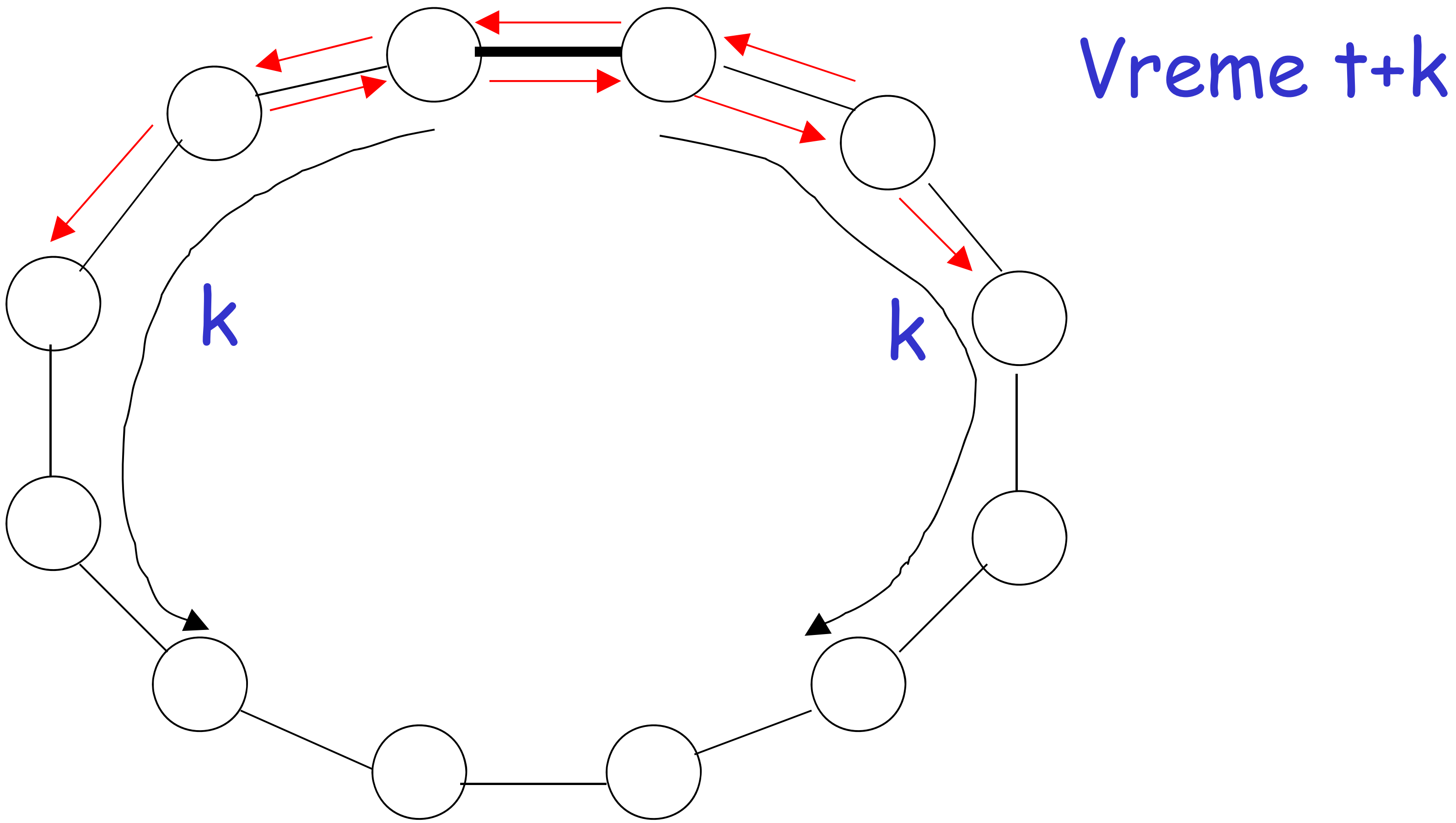
Vreme t+1

Ovo može izazvati propagaciju poruka do završetka algoritma

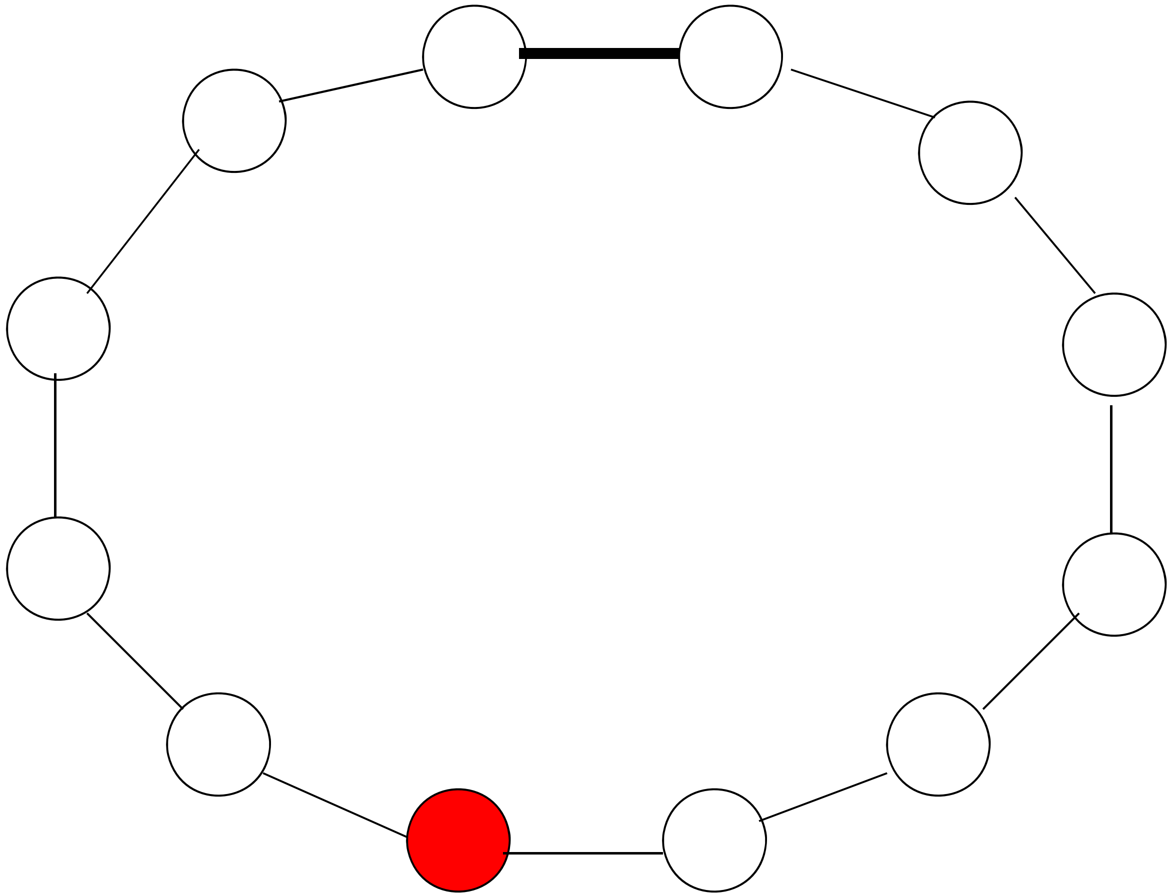


Vreme t+2

Ovo može izazvati propagaciju poruka do završetka algoritma



Posle k vremenskih koraka, radijus uticaja je k



Vreme t+x

Konačno, prsten se stabilizuje sa nekim liderom

Pokazaćemo, da postoji neko izvršenje sa n čvorova takvo da:

- postoji otvoren luk

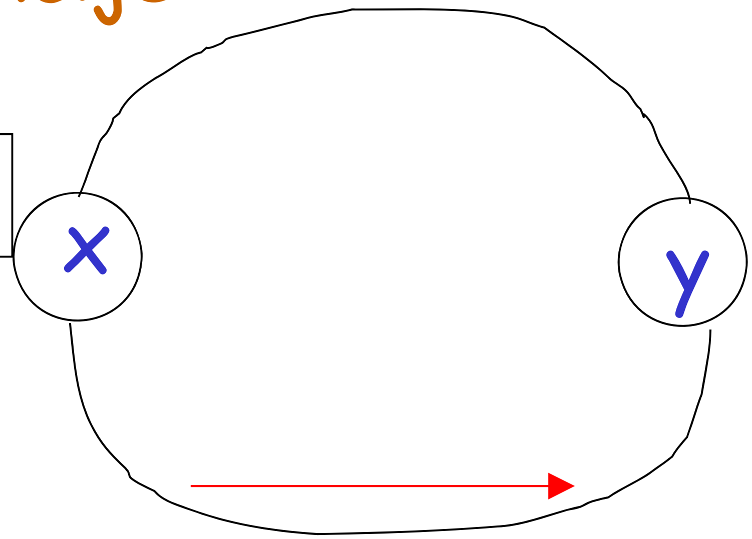
- primljeno je barem $M(n)$ poruka

gde je $M(2)=1$

$$M(n)=2M\left(\frac{n}{2}\right)+\frac{n}{4}$$

Dokaz pomoću indukcije

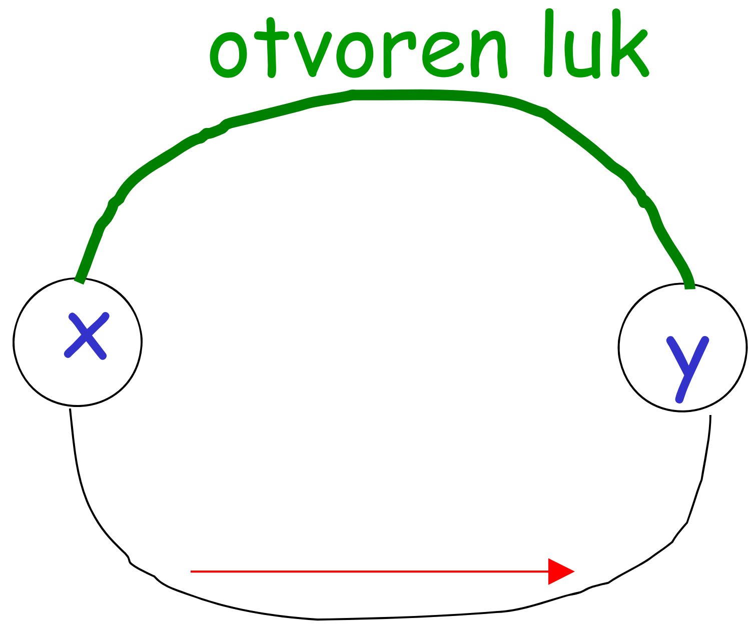
Osnovni sl. $n = 2$



Ako je $x > y$ onda je x lider

y bi trebao ovo da zna

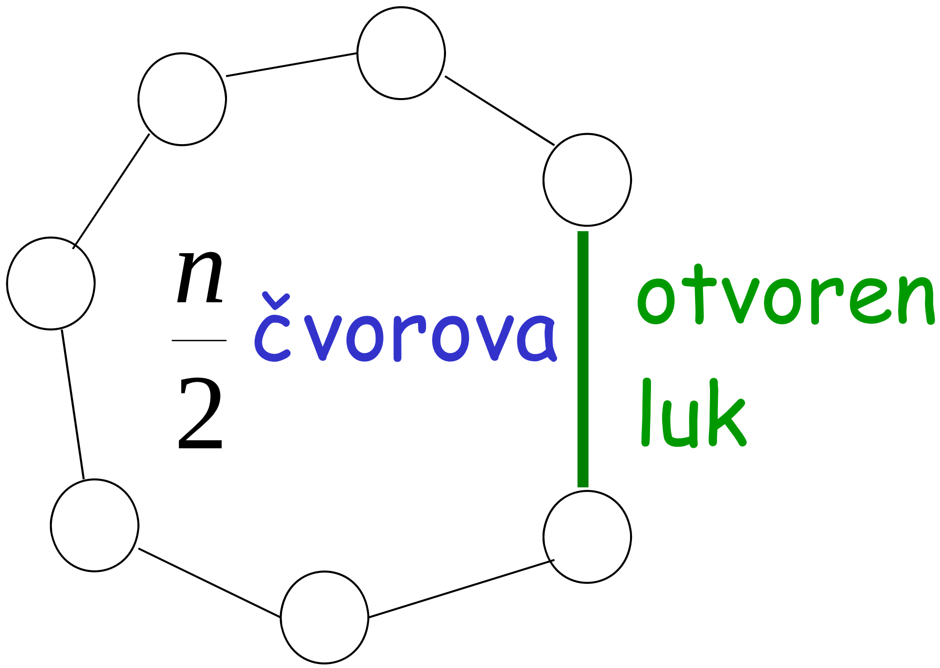
Zato se šalje jedna poruka: $M(2) = 1$



Poruka se može poslati po jednom luku

Slučaj $n > 2$

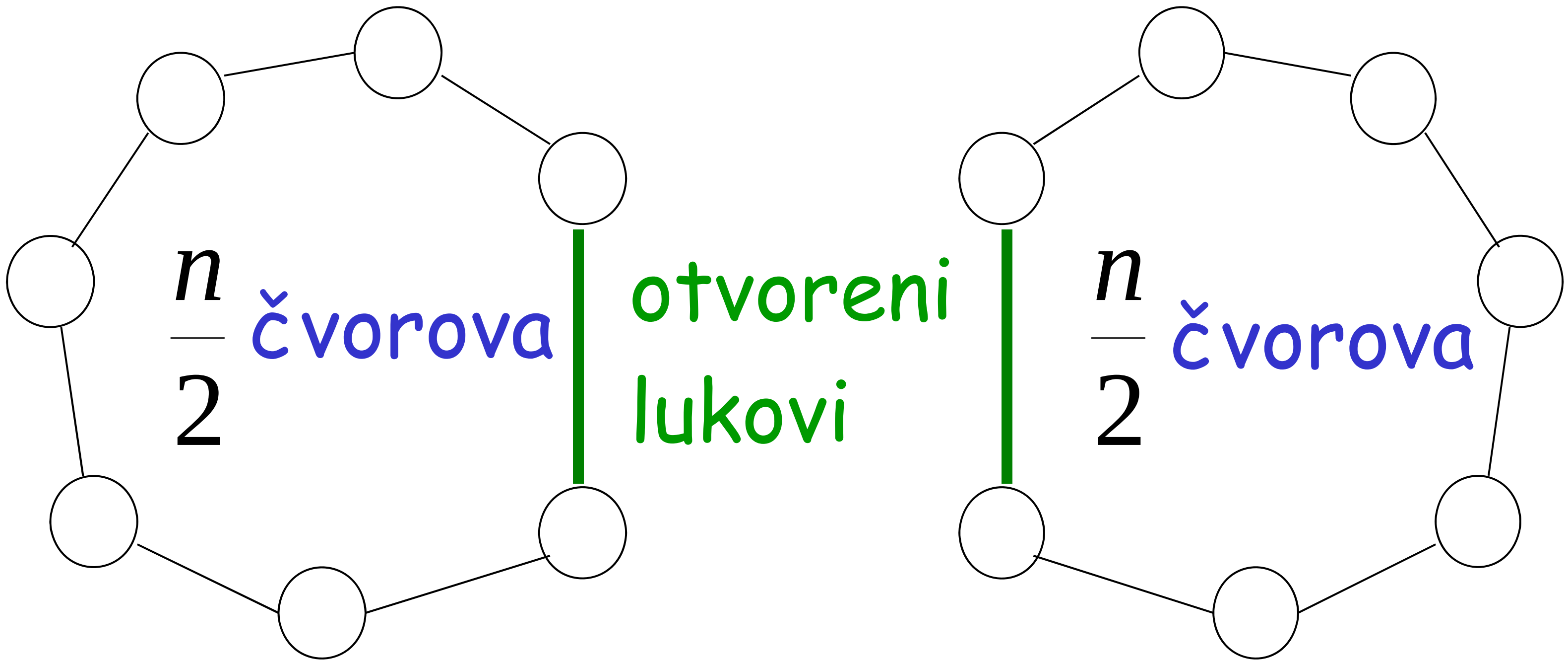
Iz indukcione hipoteze,
imamo neko izvršenje:



$M\left(\frac{n}{2}\right)$ poruka poslato

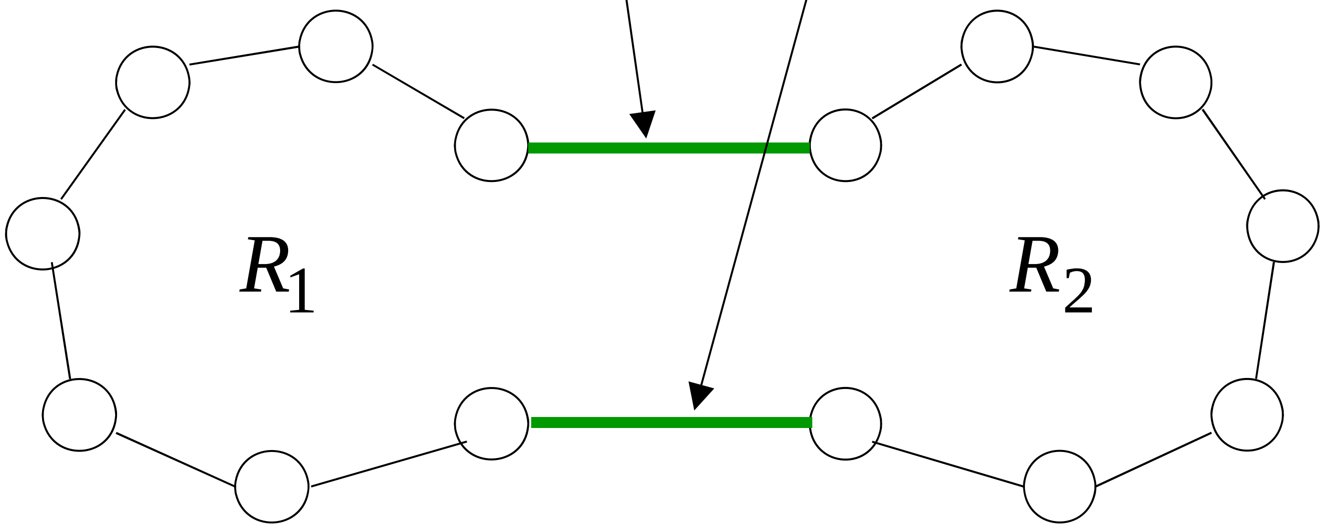
Slučaj $n > 2$

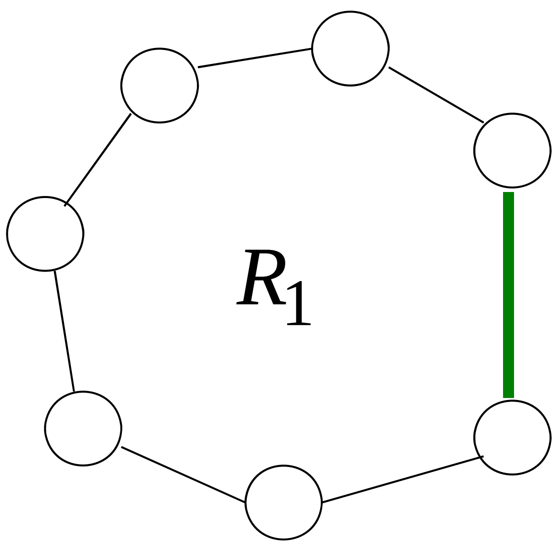
Iz indukcione hipoteze, imamo neko izvršenje:



$M\left(\frac{n}{2}\right)$ poruka poslato $M\left(\frac{n}{2}\right)$

Opservacija: Ako nema ovuda poslatih por.

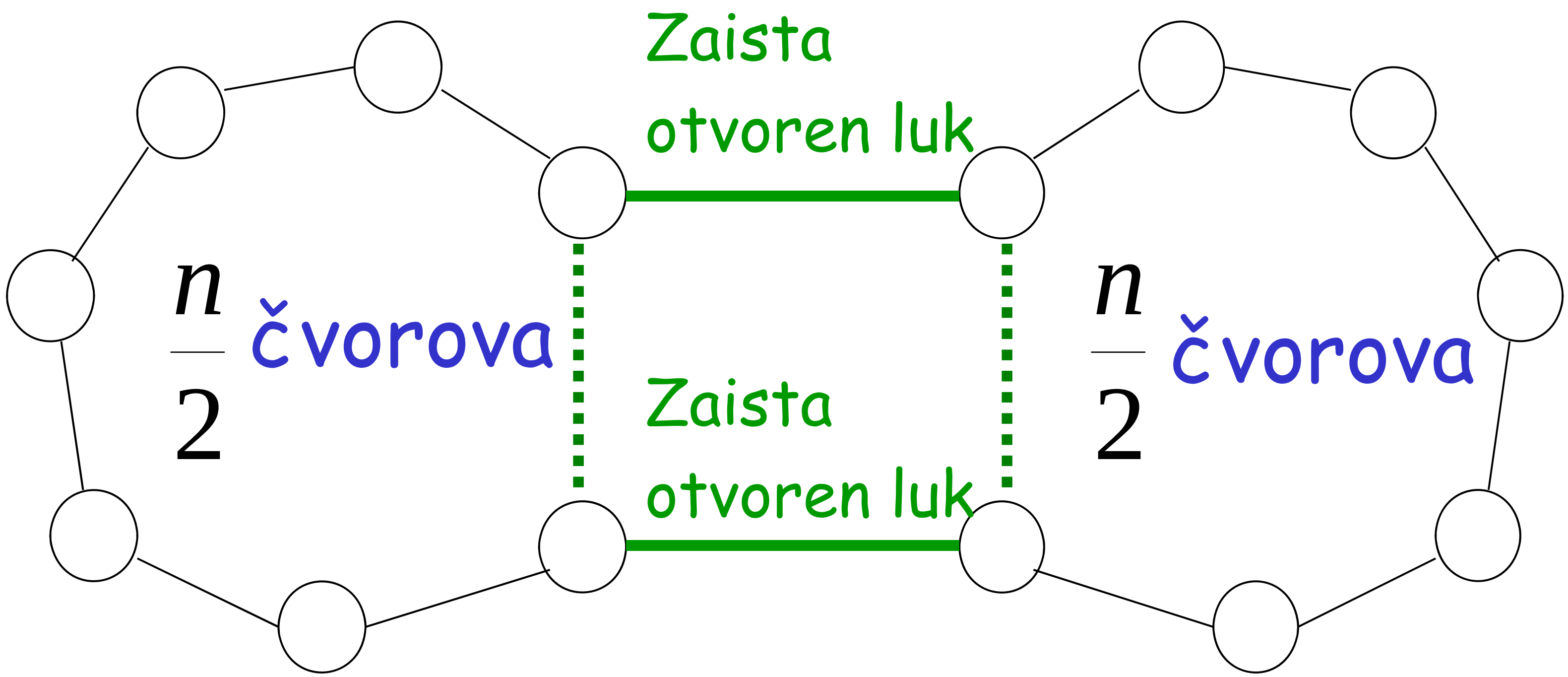




onda prsten R_1 nemože razlikovati ta dva scenarija

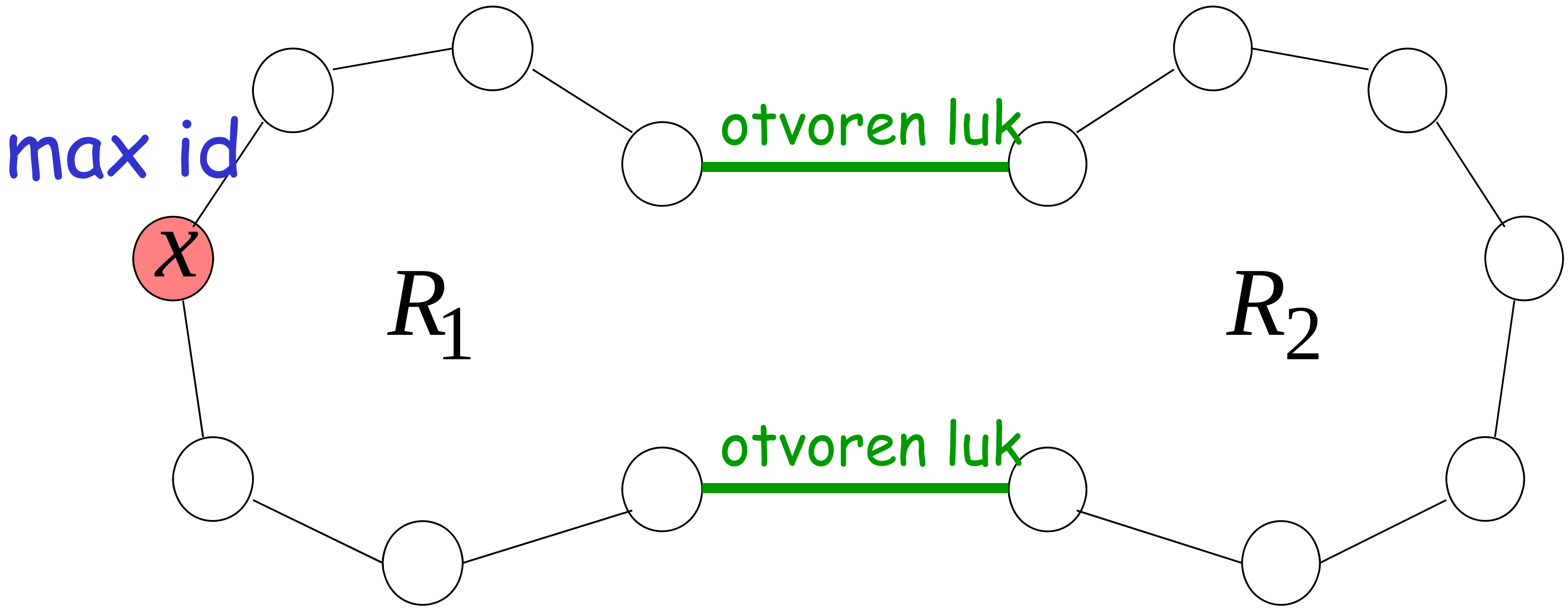
(slično za prsten R_2)

Zato, isti broj poruka mora biti poslat po pod-prstenima



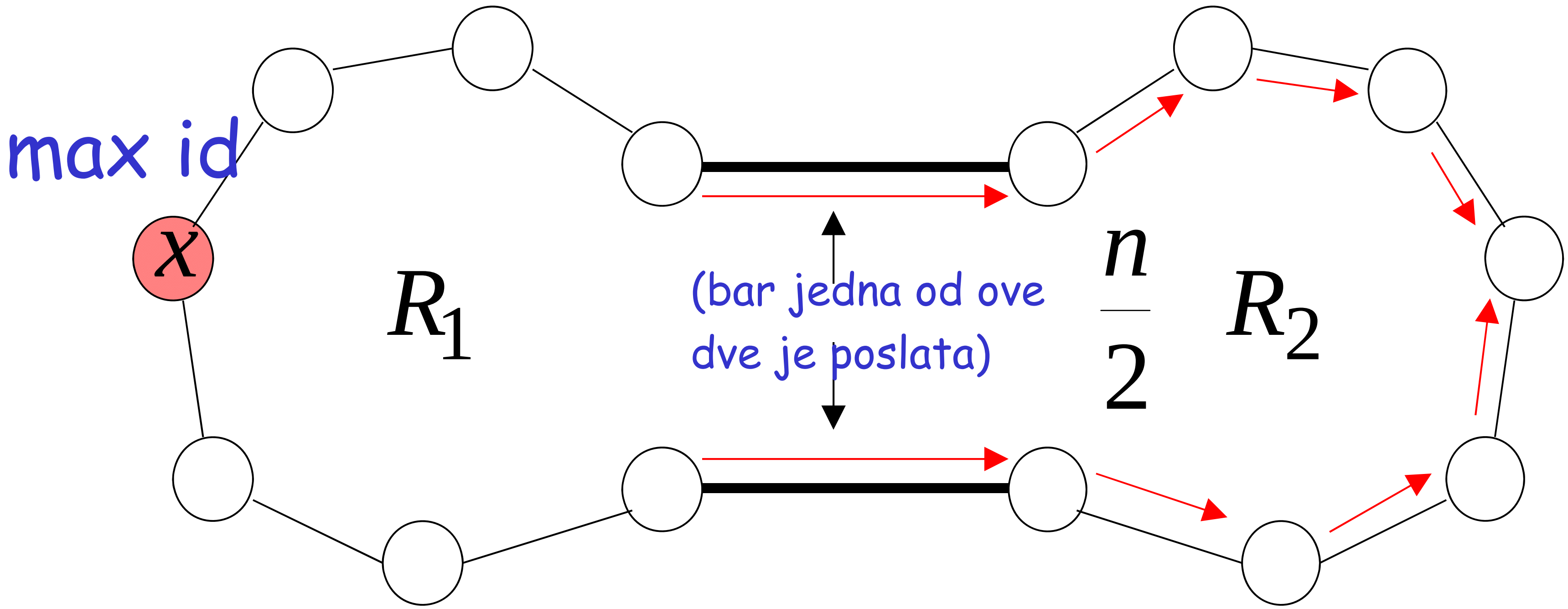
$M\left(\frac{n}{2}\right)$ poruka poslato $M\left(\frac{n}{2}\right)$

Svi čvorovi u R_2 treba da doznaju za x



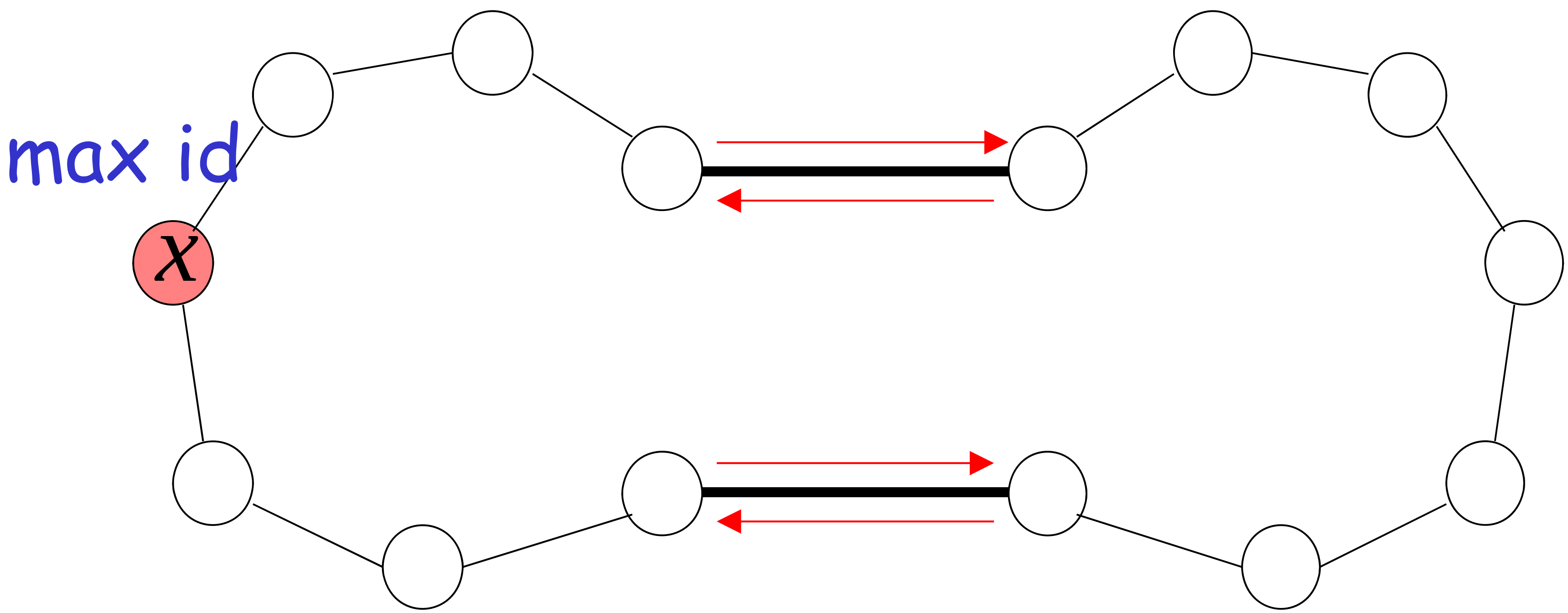
(predpost. da je max id u R_1)

Svi čvorovi u R_2 treba da doznaju za x



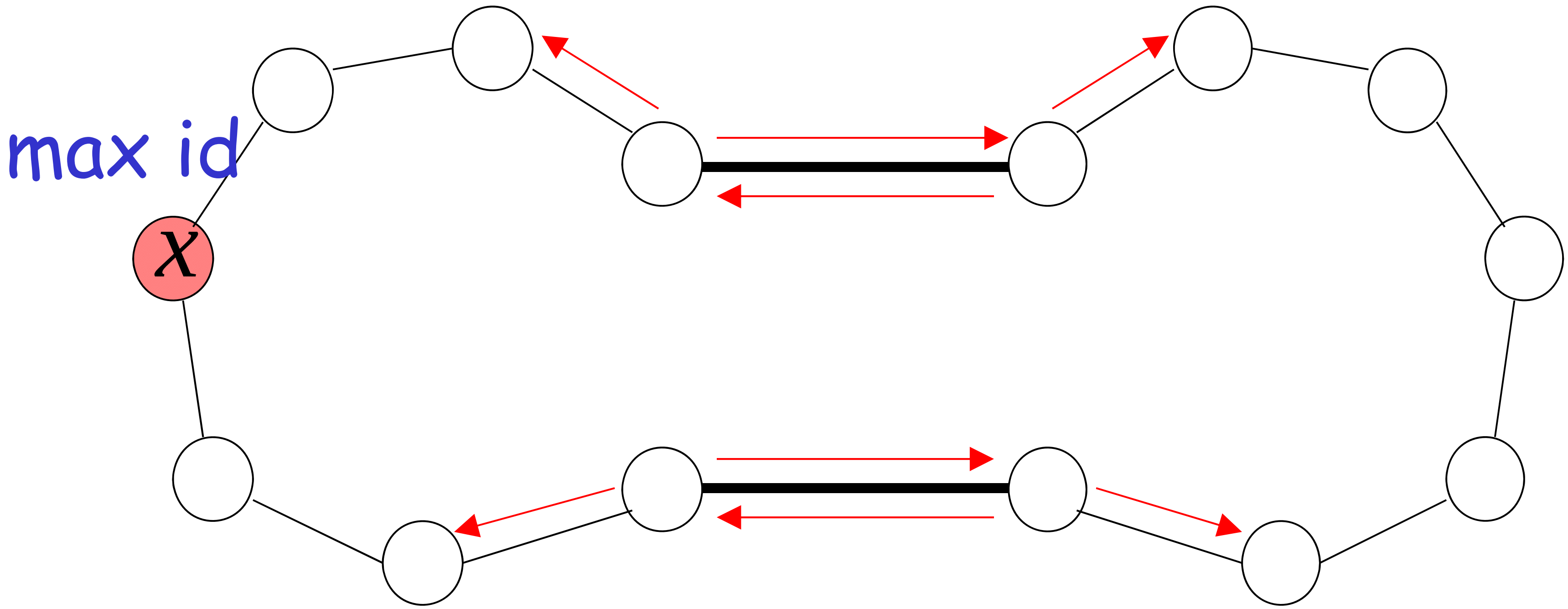
Zato, nakon što se otvoreni lukovi zatvore, barem $\frac{n}{2}$ poruka biva poslato

Pretpost. lukovi se zatvore u trenutku t



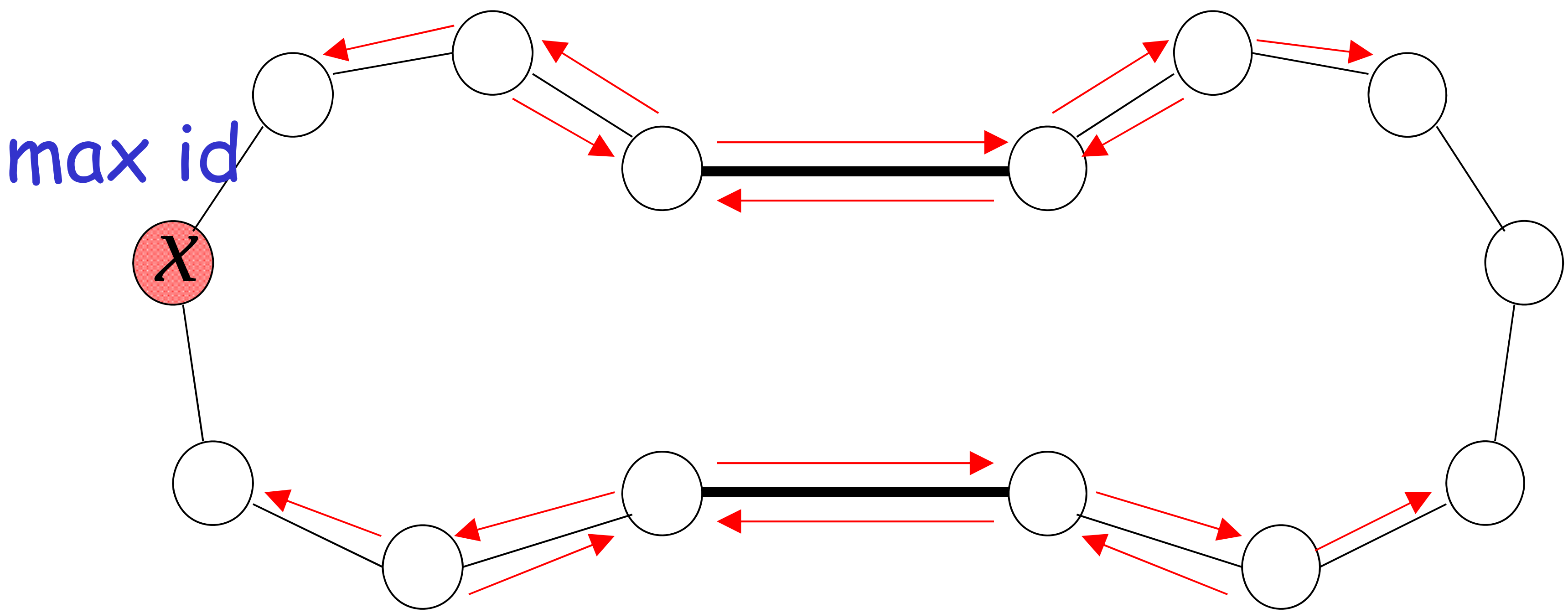
Barem jedna poruka je poslata

vreme $t+1$



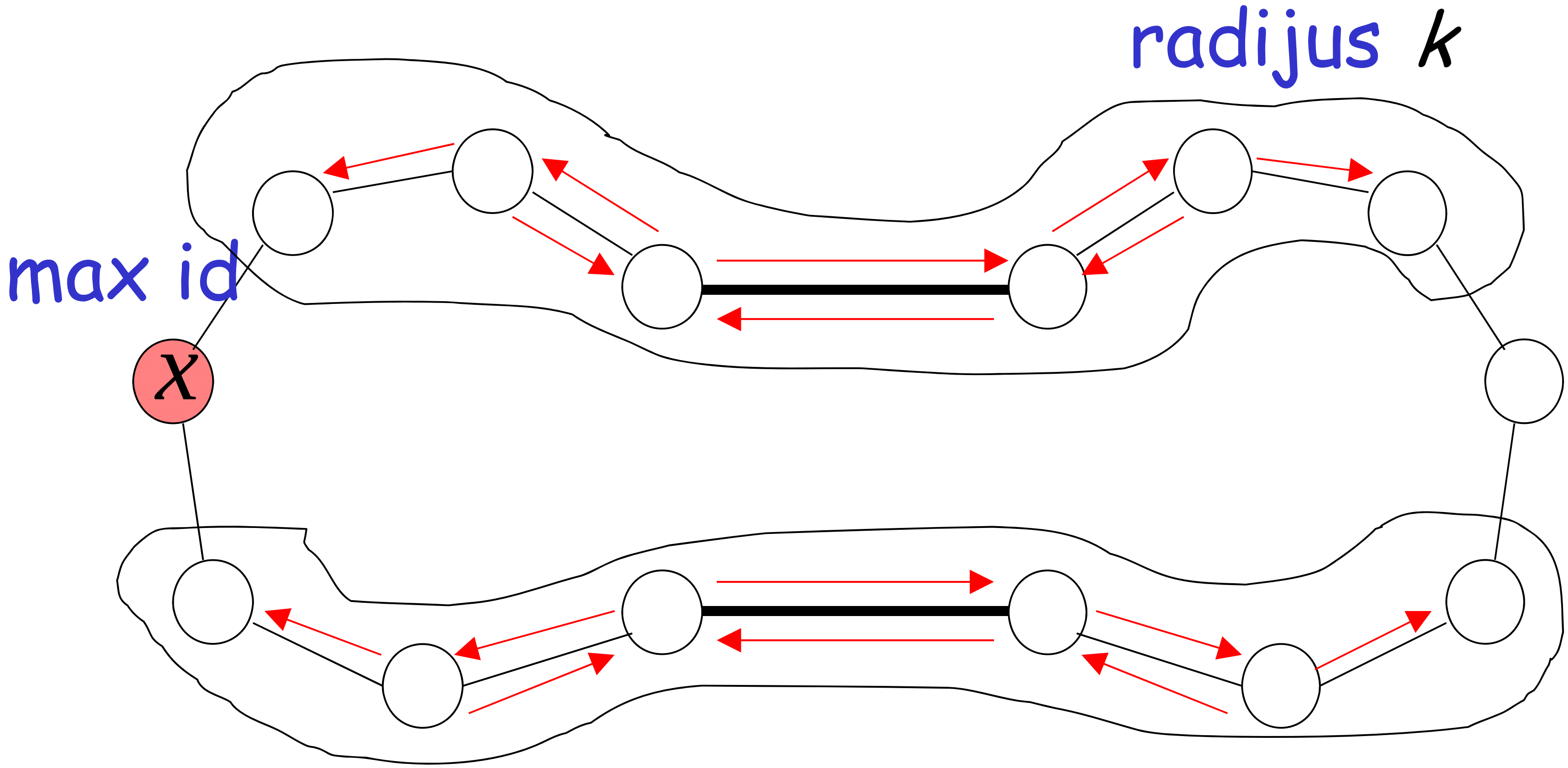
Barem jedna poruka je poslata

vreme $t+2$



Barem 2 poruke su poslate nakon trenutka t

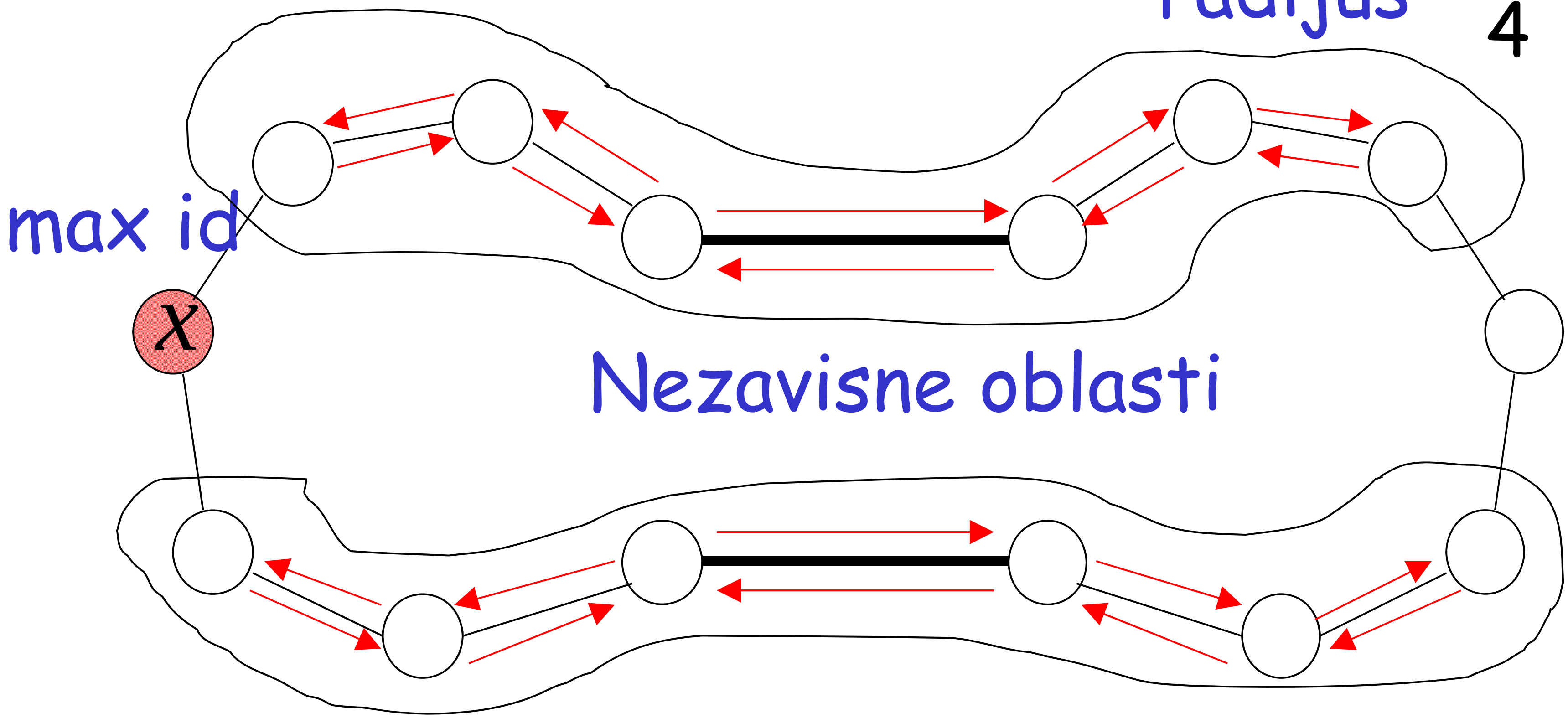
vreme $t + k$



Barem k poruka je poslato nakon trenutka t

vreme $t + \frac{n}{4}$

radijus $\frac{n}{4}$

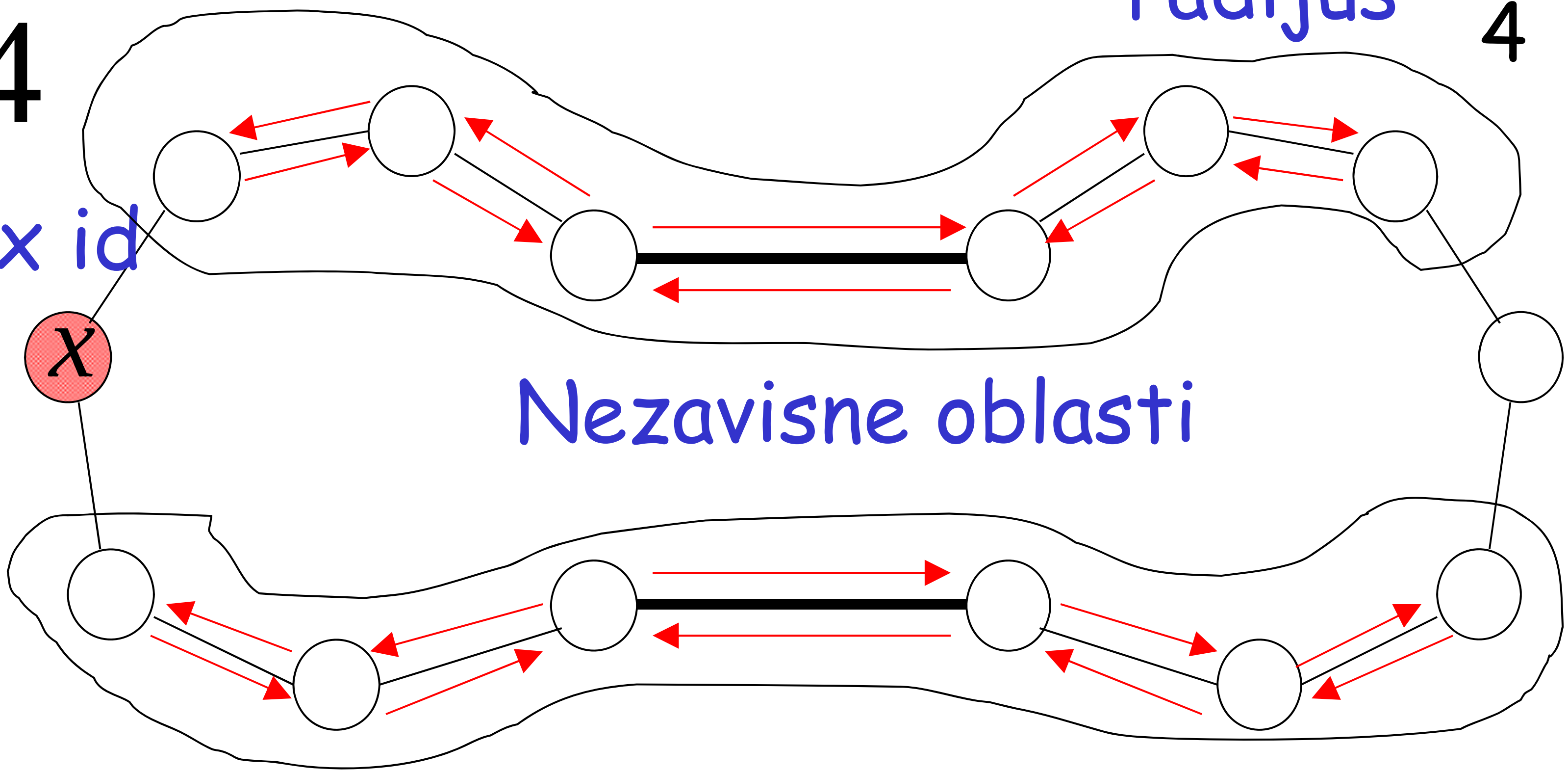


Barem $\frac{n}{4}$ poruka je poslato nakon trenutka t

Poruke $\frac{n}{4}$

vreme $t + \frac{n}{4}$

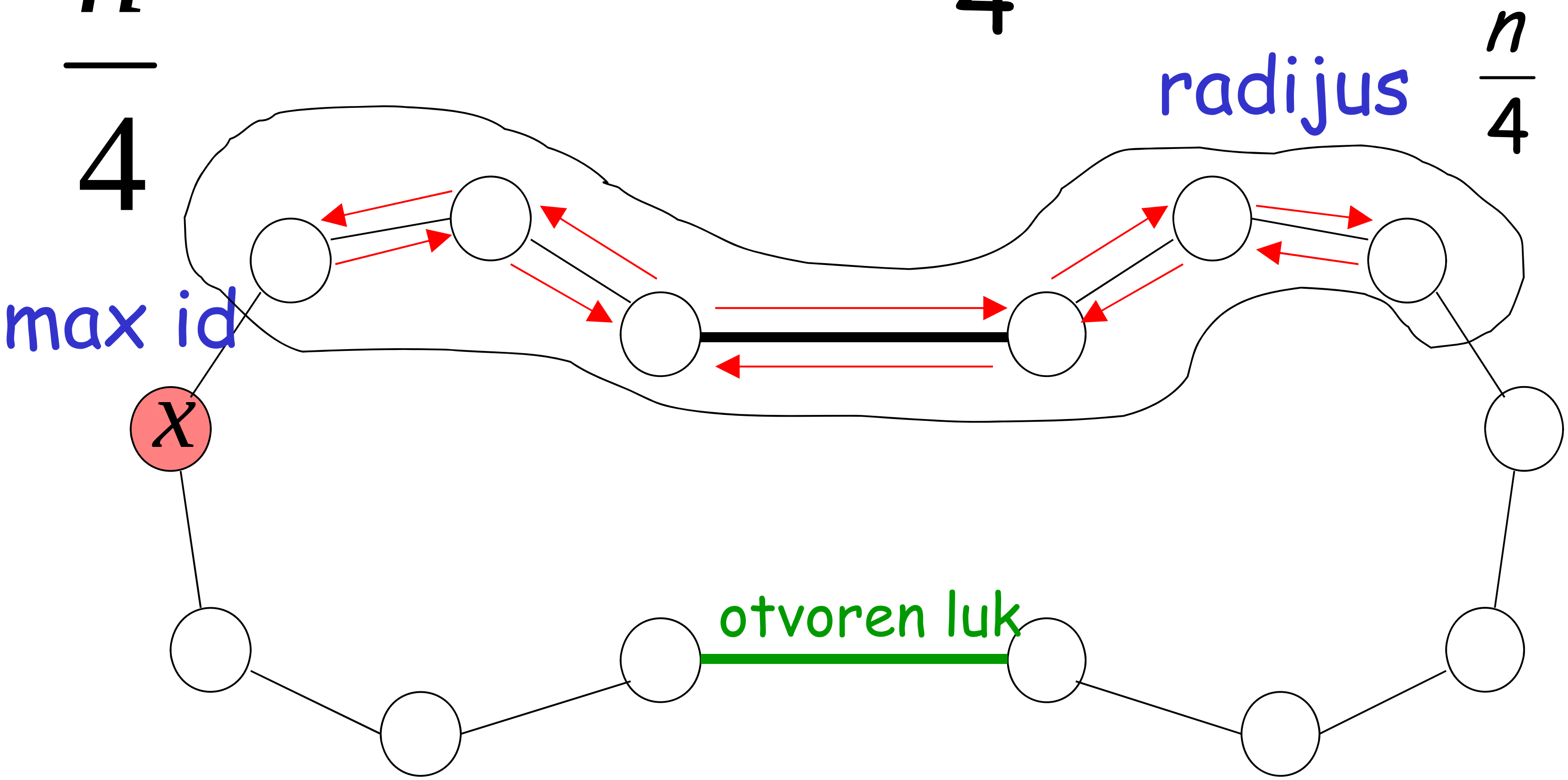
radijus $\frac{n}{4}$



Barem $\frac{n}{4}$ poruka je poslato u jednoj oblasti

Poruke $\frac{n}{4}$

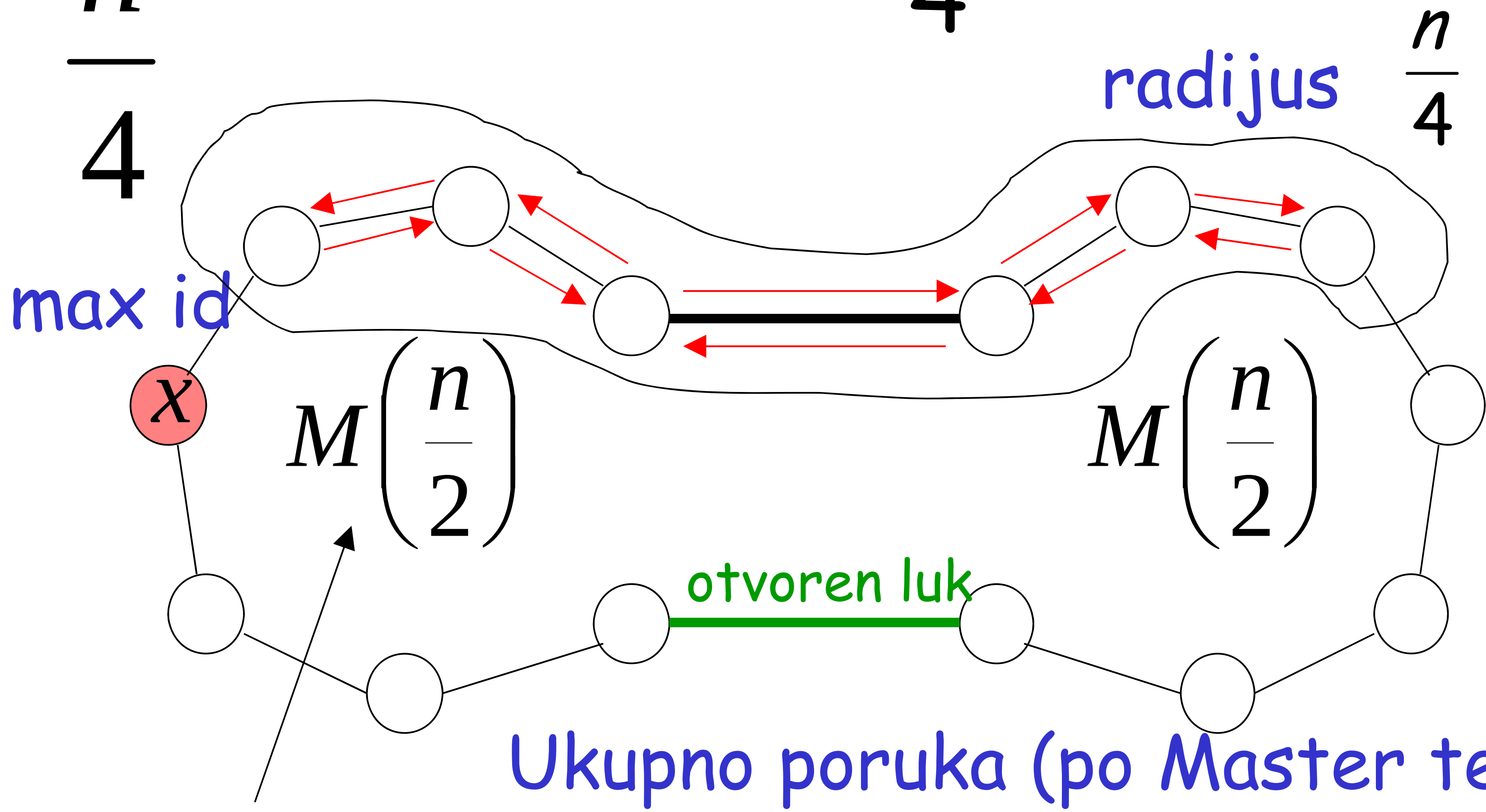
vreme $t + \frac{n}{4}$



Pošto su oblasti nezavisne, mogli smo zatvoriti samo jedan luk

Poruke $\dfrac{n}{4}$

vreme $t + \dfrac{n}{4}$



Ukupno poruka (po Master teor.)

$$M(n) = 2M\left(\frac{n}{2}\right) + \frac{n}{4} = \Omega(n \log n)$$

Indukciona Hipoteza