



# PROGETTO DI SISTEMI DISTRIBUITI E PERVASIVI

TERESA TANZI - 925574





SERVER REST



# LISTA CASE SERVICE

Root: /case

Metoto	Path	Input	Output	Errori	Descrizione
POST	/add	Casa	List [Casa]	400 BAD_REQUEST 409 CONFLICT	Id o porta negativi Id o porta già esistenti
DELETE	/remove/{id}	-	-	400 BAD_REQUEST 404 NOT_FOUND	Id negativo Id non trovato nella lista di case
GET		-	List [Casa]	404 NOT_FOUND	Non ci sono case nella rete

```
Casa:  {
        id: 1,
        ip: "localhost",
        porta: 7890
    }
```

# STATS SERVICE (I)

Root: /stats

Metodo	Path	Input	Output	Errori	Descrizione
POST	/add	Statistica	-	400 BAD_REQUEST 409 CONFLICT	Timestamp o valore negativi La statistica è già stata inserita
POST	/add/{id}	Statistica	-	400 BAD_REQUEST 409 CONFLICT	Timestamp o valore negativi La statistica è già stata inserita
GET	/get/{n}/{id}	-	List [Statistica]	400 BAD_REQUEST 404 NOT_FOUND	Id o n negativi Non ci sono statistiche della casa Nota: se ci sono meno di n stats le ritorno tutte
GET	/get/{n}	-	List [Statistica]	400 BAD_REQUEST 404 NOT_FOUND	Id o n negativi Non ci sono statistiche della casa Nota: se ci sono meno di n stats le ritorno tutte

## STATS SERVICE (2)

Metoto	Path	Input	Output	Errori	Descrizione
GET	/getAggregate/{n} /{id}	-	Aggregato	400 BAD_REQUEST 404 NOT_FOUND	Id o n negativi Non ci sono stats della casa
GET	/getAggregate/{n}	-	Aggregato	400 BAD_REQUEST 404 NOT_FOUND	Timestamp o valore negativi Non ci sono stats della casa
DELETE	/remove/{id}	-	-	400 BAD_REQUEST 404 NOT_FOUND	Id negativo La casa non esiste

Statistica: {  
    timestamp: 54696216,  
    value: 0.2599152052  
}

Aggregato: {  
    mean: 0.30380285,  
    stDev: 0,06236426  
}

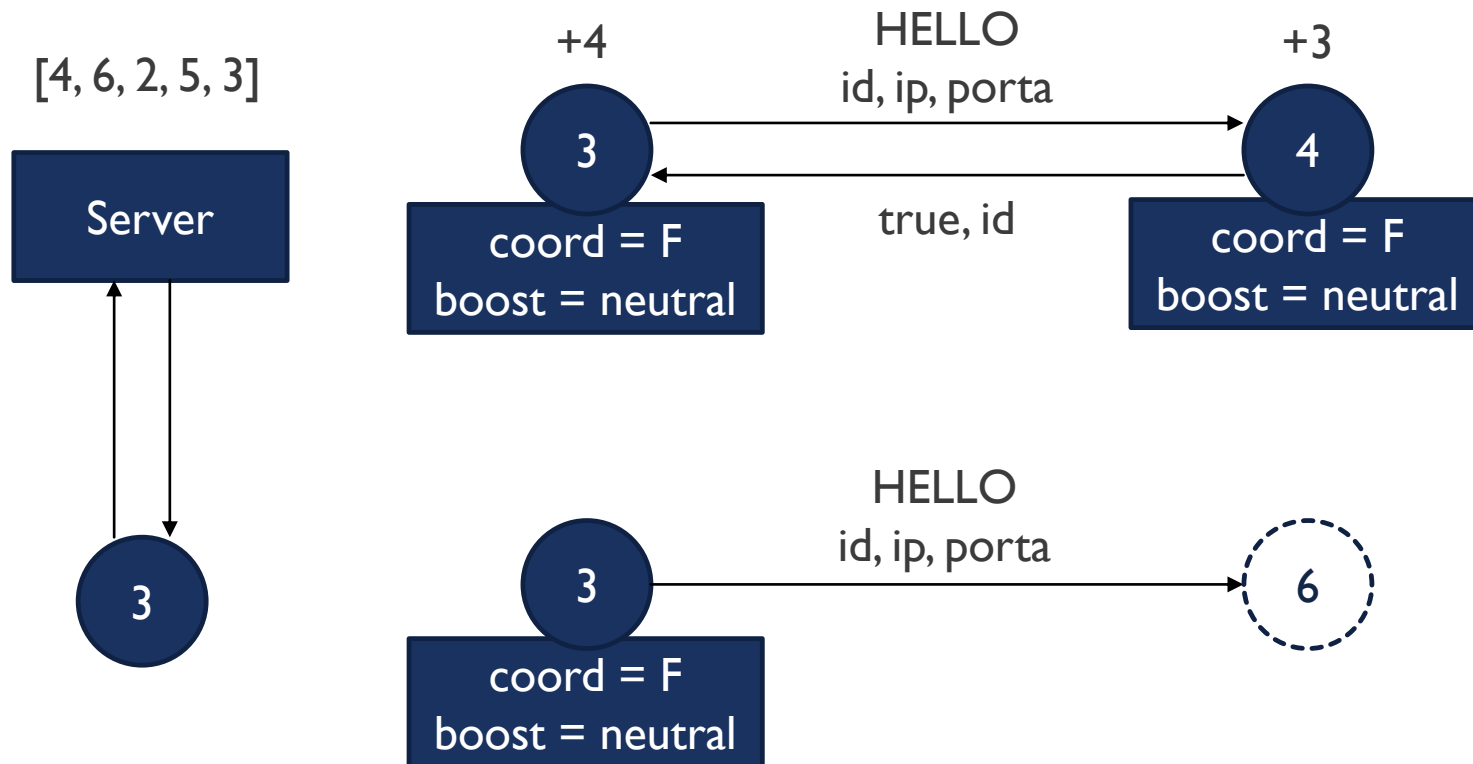


# RETE DI CASE

RETE A MAGLIA COMPLETA

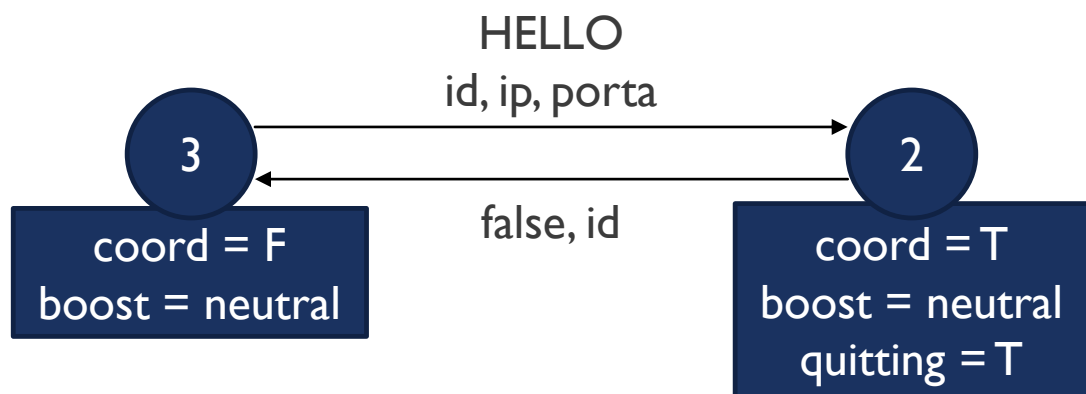


# INGRESSO NELLA RETE (I)



- 4 aggiunge 3 alla sua lista di case
- 3 aggiunge 4 alla sua lista di case
- 3 non riesce a contattare 6 (vuol dire che 6 è uscito dalla rete)
- 3 non aggiunge 6 alla sua lista di case

## INGRESSO NELLA RETE (2)



- 2 sta uscendo dalla rete ed ha indetto un'elezione
- 2 continua senza aggiungere 3 alla sua lista di case
- 3 non aggiunge 2 alla sua lista di case
- Se 2 non dovesse riuscire ad eleggere nessuno (stanno tutti uscendo), 3 diventerebbe coordinatore



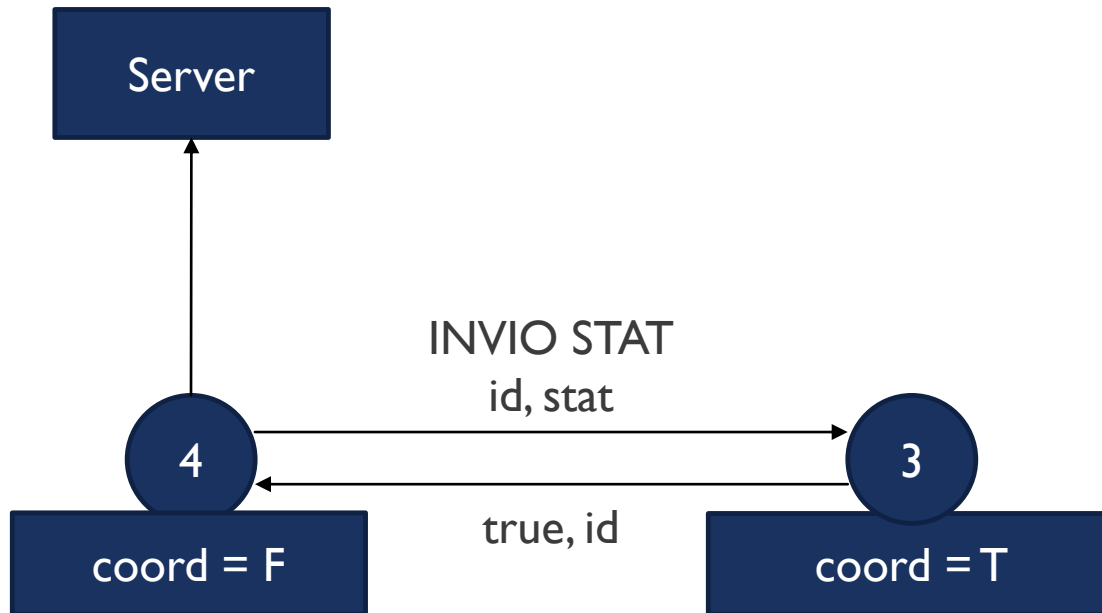
- 5 sta aspettando che gli venga concesso l'uso della corrente extra (sta aspettando un ack da tutti)
- 5 aggiunge comunque 3 alla sua lista di case (aspetta gli ack su una copia della lista di case in cui 3 non c'è)
- 3 aggiunge 5 alla sua lista di case



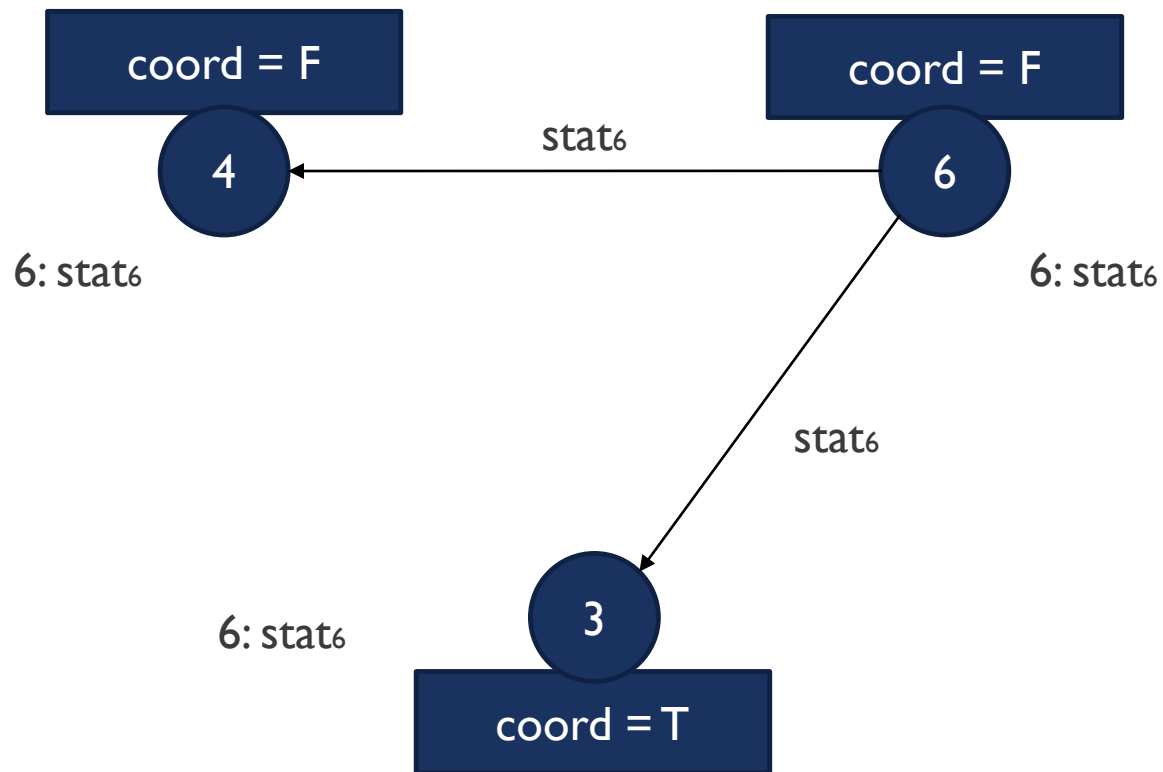
# INVIO STATISTICHE (I)

4: [stat<sub>4,1</sub>, stat<sub>4,2</sub>, ...],  
6: [stat<sub>6,1</sub>, stat<sub>6,2</sub>, ...]

- Appena 4 produce la sua nuova statistica locale, la memorizza, la invia a tutti i nodi nella rete e la invia al server amministratore

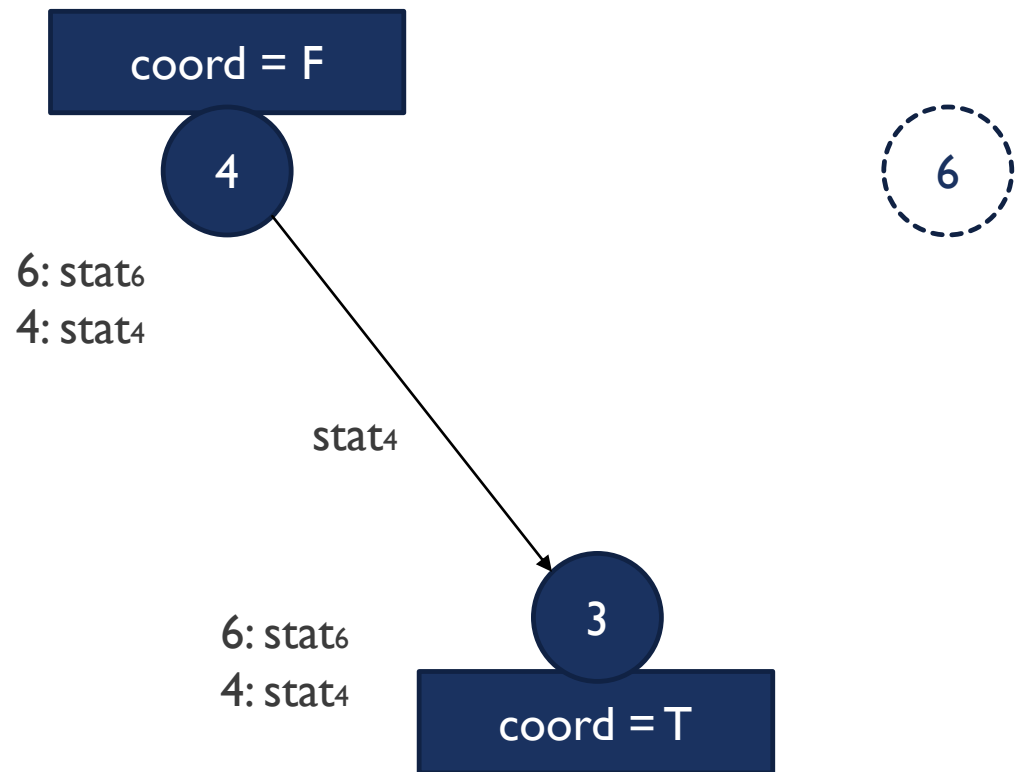


## INVIO DI STATISTICHE (2)



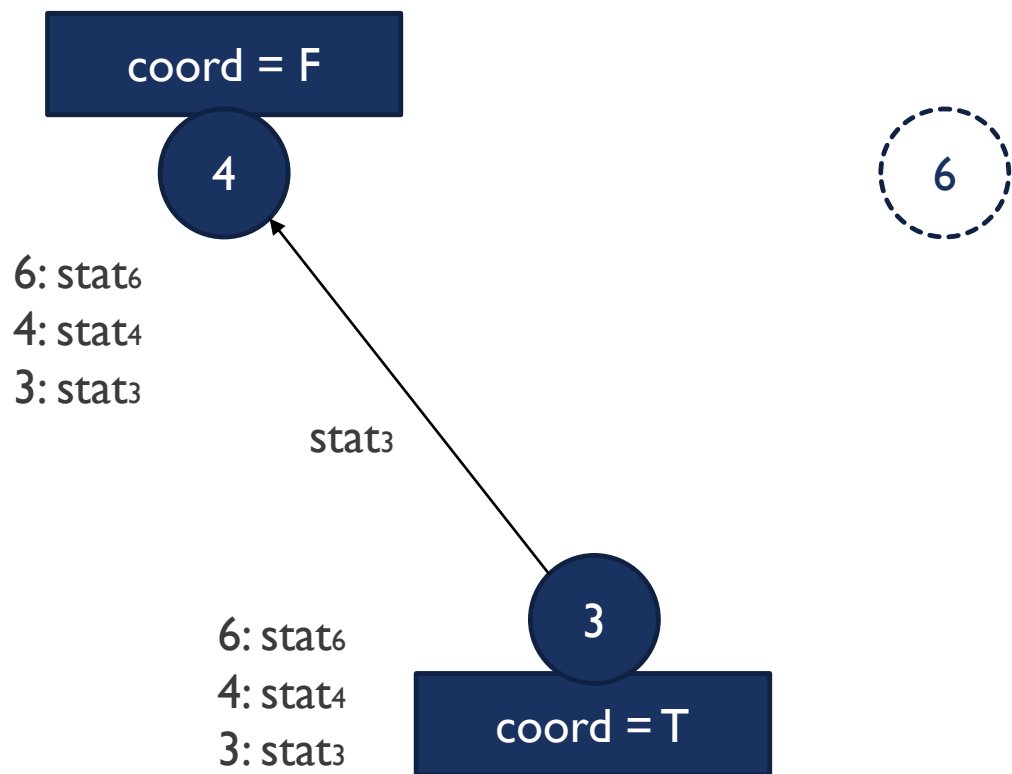
- 6 produce la statistica locale  $stat_6$  e la invia a tutti i nodi della rete

## INVIO DI STATISTICHE (3)



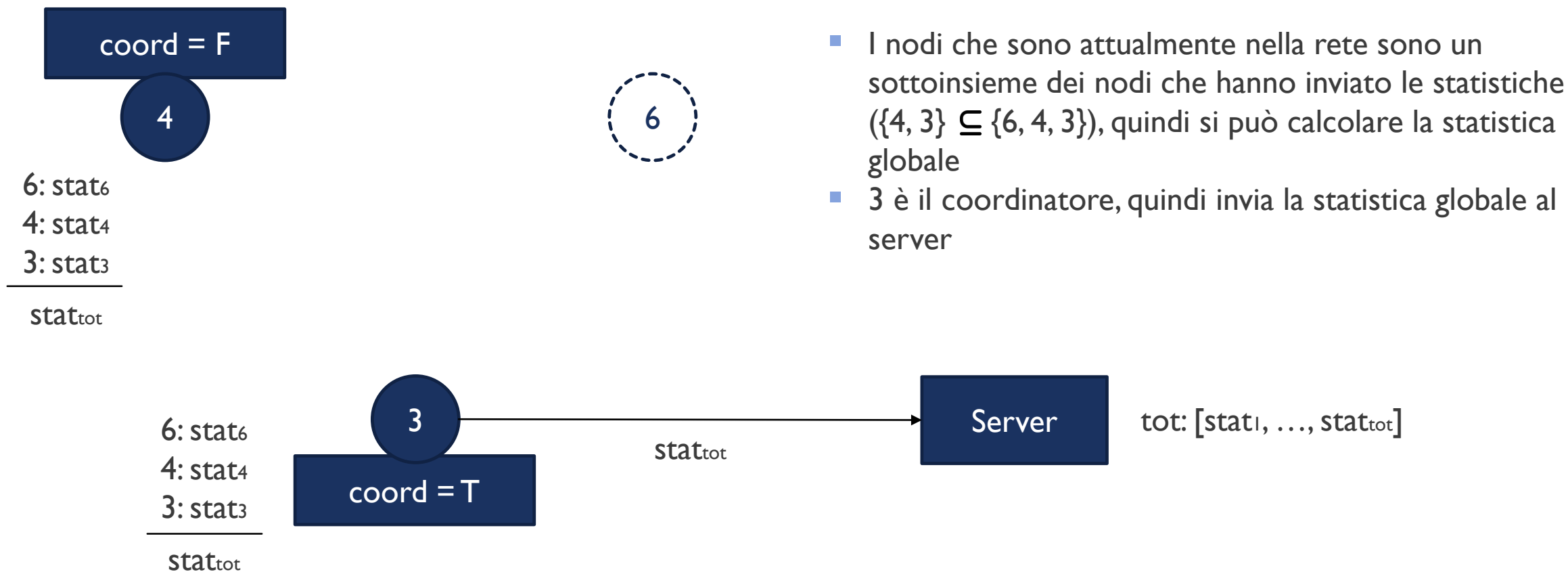
- 6 esce dalla rete (seguendo il protocollo)
- 4 produce la statistica locale `stat4` e la invia a tutti i nodi nella rete

## INVIO DI STATISTICHE (4)

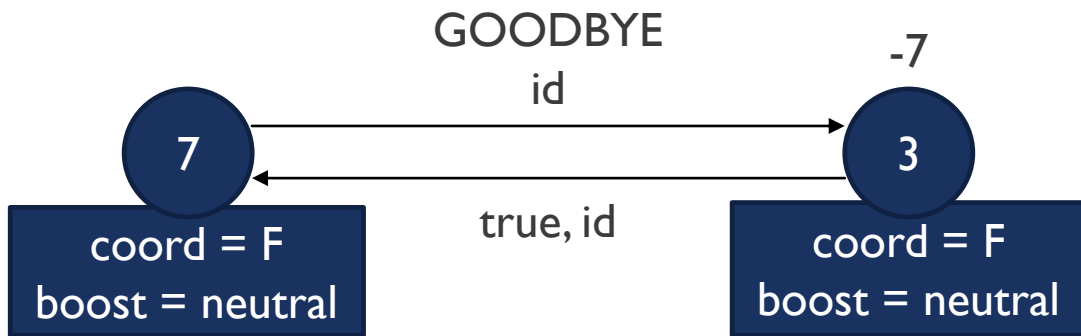


- 3 produce la statistica locale  $stat_3$  e la invia a tutti i nodi della rete

## INVIO DI STATISTICHE (5)



# USCITA DALLA RETE (I)

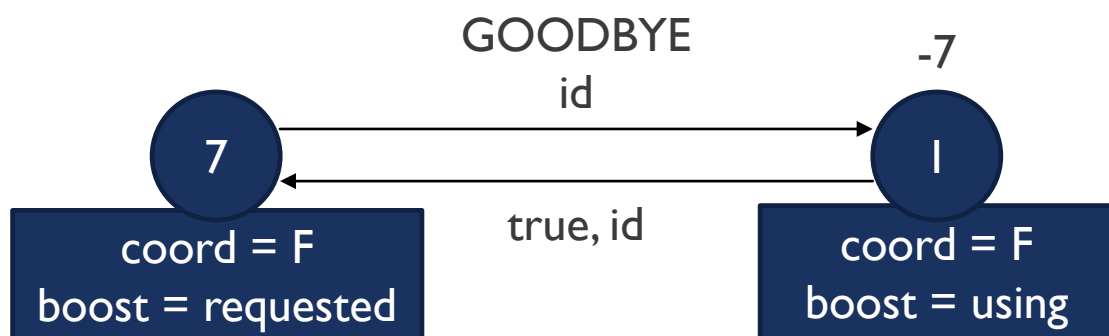


- 3 rimuove 7 dalla sua lista di case

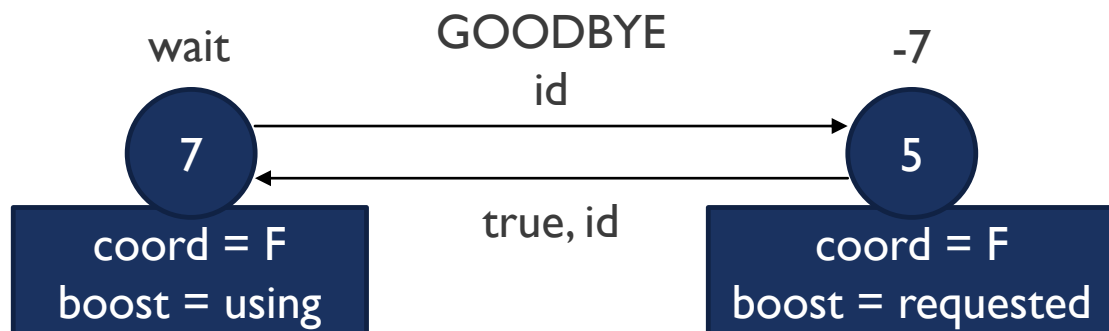


- 4 sta aspettando che gli venga concesso l'uso della corrente extra (sta aspettando un ack da tutti)
- Se 4 sta aspettando un ack da 7, considera il messaggio di uscita come un ack

## USCITA DALLA RETE (2)



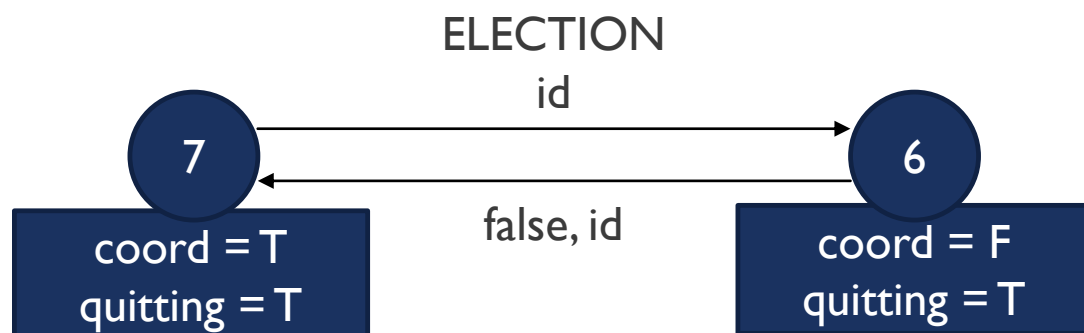
- 7 sta aspettando di usare la corrente extra, ma decide di uscire dalla rete
- 1 rimuove 7 dalla sua lista delle case: quando finirà il boost manderà un ack a 7, ma fallirà



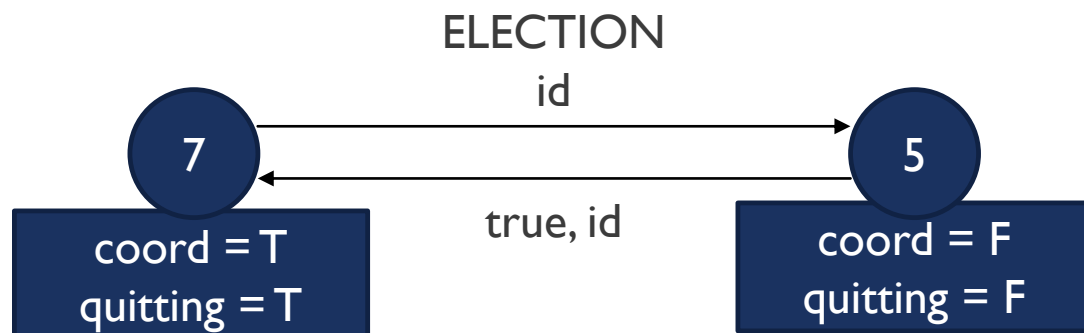
- Prima di uscire dalla rete, 7 deve aspettare di terminare il suo consumo di corrente extra
- Successivamente, esce normalmente

# ELEZIONE

Se il nodo che vuole uscire dalla rete è il coordinatore, prima di uscire indice un'elezione:



- 6 sta uscendo dalla rete, quindi non può diventare coordinatore
- 7 deve chiedere a qualcun altro: chiede a 5



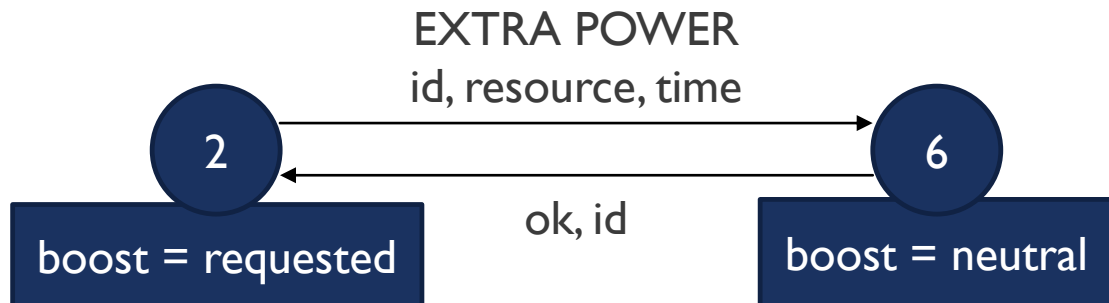
- 5 non sta uscendo dalla rete, quindi diventa coordinatore
- 7 esce normalmente dalla rete



# MUTUA ESCLUSIONE (I)

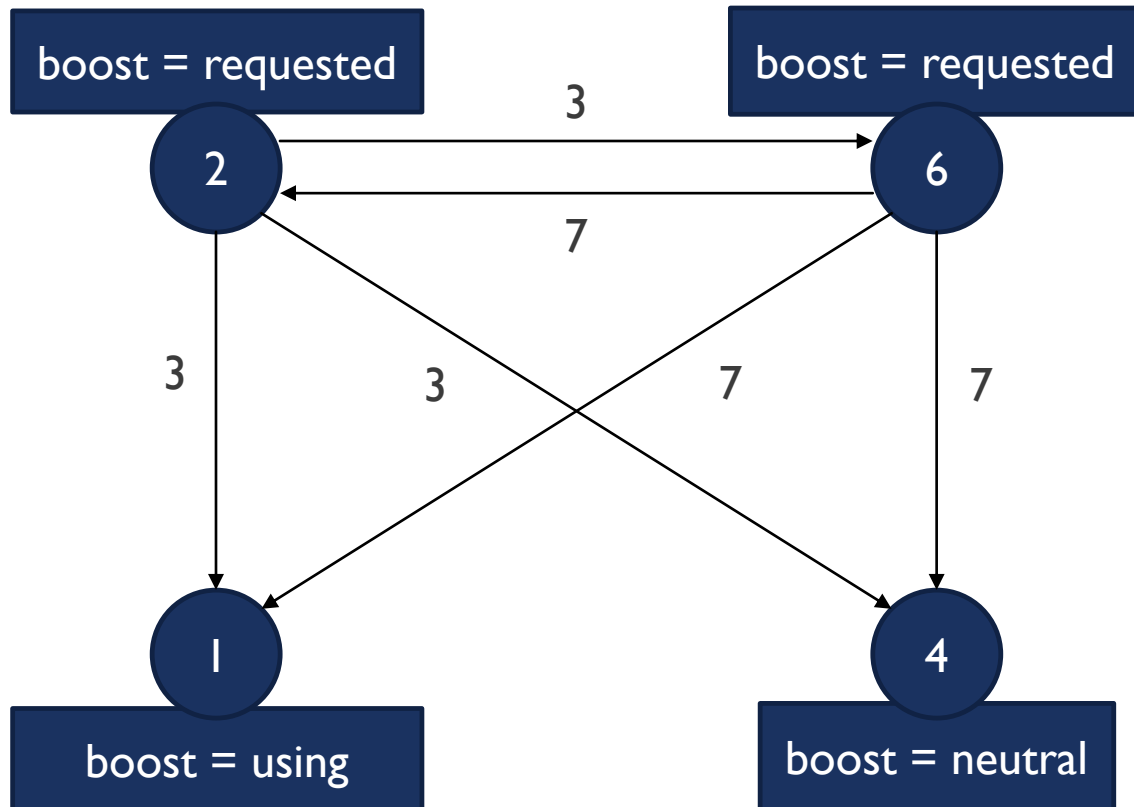
Quando un nodo fa una richiesta di corrente extra:

- Imposta la variabile `boost` a `requested`
- Fa una copia dello stato della rete in questo momento
- Memorizza il valore del clock fisico
- Invia una richiesta di corrente extra a tutti i nodi della rete



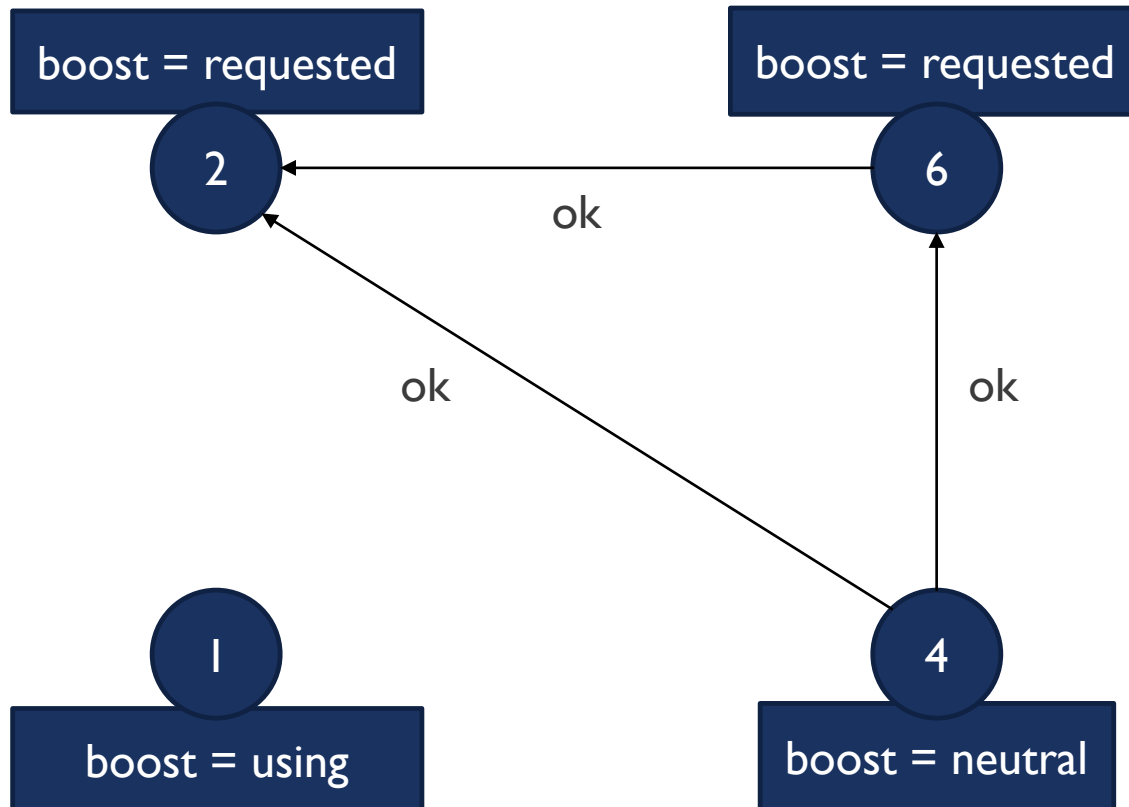
- 6 non usa né ha intenzione di usare la corrente extra, quindi risponde a 2 con un ack

## MUTUA ESCLUSIONE (2)



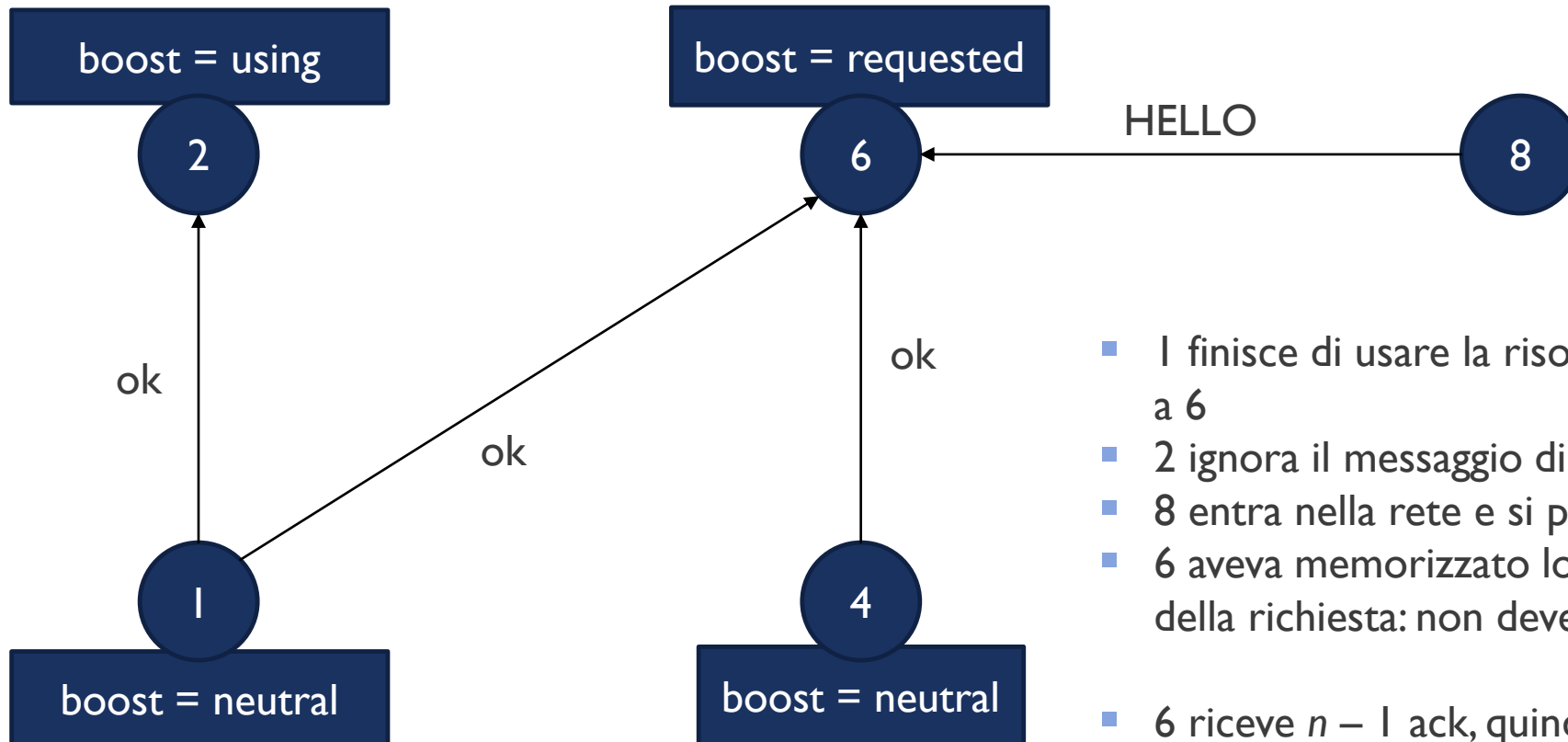
- 2 e 6 mandano in broadcast la richiesta per ottenere il boost a tutti i nodi nella copia della rete e vi scrivono il valore del clock fisico memorizzato

## MUTUA ESCLUSIONE (3)



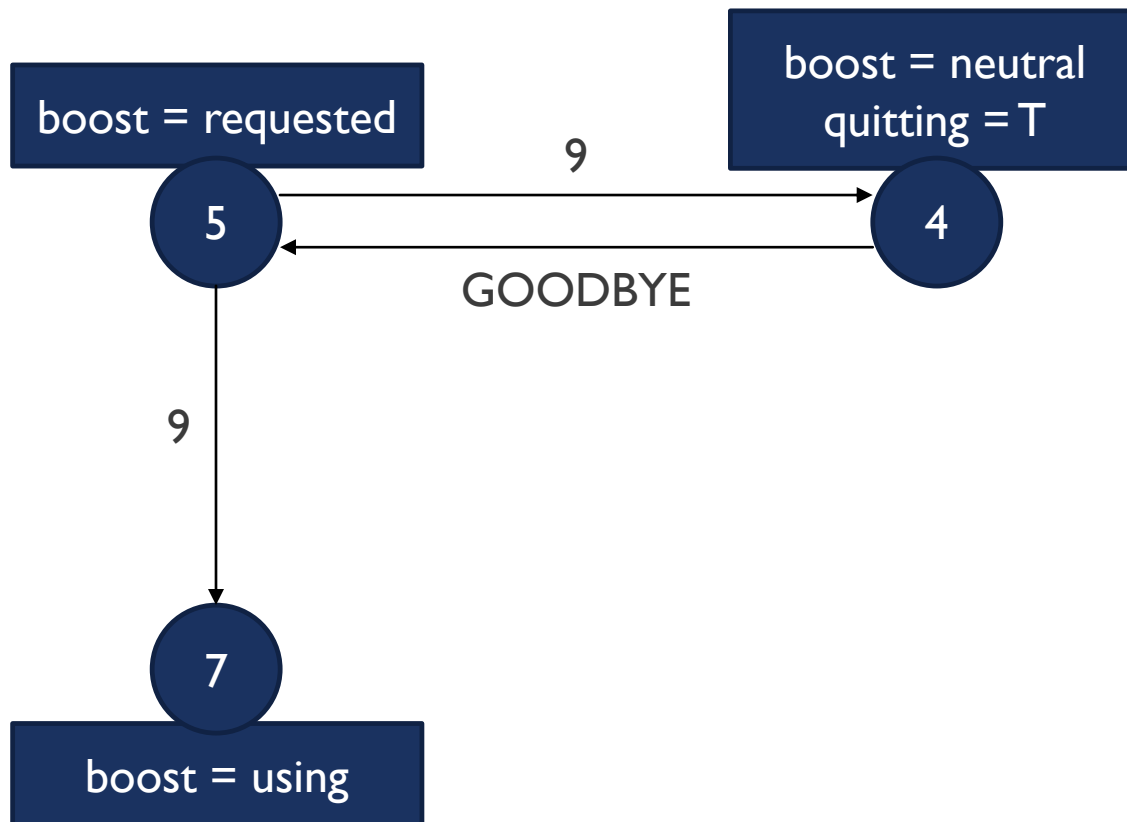
- 1 sta usando la risorsa, quindi non risponde a nessuno
- 4 è indifferente rispetto alla risorsa, quindi invia un ack sia a 2 che a 6
- 2 non risponde a 6 ( $3 < 7$ )
- 6 manda l'ack a 2 ( $7 > 3$ )
- 2 ha ricevuto l'ack da  $n - 1$  nodi (dove  $n$  è il numero di nodi a cui ha mandato il messaggio), quindi usa la corrente extra

## MUTUA ESCLUSIONE (4)



- 1 finisce di usare la risorsa, quindi manda un ack a 2 e a 6
- 2 ignora il messaggio di 1 (ha già usato la risorsa)
- 8 entra nella rete e si presenta a 6
- 6 aveva memorizzato lo stato della rete nel momento della richiesta: non deve aspettare un ack da 8
- 6 riceve  $n - 1$  ack, quindi usa la corrente extra

## MUTUA ESCLUSIONE (5)



- 5 manda una richiesta di corrente extra a 4 e a 7
- 7 sta usando la corrente extra, quindi non risponde
- 4 manda un messaggio di uscita dalla rete prima di ricevere la richiesta di 5
- Il messaggio di 4 viene considerato da 5 come un ack, quindi 5 usa la corrente extra