Computação Distribuída

Odorico Machado Mendizabal

Universidade Federal de Santa Catarina – UFSC
Departamento de Informática e Estatística – INE

Paxos

Breve contexto histórico

Leslie Lamport

Publicou The Part-Time Parliament

Barbara Liskov e Brian M-Oki

Lamport explicou o seu artigo original, “em inglês”

1998 | 1999 | 2001

Viewstamped Replication: A New Primary Copy Method to Support Highly-Available Distributed Systems

The Part-Time Parliament

LESLIE LAMPORT

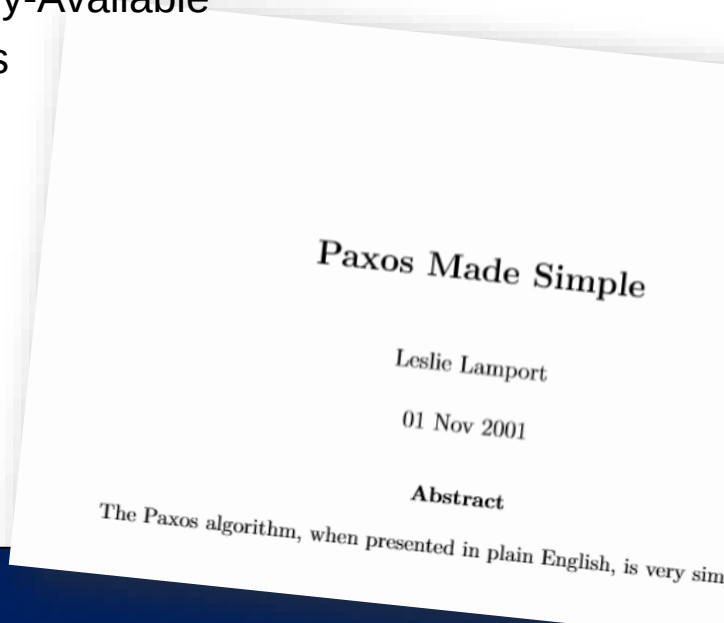
Digital Equipment Corporation

Recent archaeological discoveries on the island of Paxos reveal that the parliament functioned despite the peripatetic propensity of its part-time legislators. The legislators maintained consistent copies of the parliamentary record, despite their frequent forays from the chamber and the forgetfulness of their messengers. The Paxon parliament's protocol provides a new way of implementing the state-machine approach to the design of distributed systems.

Categories and Subject Descriptors: C2.4 [Computer-Communications Networks]: Distributed Systems—Network operating systems; D4.5 [Operating Systems]: Reliability—Fault-tolerance; J.1 [Administrative Data Processing]: Government

General Terms: Design, Reliability

Additional Key Words and Phrases: State machines, three-phase commit, voting



Paxos Made Simple

Leslie Lamport

01 Nov 2001

Abstract

The Paxos algorithm, when presented in plain English, is very sim

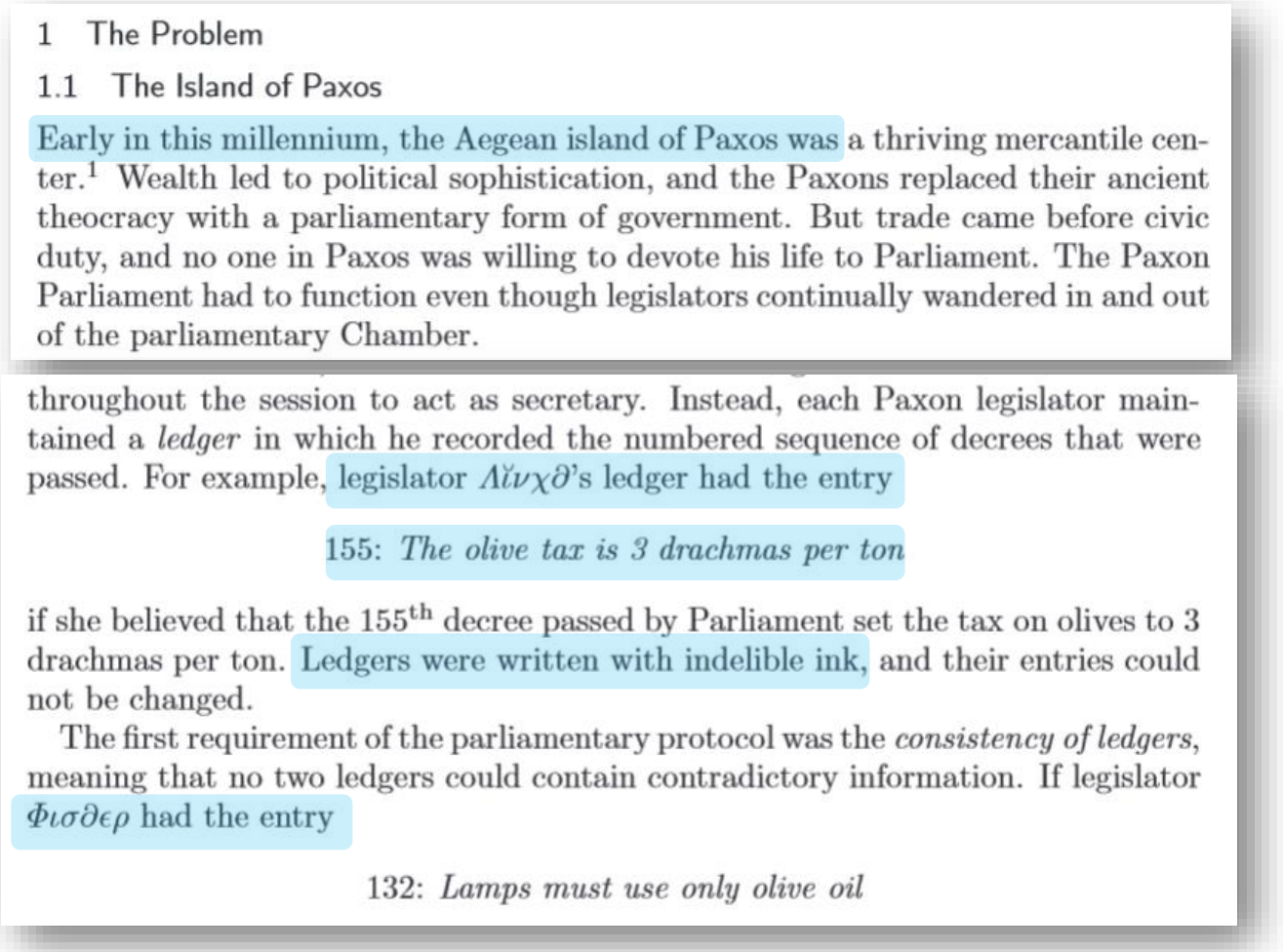
Paxos

- *Definição de papéis:*
  - *Proposer*
  - *Acceptor*
  - *Learner*

Paxos

- *Definição de papéis:*
  - *Proposer*
  - *Acceptor*
  - *Learner*

Estilo apresentado no artigo original:



1 The Problem

1.1 The Island of Paxos

Early in this millennium, the Aegean island of Paxos was a thriving mercantile center.[1] Wealth led to political sophistication, and the Paxons replaced their ancient theocracy with a parliamentary form of government. But trade came before civic duty, and no one in Paxos was willing to devote his life to Parliament. The Paxon Parliament had to function even though legislators continually wandered in and out of the parliamentary Chamber.

throughout the session to act as secretary. Instead, each Paxon legislator maintained a *ledger* in which he recorded the numbered sequence of decrees that were passed. For example, legislator $\Lambda\check{\imath}\nu\chi\partial$'s ledger had the entry

155: *The olive tax is 3 drachmas per ton*

if she believed that the 155$^{\text{th}}$ decree passed by Parliament set the tax on olives to 3 drachmas per ton. Ledgers were written with indelible ink, and their entries could not be changed.

The first requirement of the parliamentary protocol was the *consistency of ledgers*, meaning that no two ledgers could contain contradictory information. If legislator $\Phi\iota\sigma\partial\epsilon\rho$ had the entry

132: *Lamps must use only olive oil*

Paxos

Apresentação do algoritmo no artgo “Paxos Made Simple”

Phase 1. (a) A proposer selects a proposal number n and sends a *prepare* request with number n to a majority of acceptors.

(b) If an acceptor receives a *prepare* request with number n greater than that of any *prepare* request to which it has already responded, then it responds to the request with a promise not to accept any more proposals numbered less than n and with the highest-numbered proposal (if any) that it has accepted.

Phase 2. (a) If the proposer receives a response to its *prepare* requests (numbered n) from a majority of acceptors, then it sends an *accept* request to each of those acceptors for a proposal numbered n with a value v, where v is the value of the highest-numbered proposal among the responses, or is any value if the responses reported no proposals.

(b) If an acceptor receives an *accept* request for a proposal numbered n, it accepts the proposal unless it has already responded to a *prepare* request having a number greater than n.

Paxos

Apresentação do algoritmo no artgo “Paxos Made Simple”

Phase 1. (a) A proposer selects a proposal number n and sends a *prepare* request with number n to a majority of acceptors.

(b) If an acceptor receives a *prepare* request with number n greater than that of any *prepare* request to which it has already responded, then it responds to the request with a promise not to accept any more proposals numbered less than n and with the highest-numbered proposal (if any) that it has accepted.

Phase 2. (a) If the proposer receives a response to its *prepare* requests (numbered n) from a majority of acceptors, then it sends an *accept* request to each of those acceptors for a proposal numbered n with a value v, where v is the value of the highest-numbered proposal among the responses, or is any value if the responses reported no proposals.

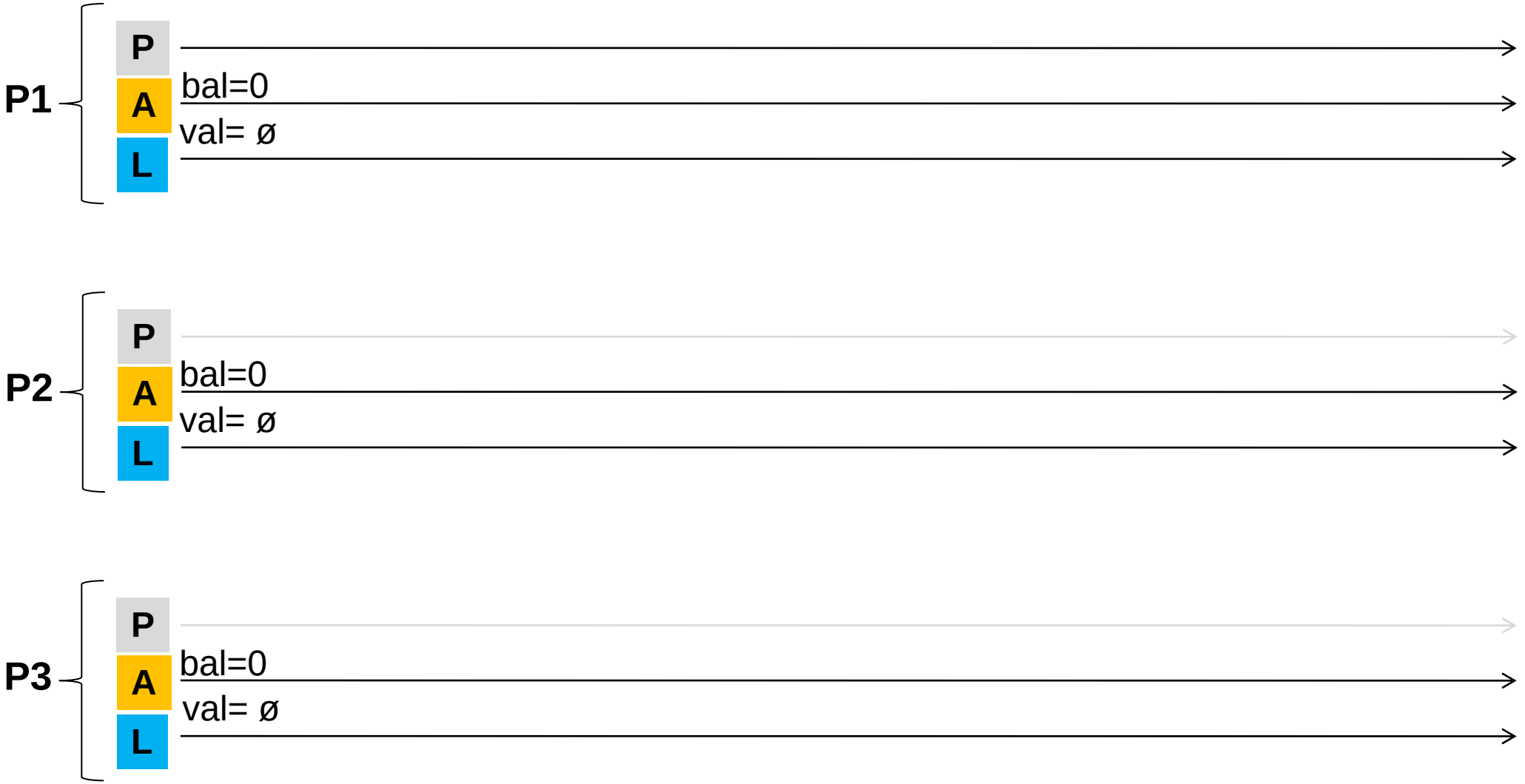
(b) If an acceptor receives an *accept* request for a proposal numbered n, it accepts the proposal unless it has already responded to a *prepare* request having a number greater than n.

- *Acceptors* nunca se comprometem com instâncias antigas – eles sempre participam em decisões com instâncias mais novas;
- *Acceptors* nunca “mudam de ideia”. Se um valor já foi aceito, eles sempre responderão solicitações de *prepare* com o valor já aceito

- *Proposers* enviam requisições *accept* apenas quando uma maioria de *acceptors* respondem a requisição *prepare*;
- *Proposers* enviam requisições *accept* com valor v, onde v é um valor já aceito por algum *acceptor* (o de maior instância entre os aceitos) ou é um valor qualquer;

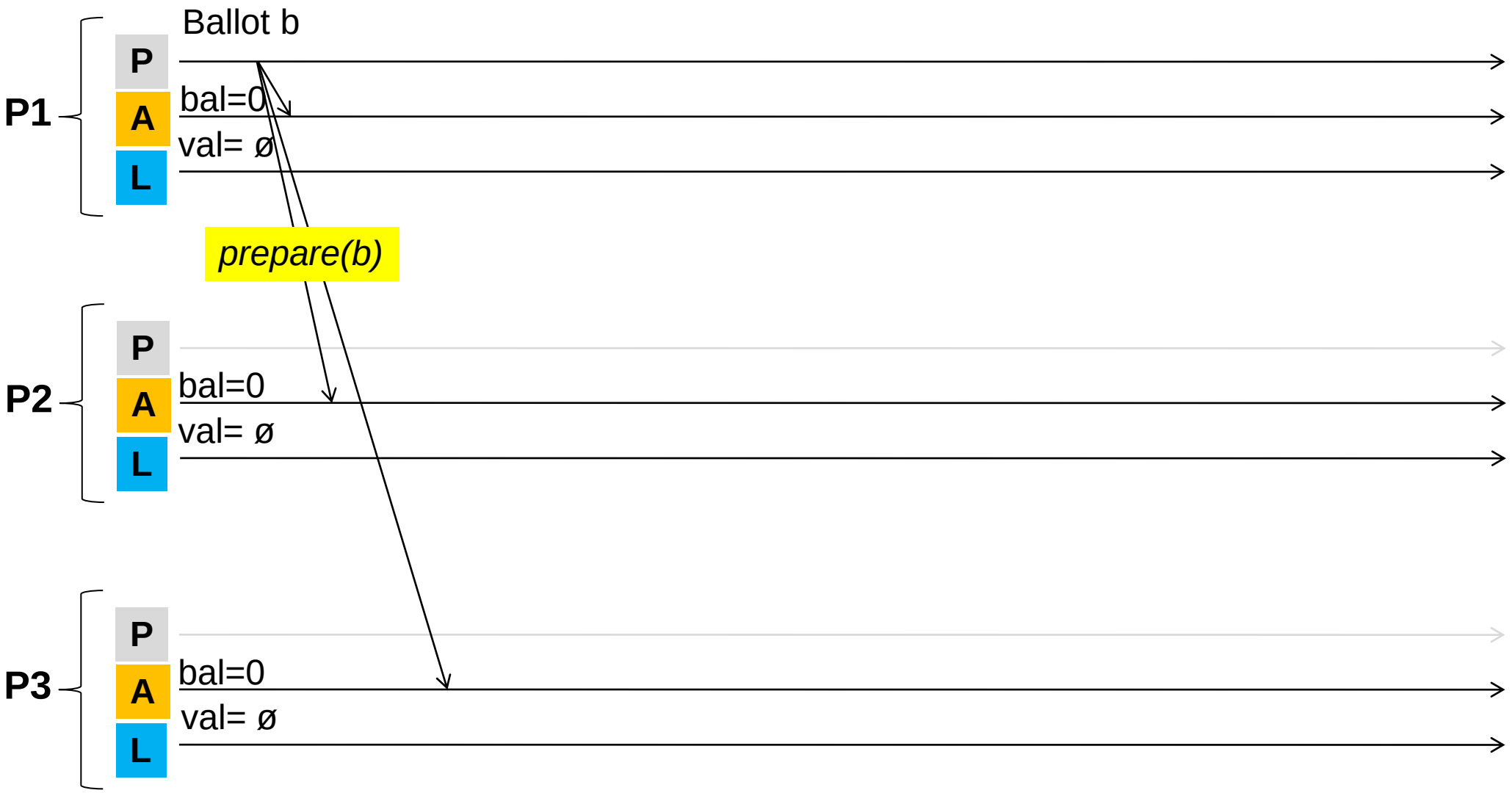
Paxos: Proposta de um valor

Proposta do valor v por P1



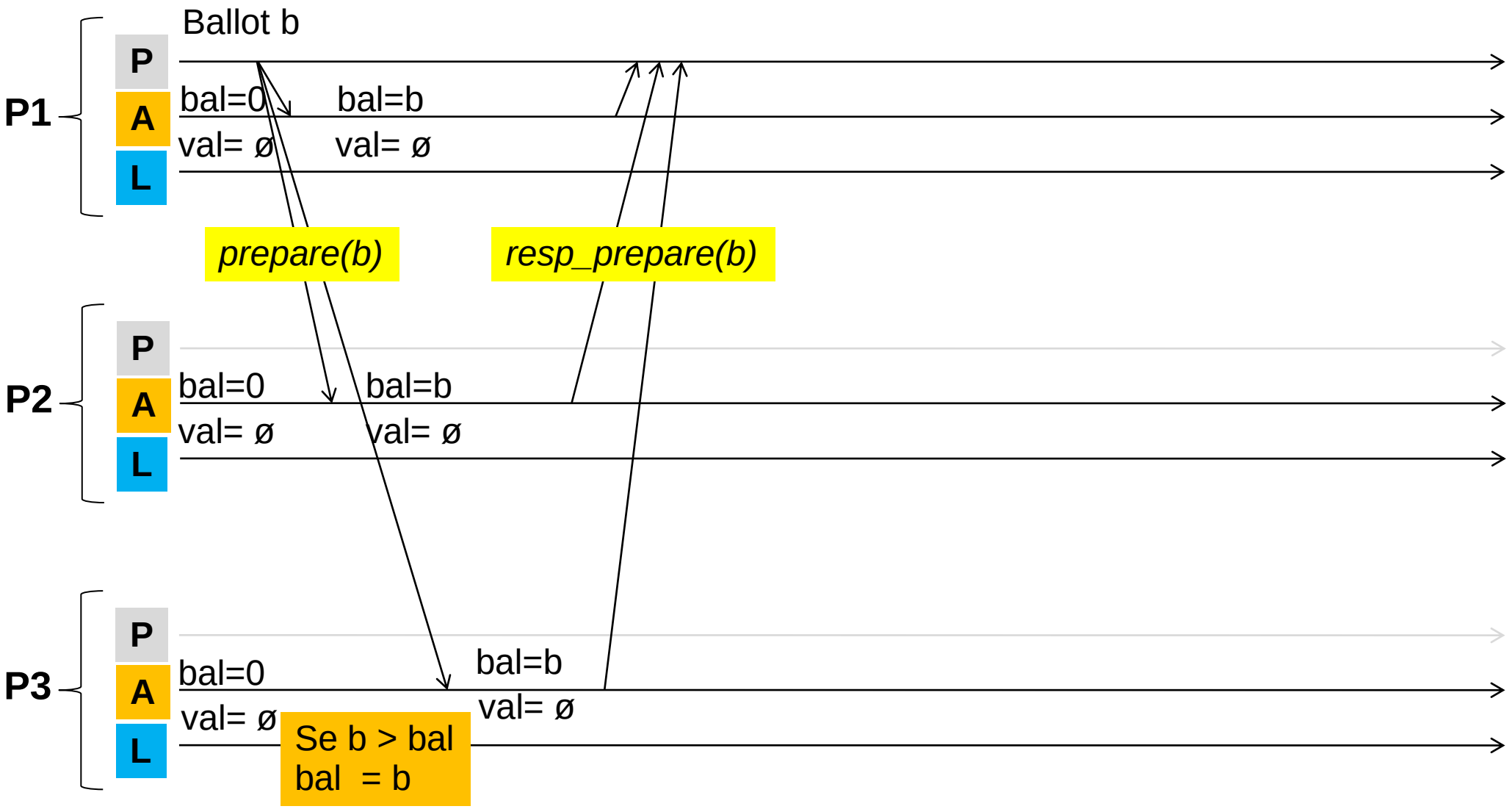
Paxos: Proposta de um valor

Proposta do valor v por P1



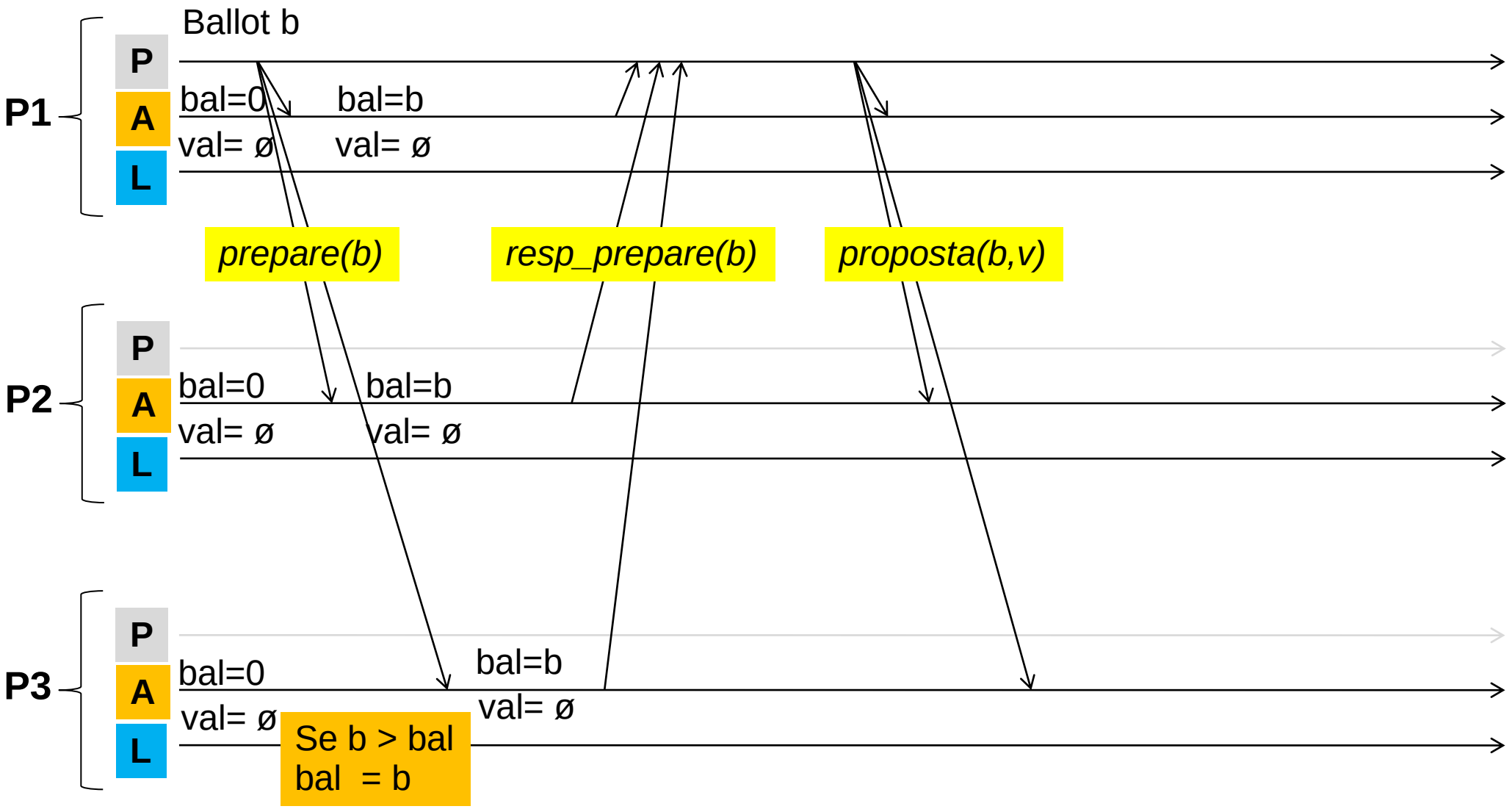
Paxos: Proposta de um valor

Proposta do valor v por P1



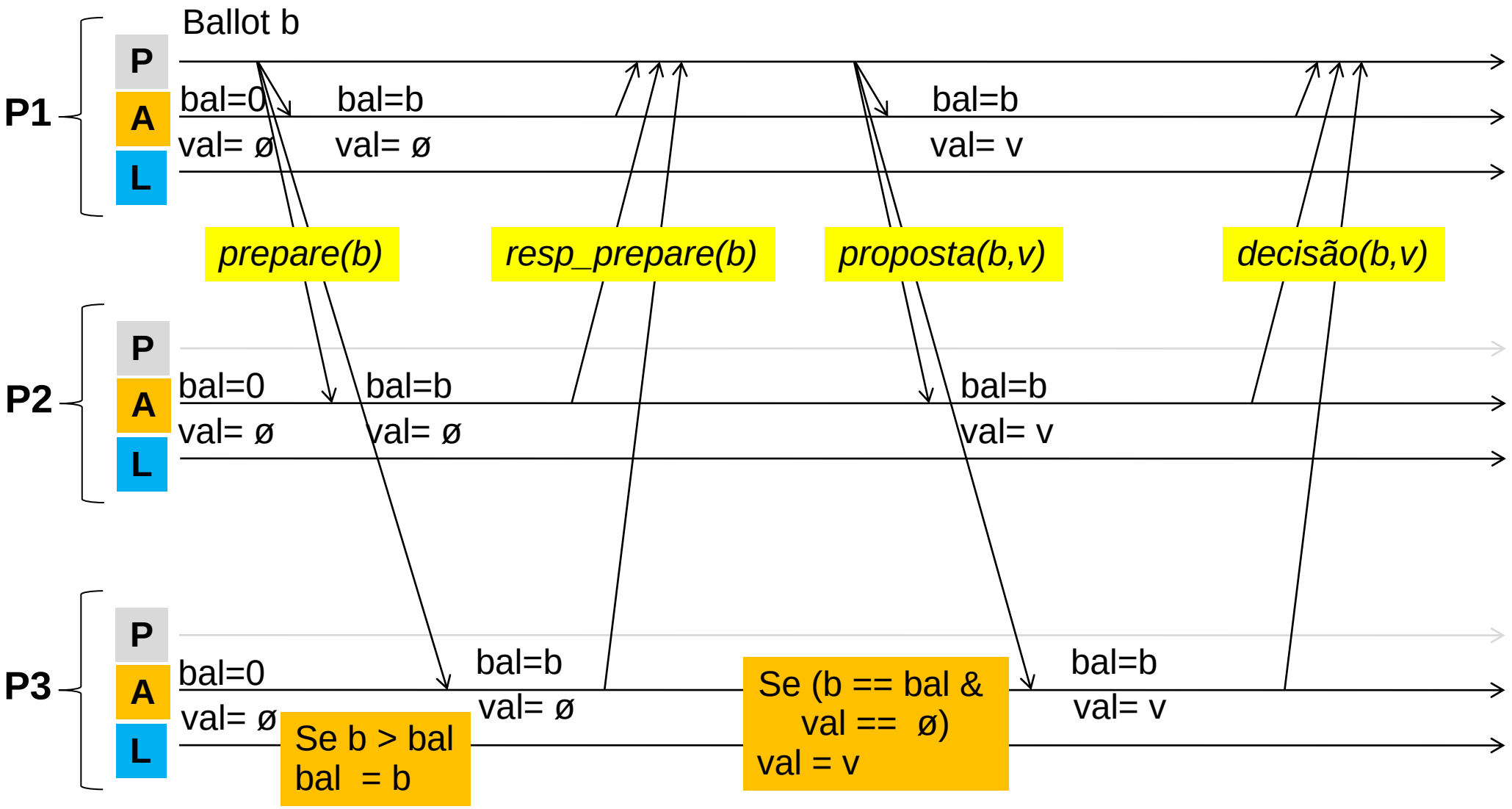
Paxos: Proposta de um valor

Proposta do valor v por P1



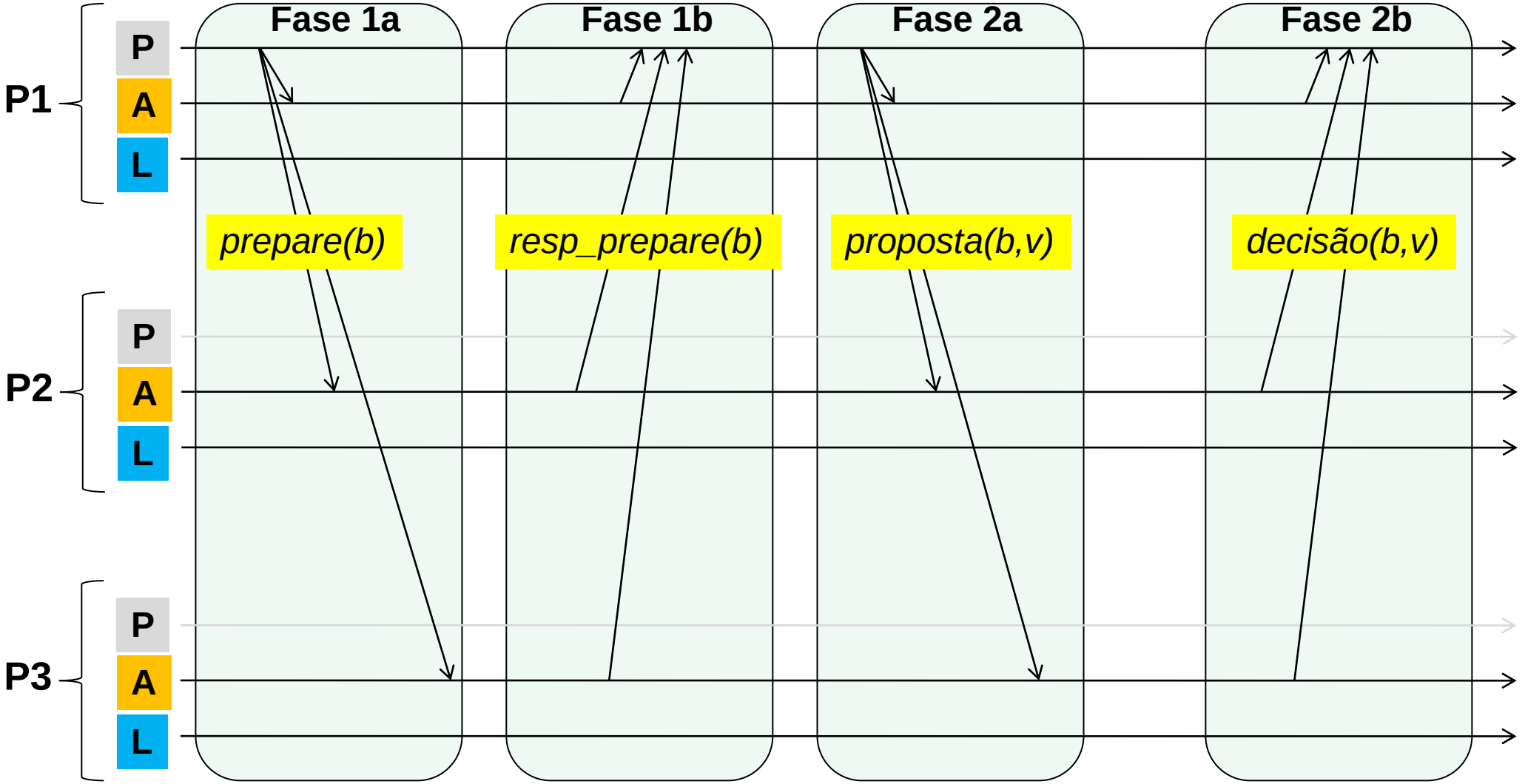
Paxos: Proposta de um valor

Proposta do valor v por P1



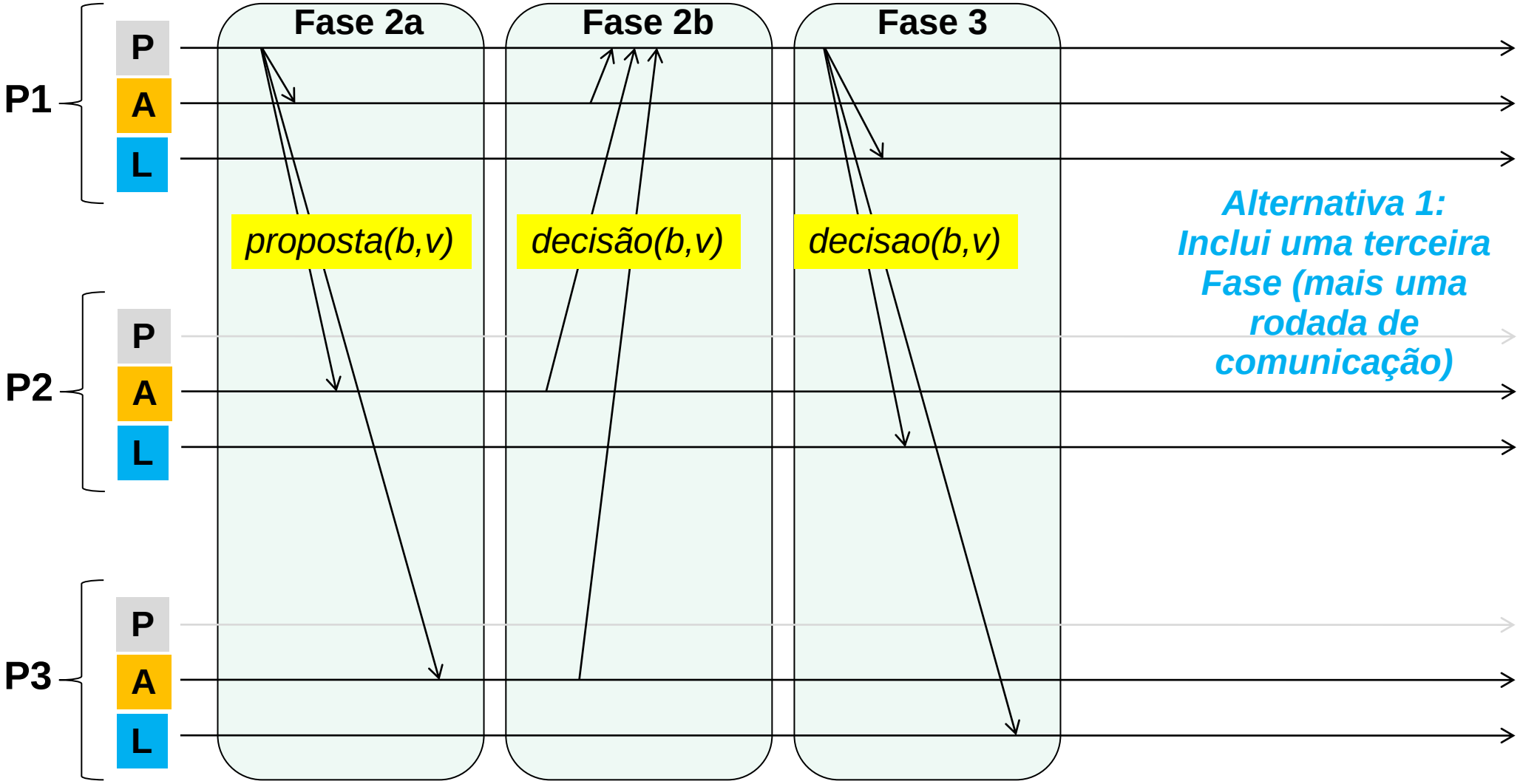
Paxos: Proposta de um valor

Proposta do valor v por P1

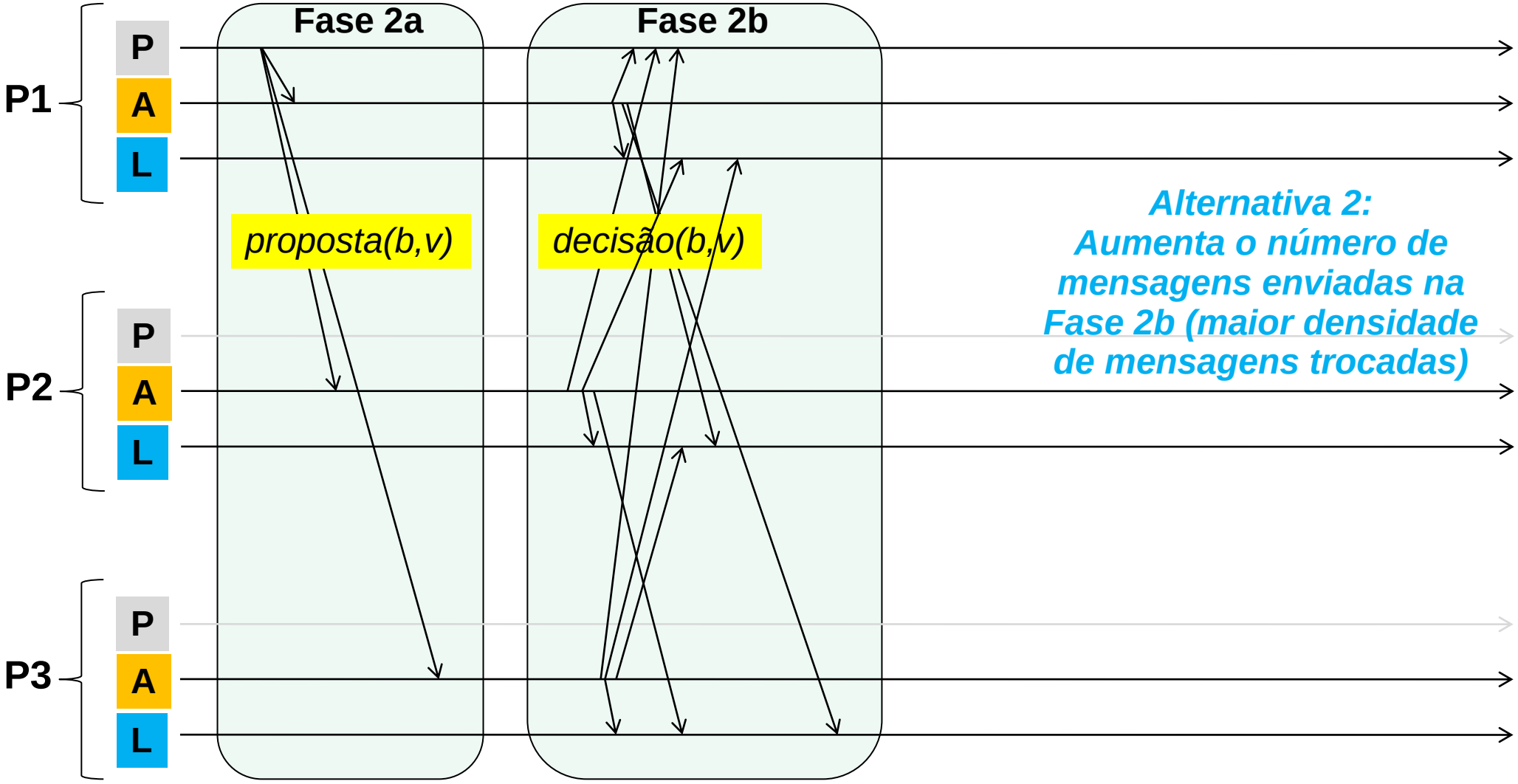


Paxos: Aprendendo um valor decidido (alternativa 1)

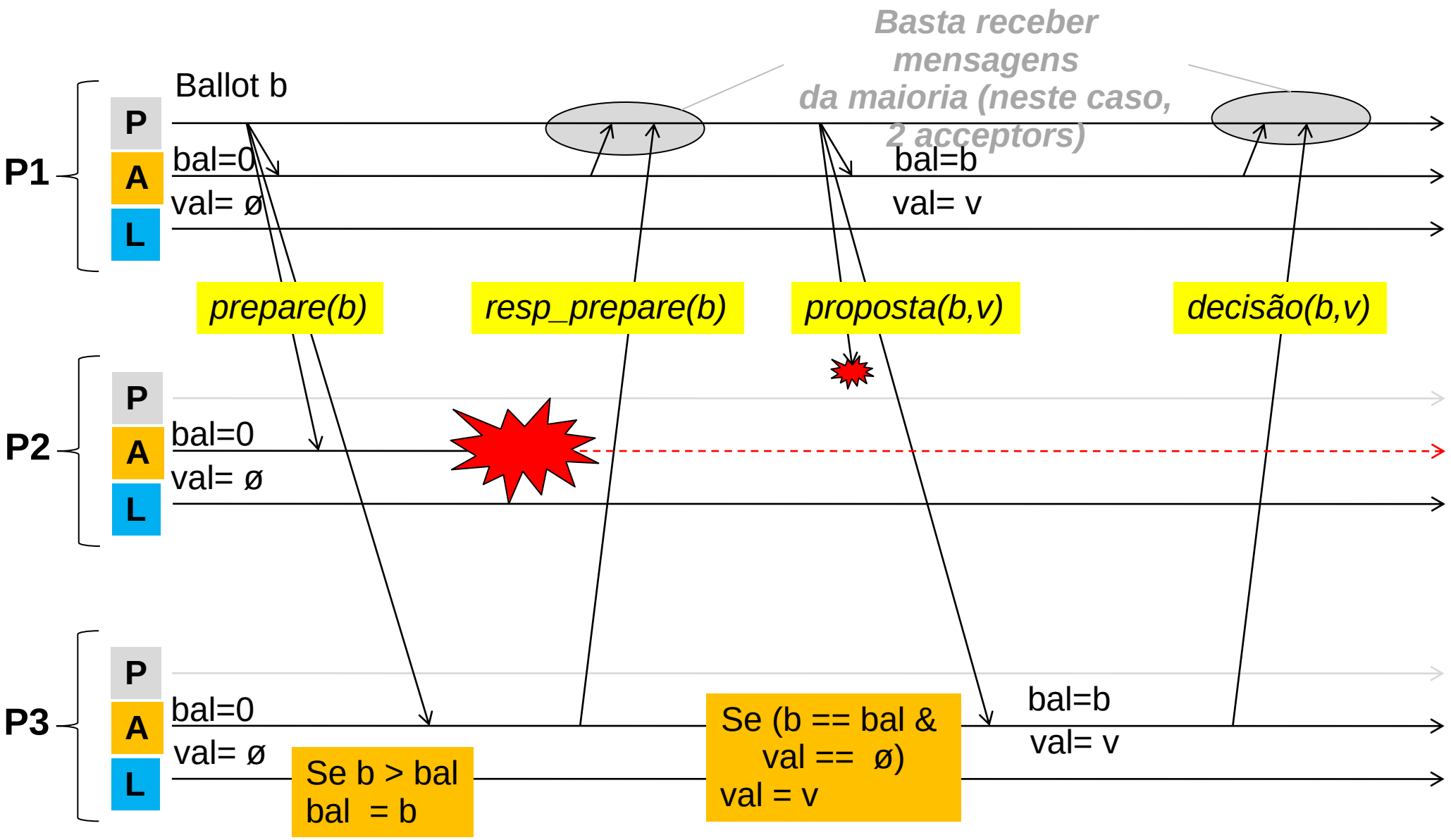
Proposer* repassa valor decidido aos *learners



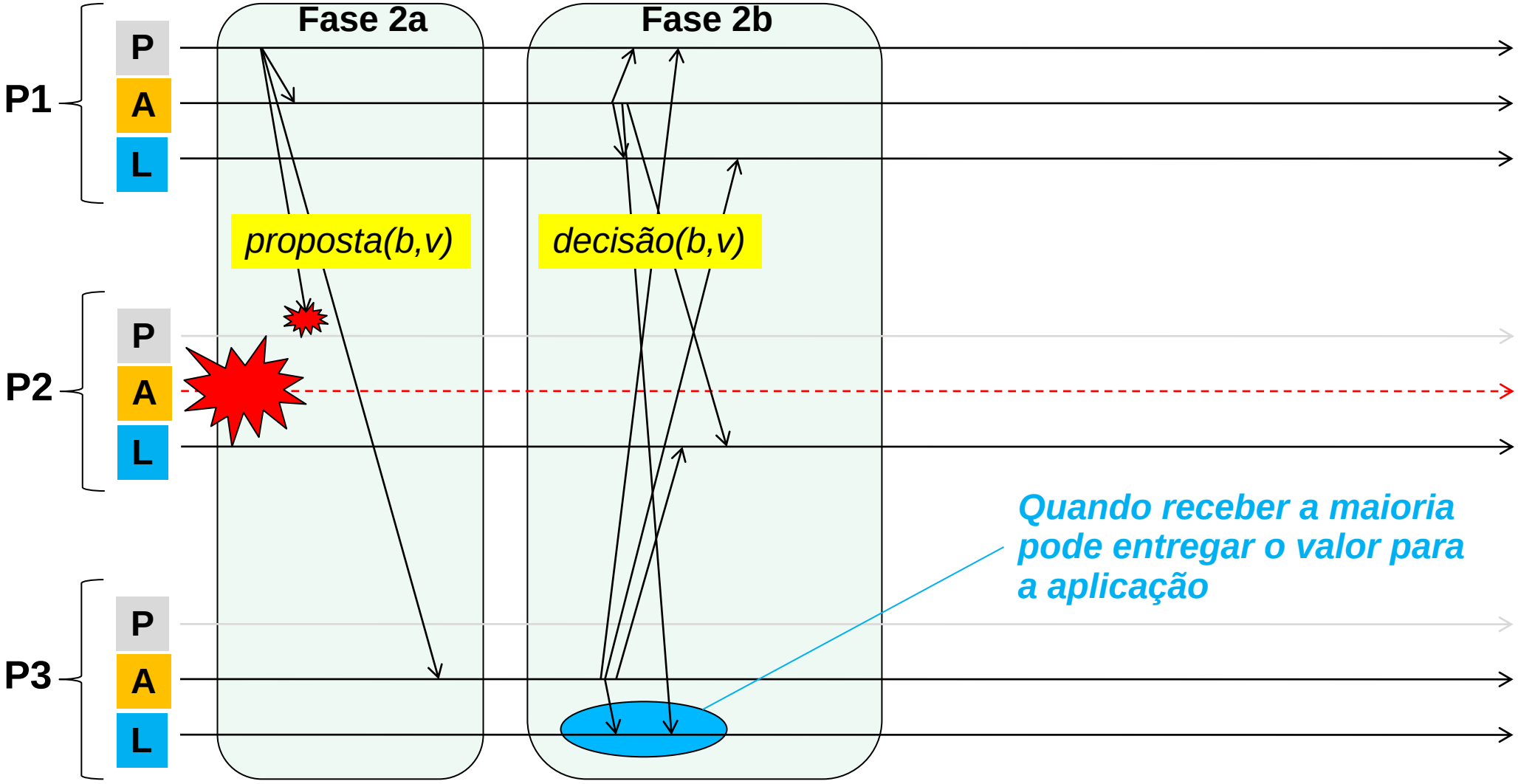
Paxos: Aprendendo um valor decidido (alternativa 2)



Paxos: Proposta de um valor (com falha de *acceptor*)

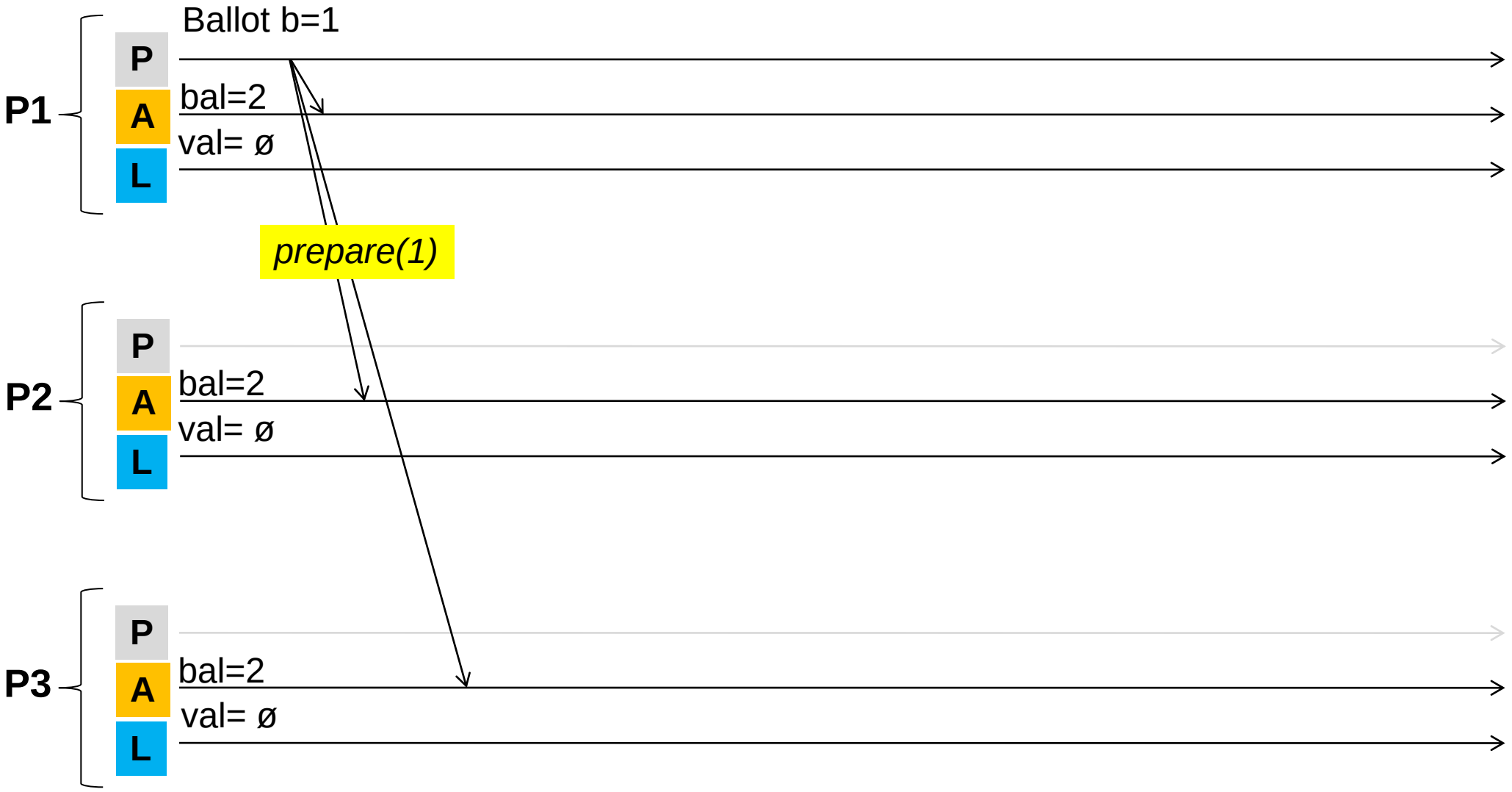


Paxos: Aprendendo um valor (com falha de *acceptor*)



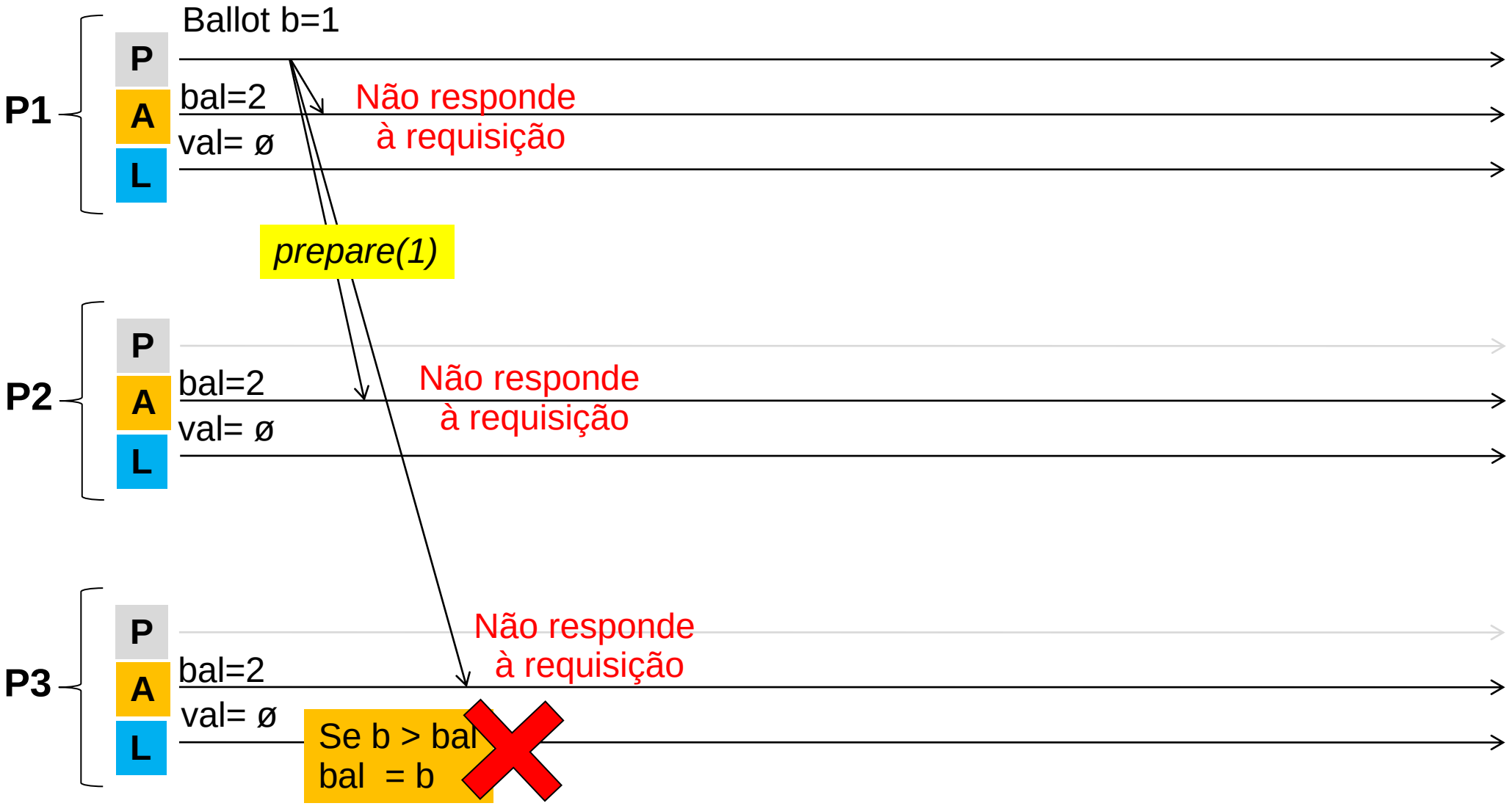
Paxos: Tratando propostas de instâncias antigas

Proposer P1 envia ballot antigo (valor de instância menor do que alguma já aceita)



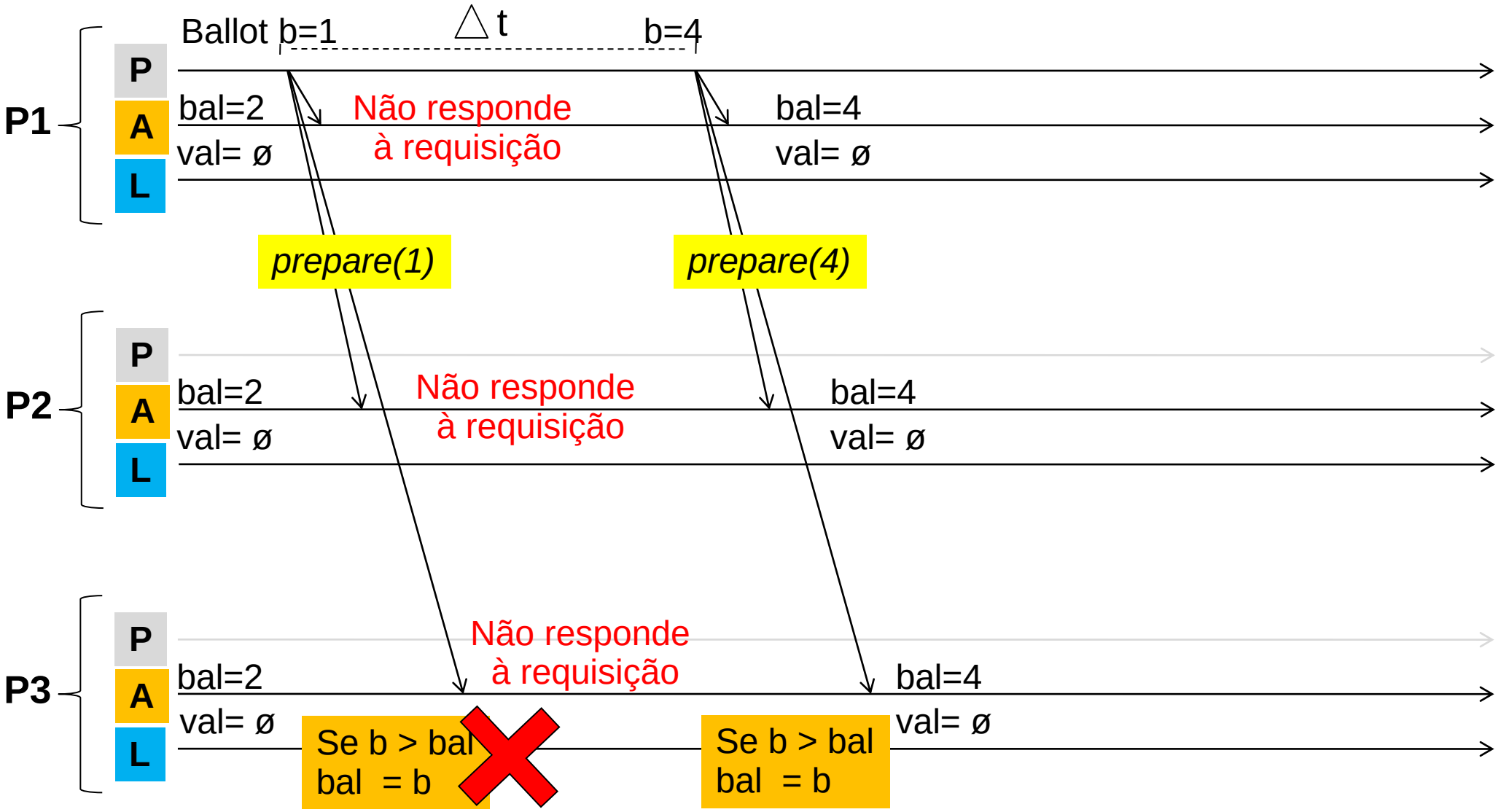
Paxos: Tratando propostas de instâncias antigas

Proposer P1 envia ballot antigo (valor de instância menor do que alguma já aceita)



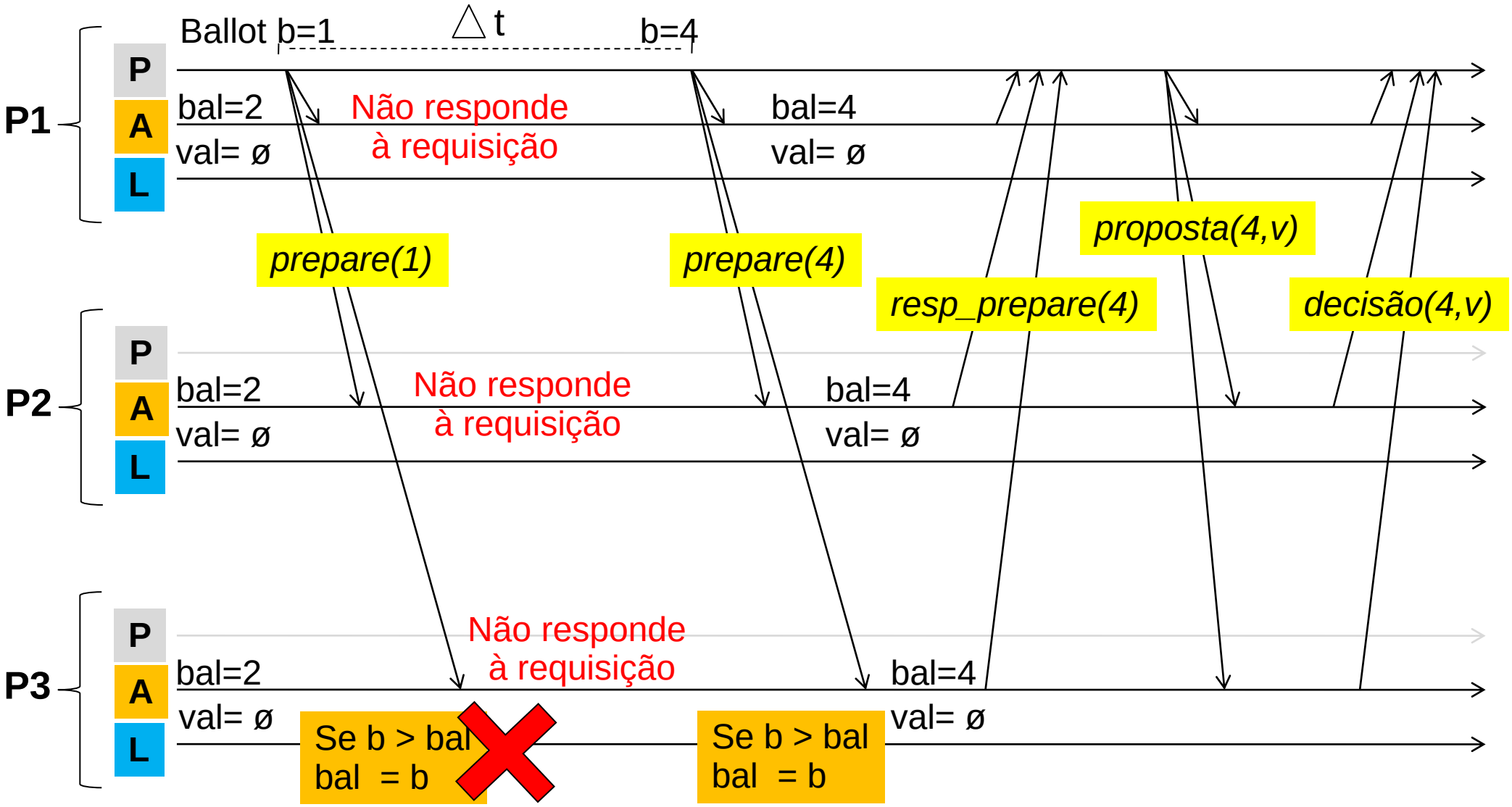
Paxos: Tratando propostas de instâncias antigas

Proposer P1 envia ballot antigo (valor de instância menor do que alguma já aceita)



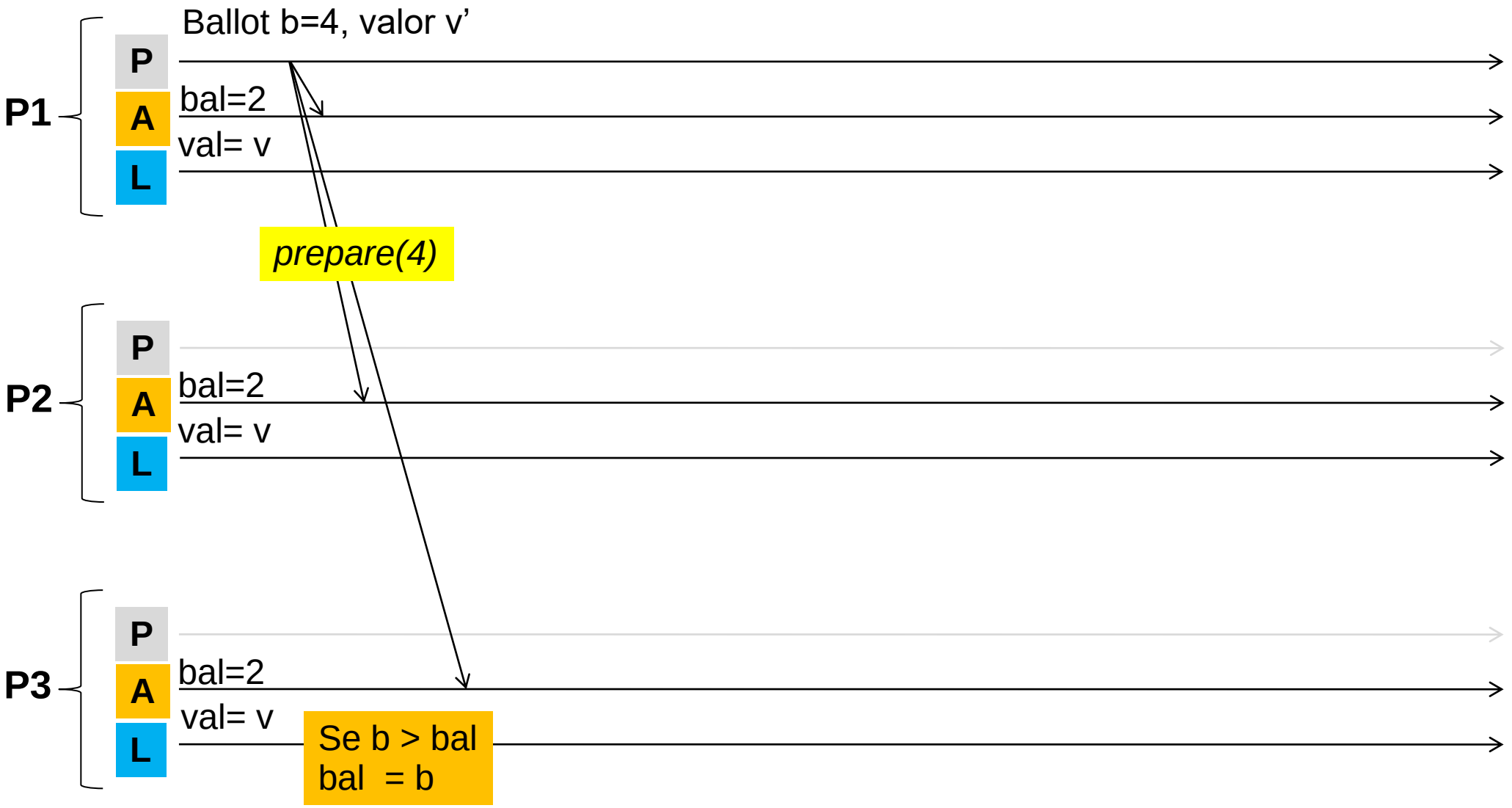
Paxos: Tratando propostas de instâncias antigas

Proposer P1 envia ballot antigo (valor de instância menor do que alguma já aceita)



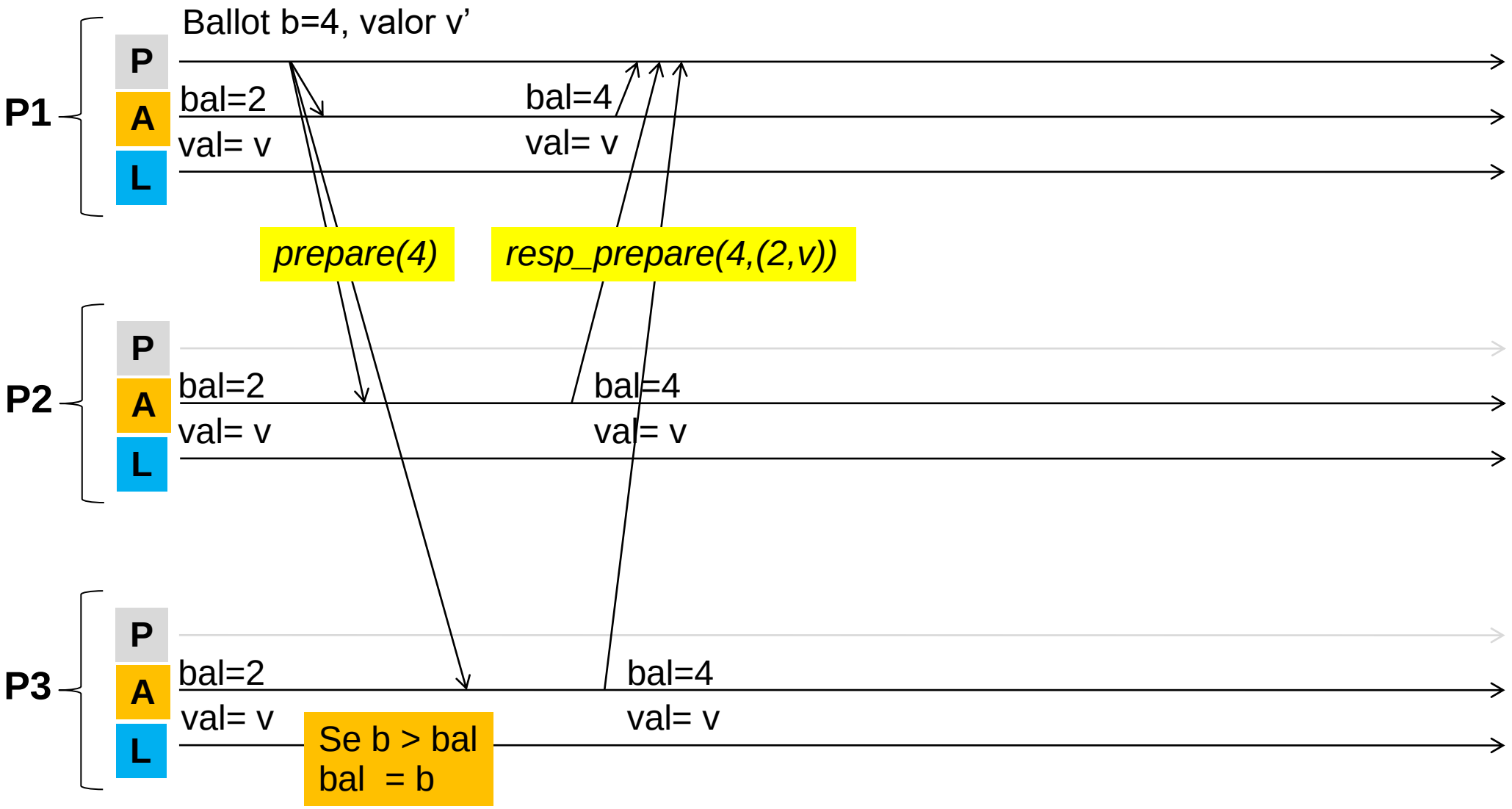
Paxos: Tratando propostas com valor já aceito

Proposer P1 envia ballot de próxima instância, mas valor já foi aceito em instância anterior



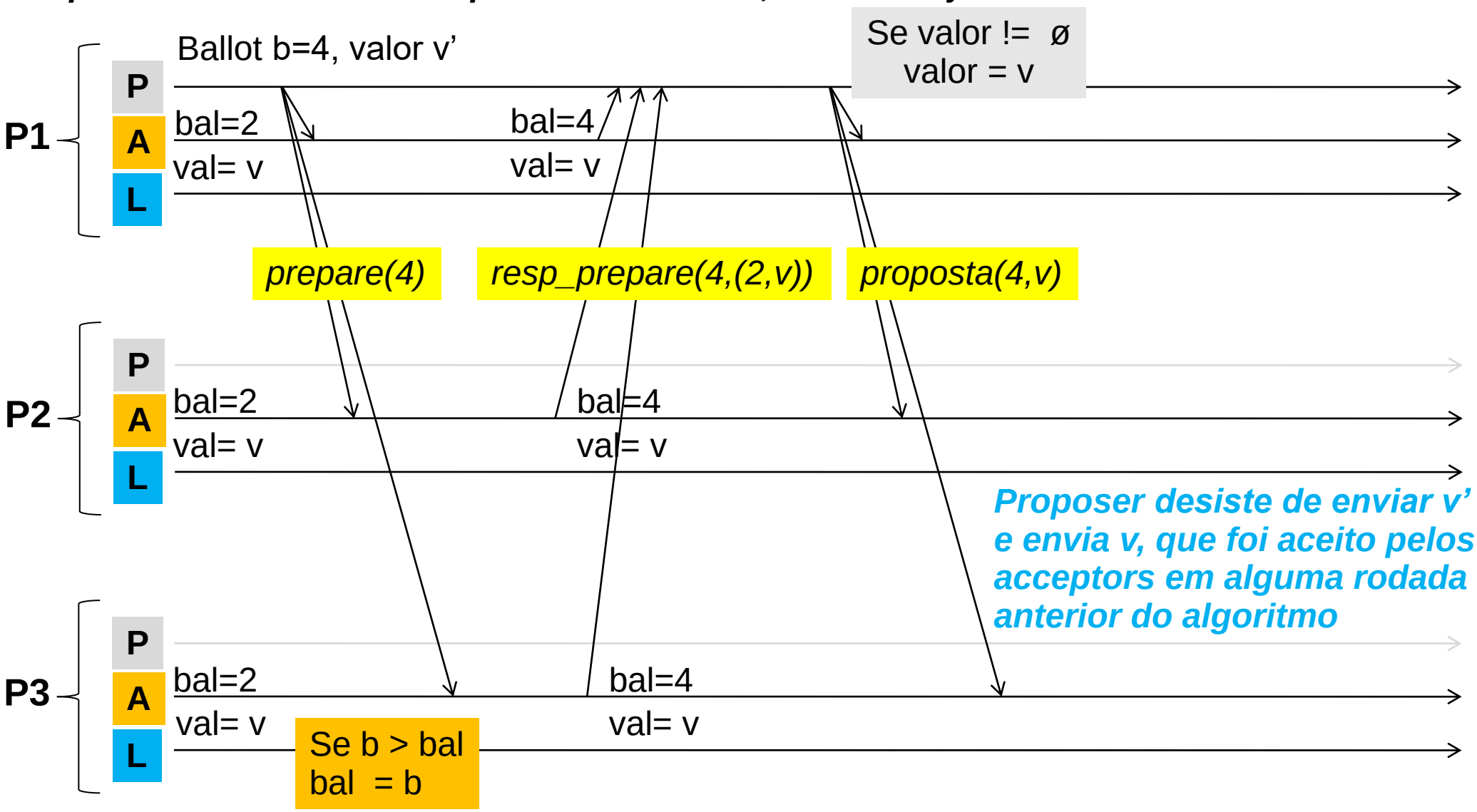
Paxos: Tratando propostas com valor já aceito

Proposer P1 envia ballot de próxima instância, mas valor já foi aceito em instância anterior



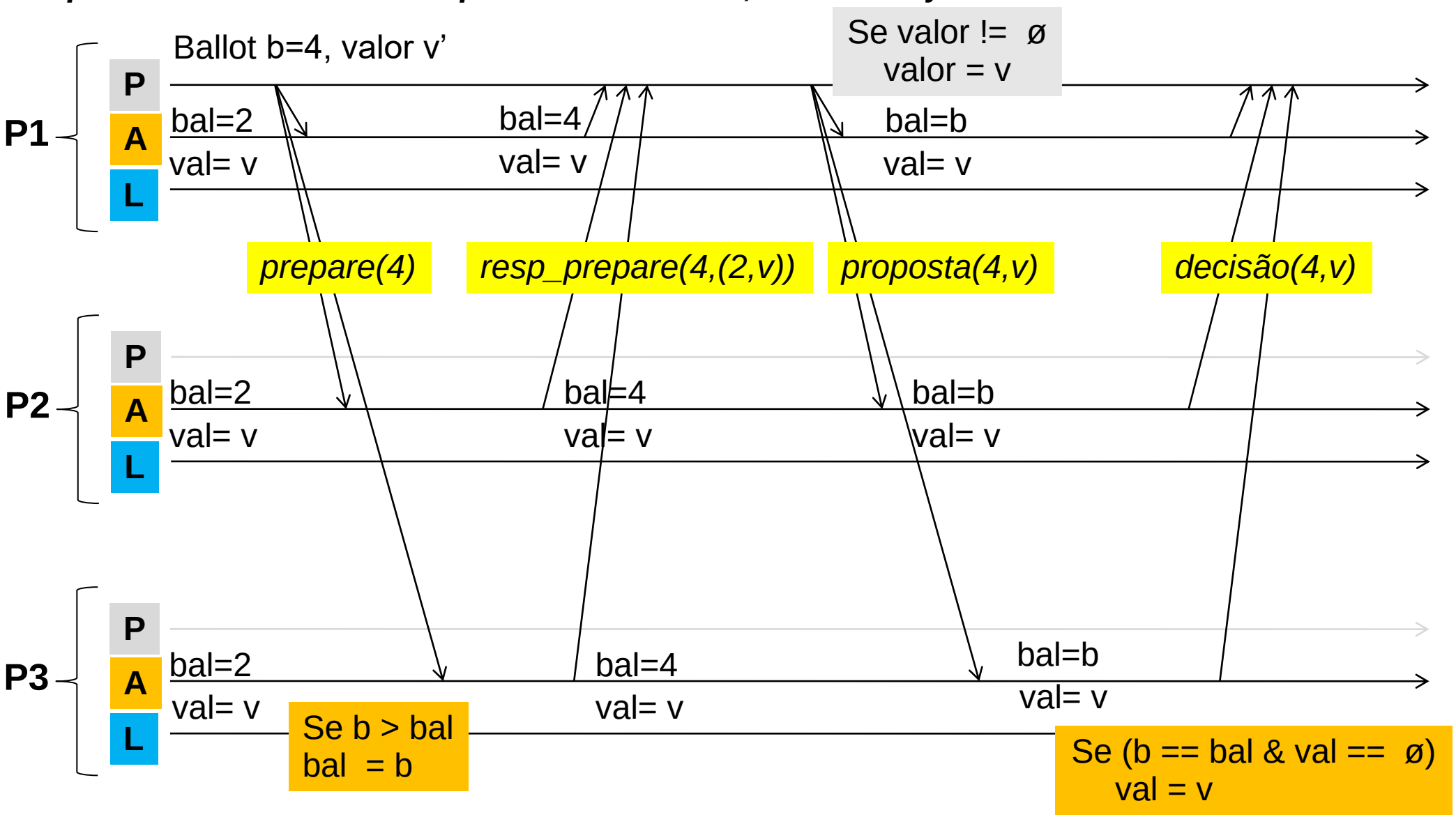
Paxos: Tratando propostas com valor já aceito

Proposer P1 envia ballot de próxima instância, mas valor já foi aceito em instância anterior

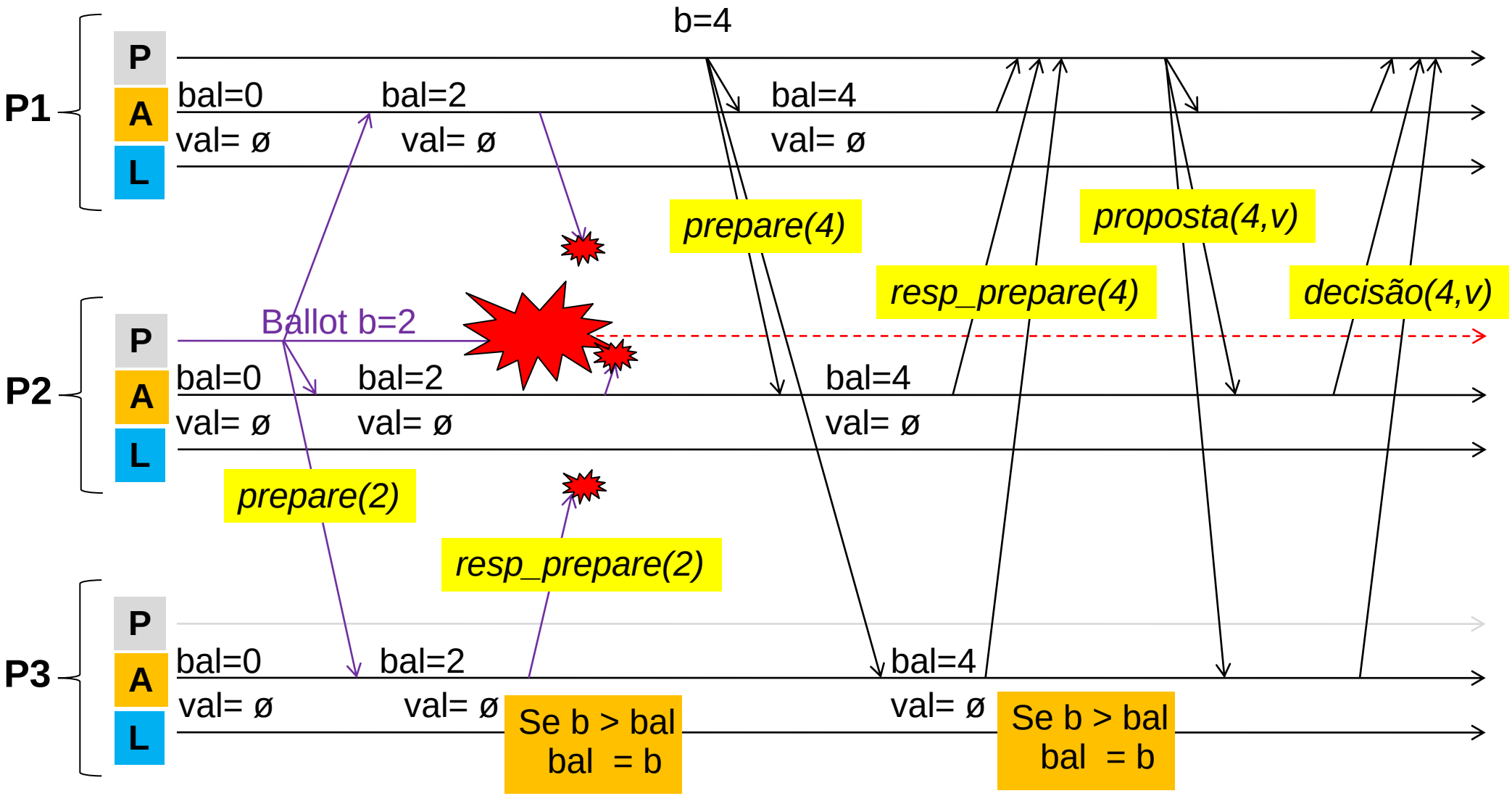


Paxos: Tratando propostas com valor já aceito

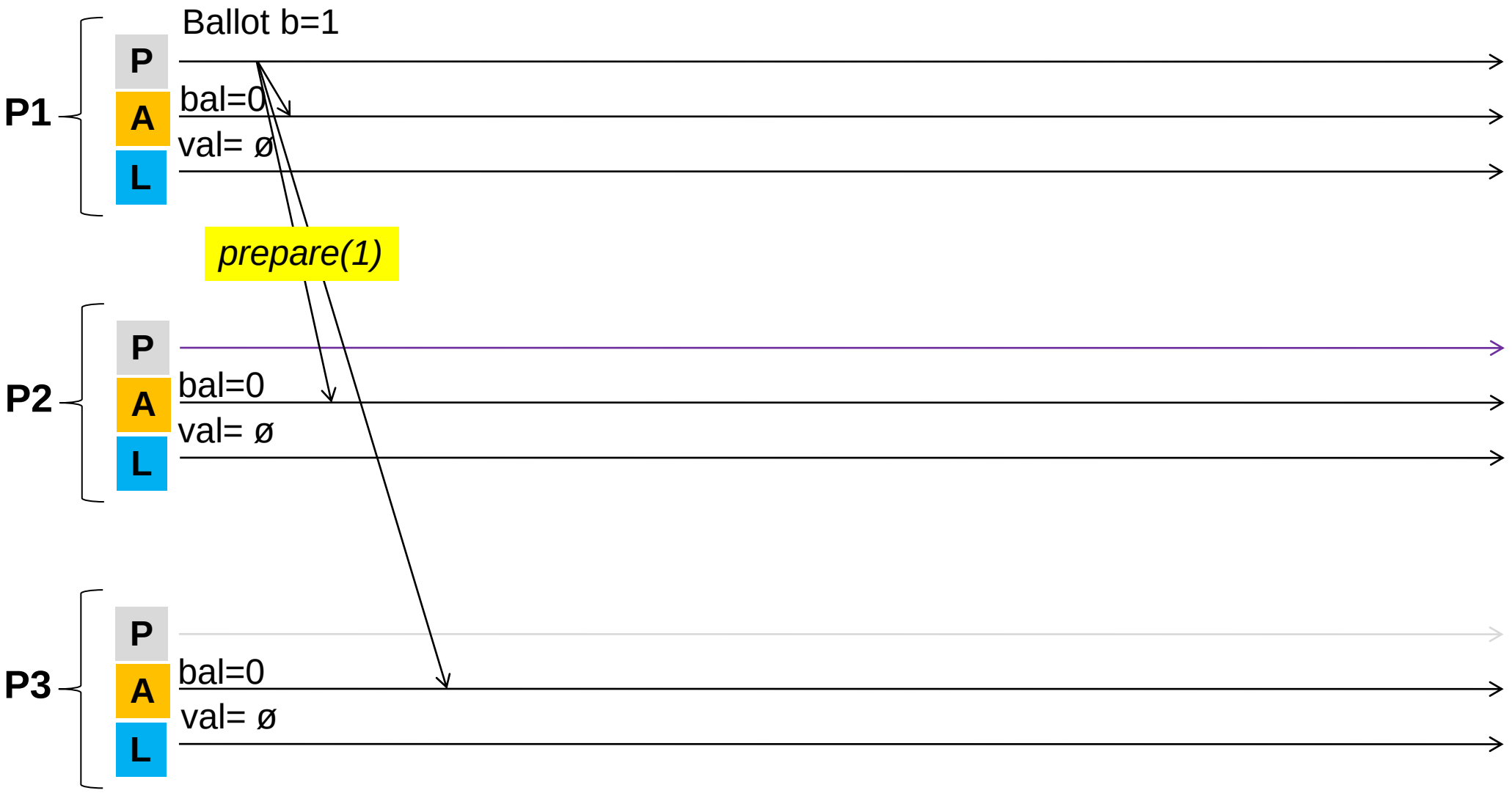
Proposer P1 envia ballot de próxima instância, mas valor já foi aceito em instância anterior



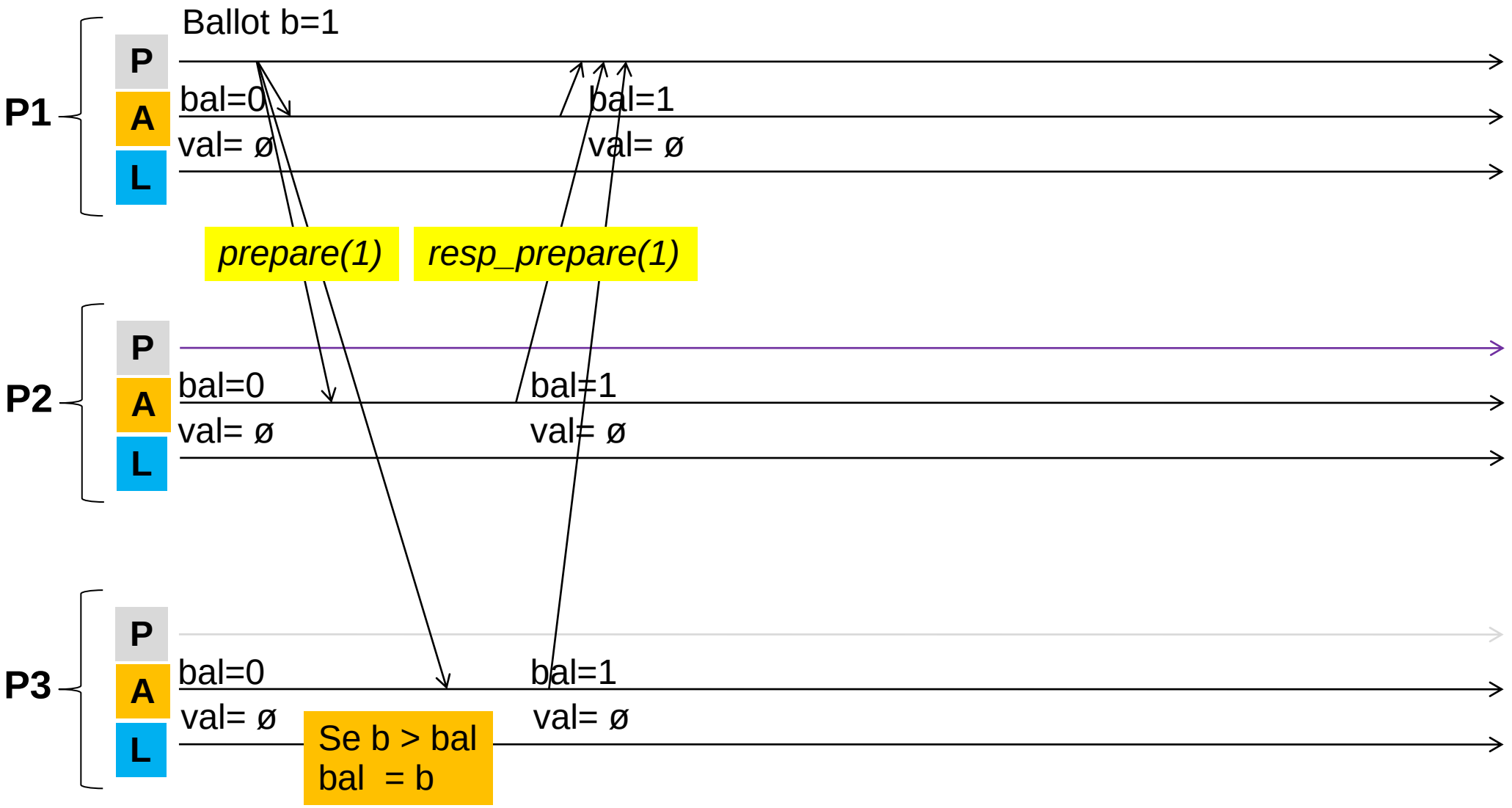
Paxos: proposta de um valor (com falha do proposer)



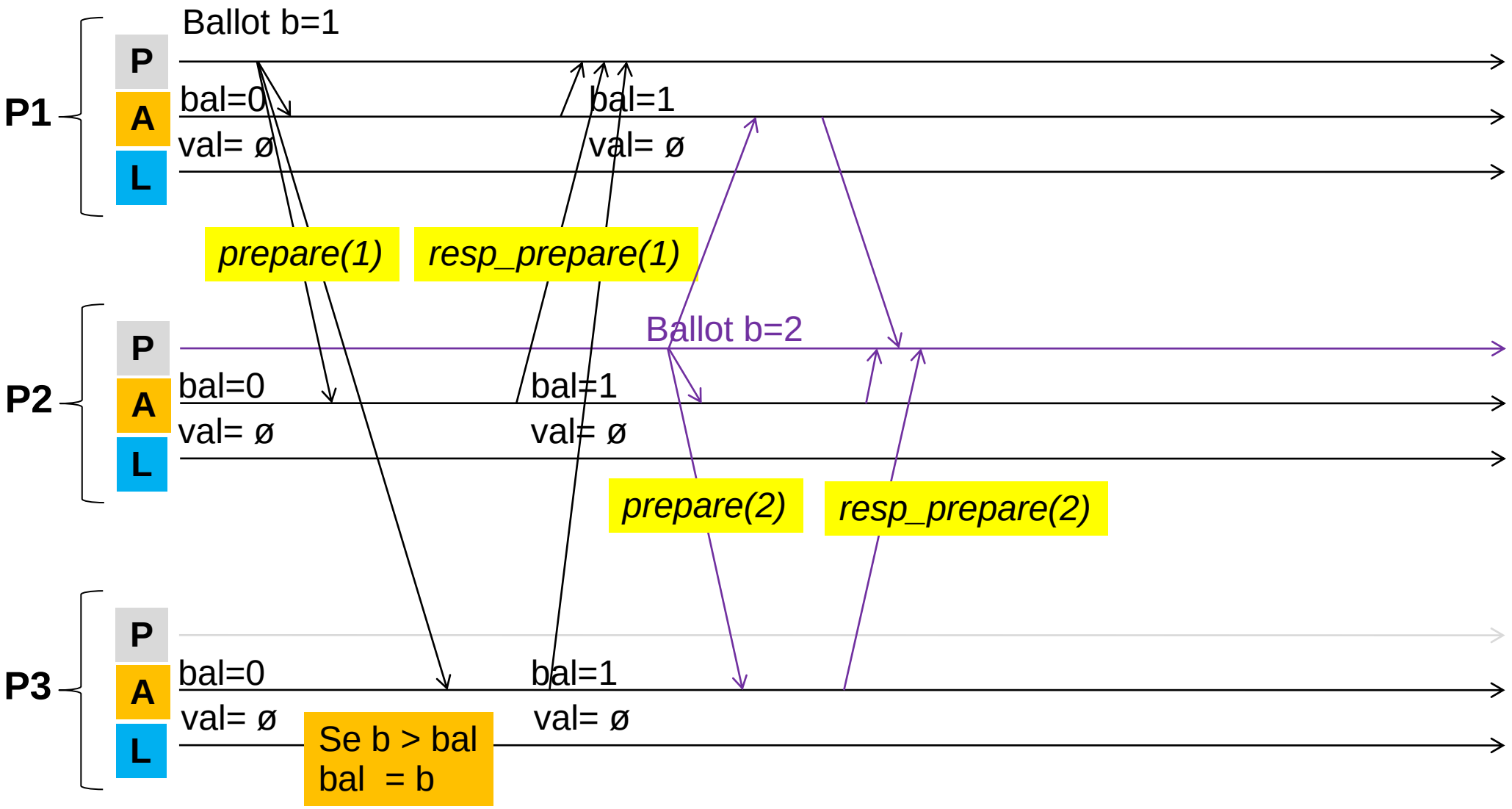
Paxos: problema da terminação



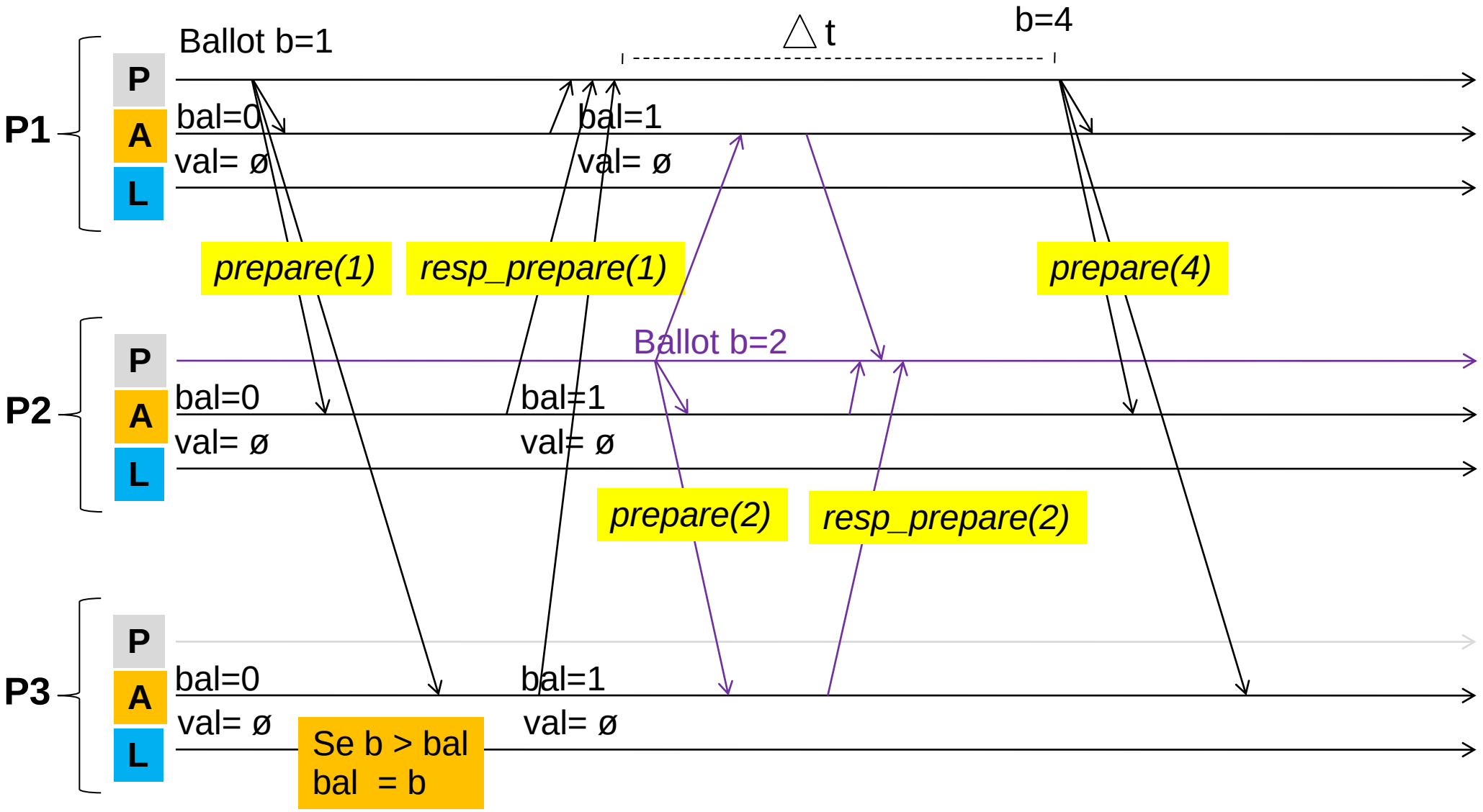
Paxos: problema da terminação



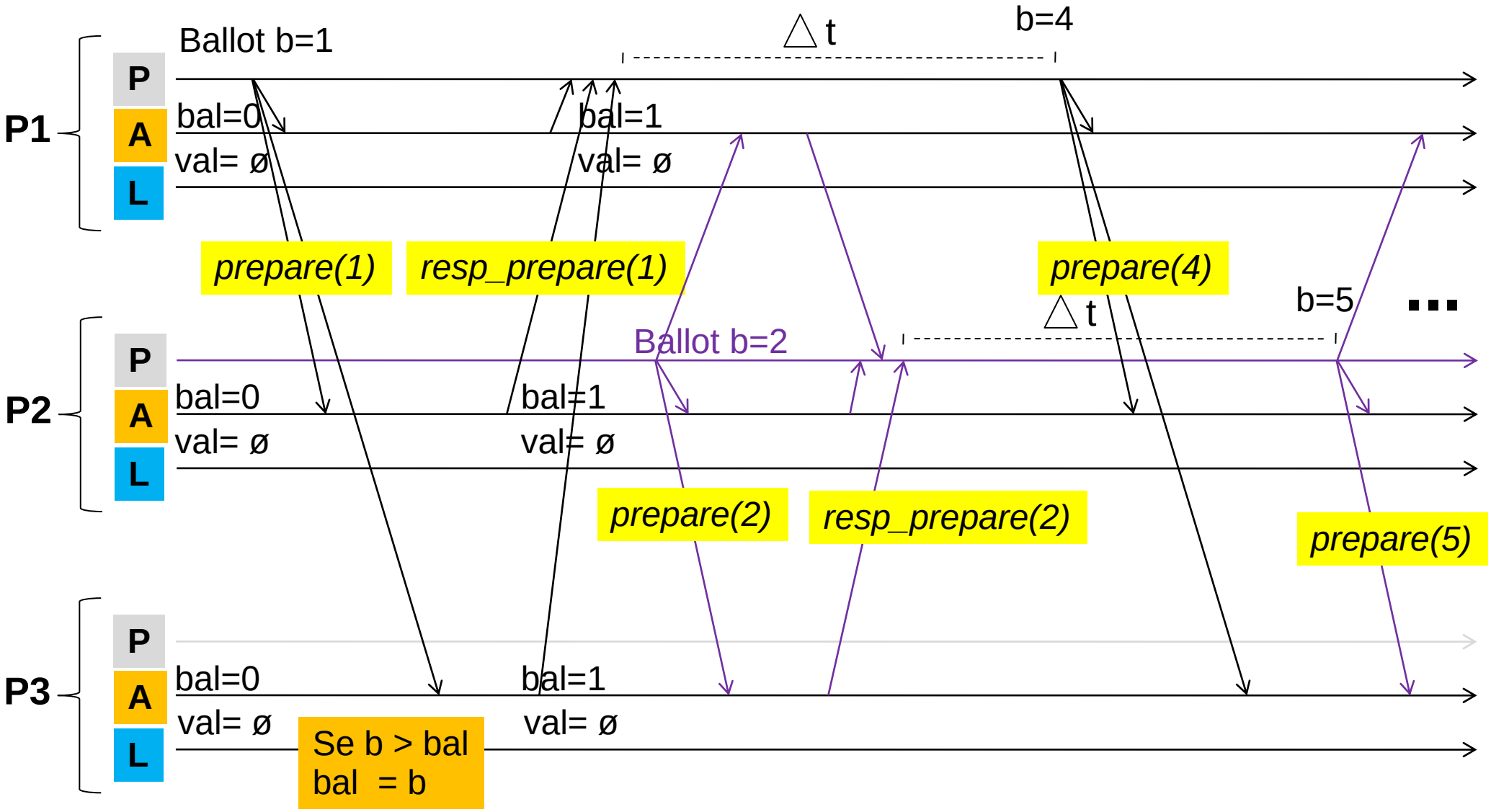
Paxos: problema da terminação



Paxos: problema da terminação



Paxos: problema da terminação



Paxos: problema da terminação

Então o consenso pode nunca ser alcançado?

Não em sistemas assíncronos.

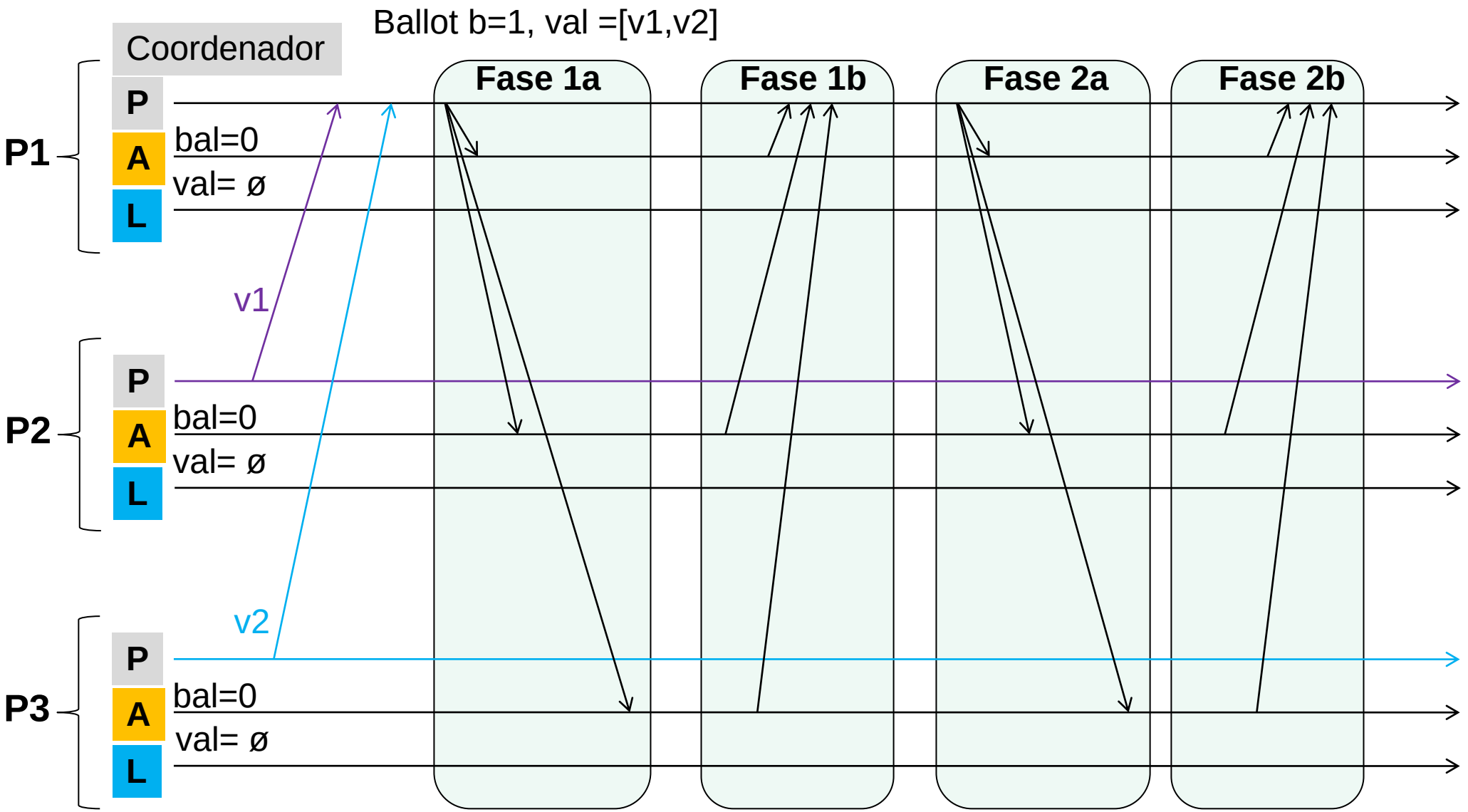
Em sistemas parcialmente síncronos, eventualmente as fases 1 e 2 ocorrerão em um intervalo no qual o sistema comporta-se de forma síncrona

Paxos pode comprometer *liveness* enquanto o sistema comparta-se de modo assíncrono, mas nunca compromete *safety*

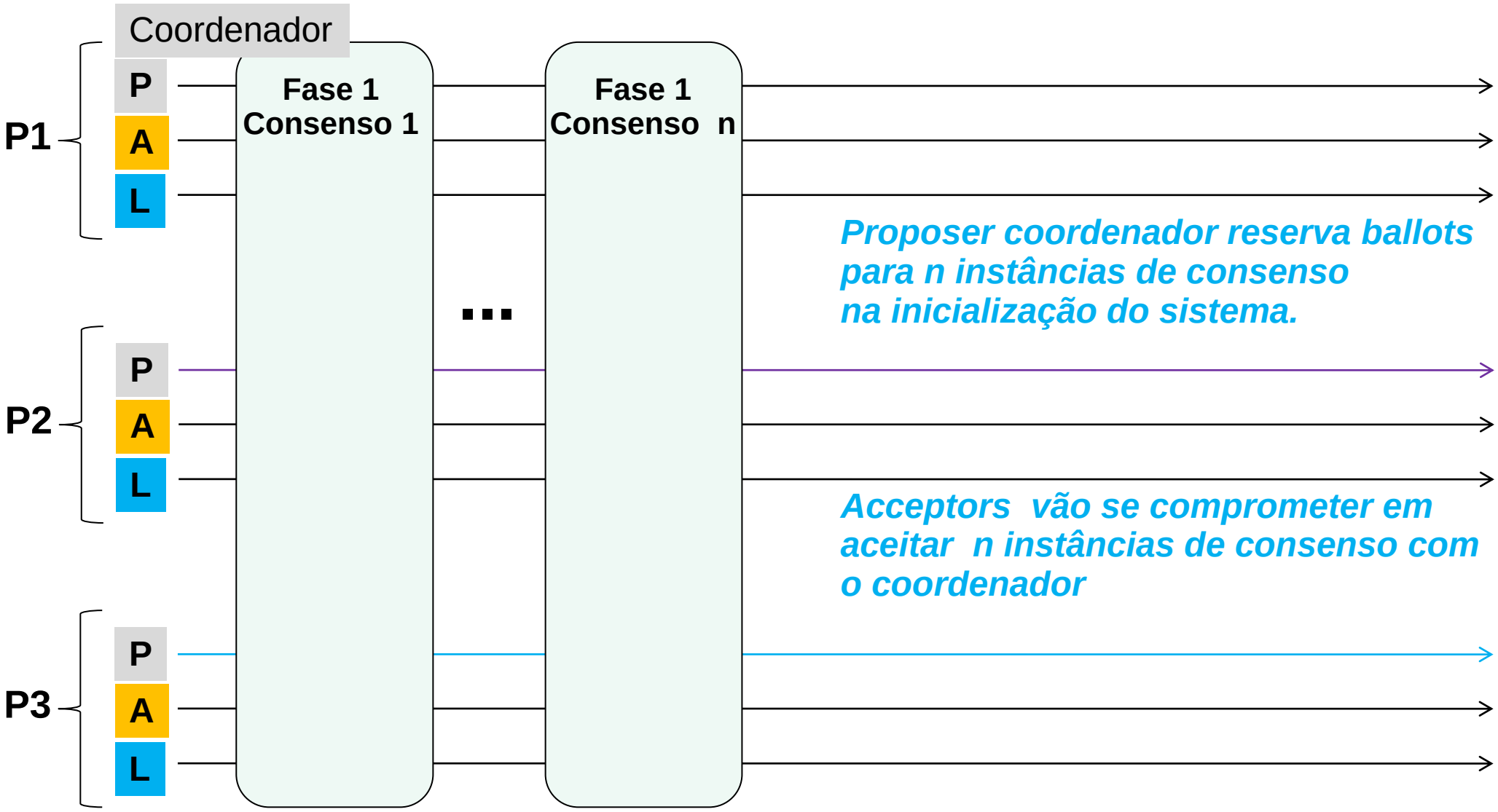
Paxos: problema da terminação

- Para minimizar as colisões nas propostas por valores
  - Uso de ***proposer* coordenador**
  - O coordenador é o único que faz propostas
  - Caso o coordenador falhe, um novo coordenador é eleito

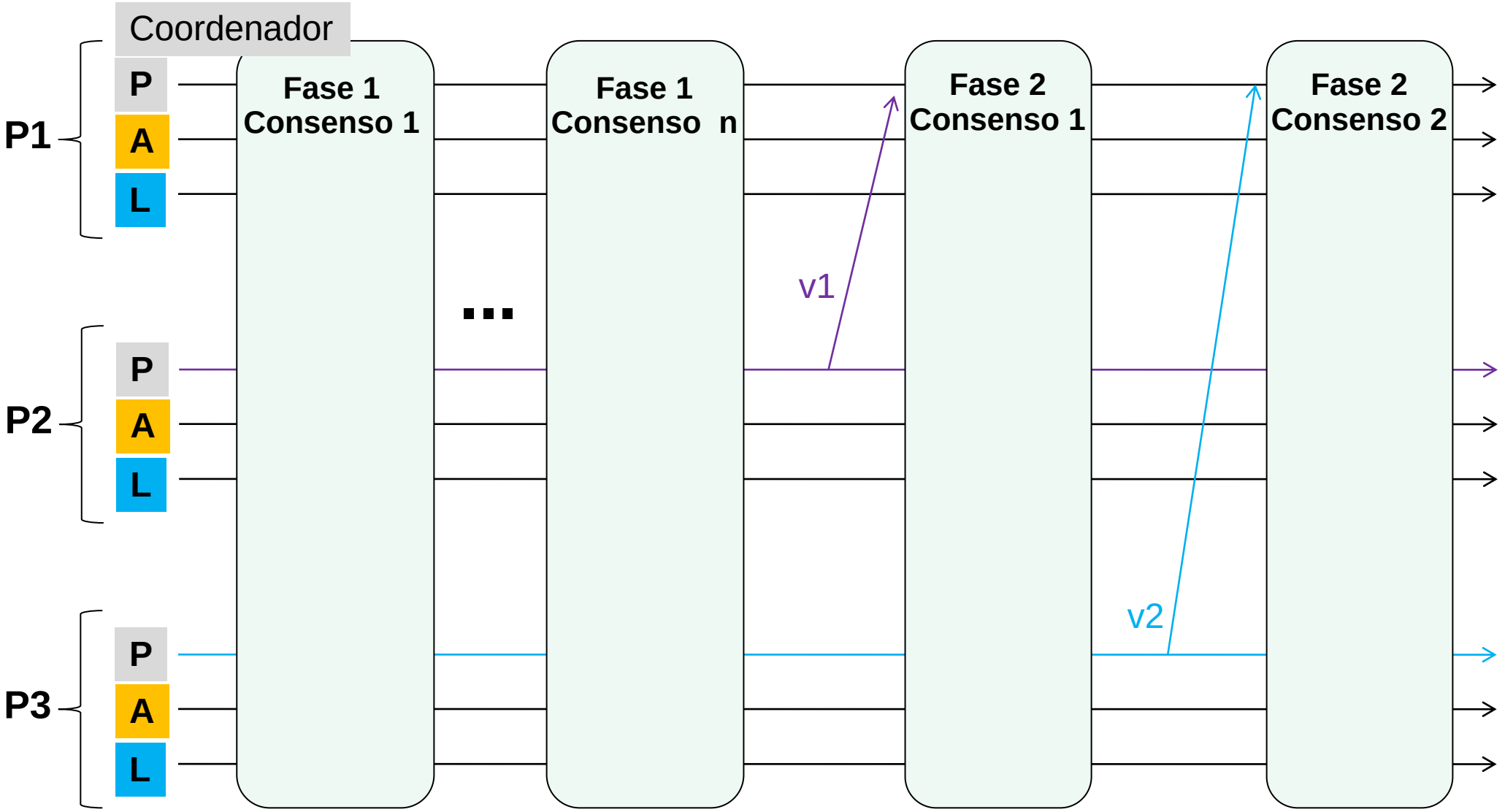
Paxos: *Proposer* coordenador



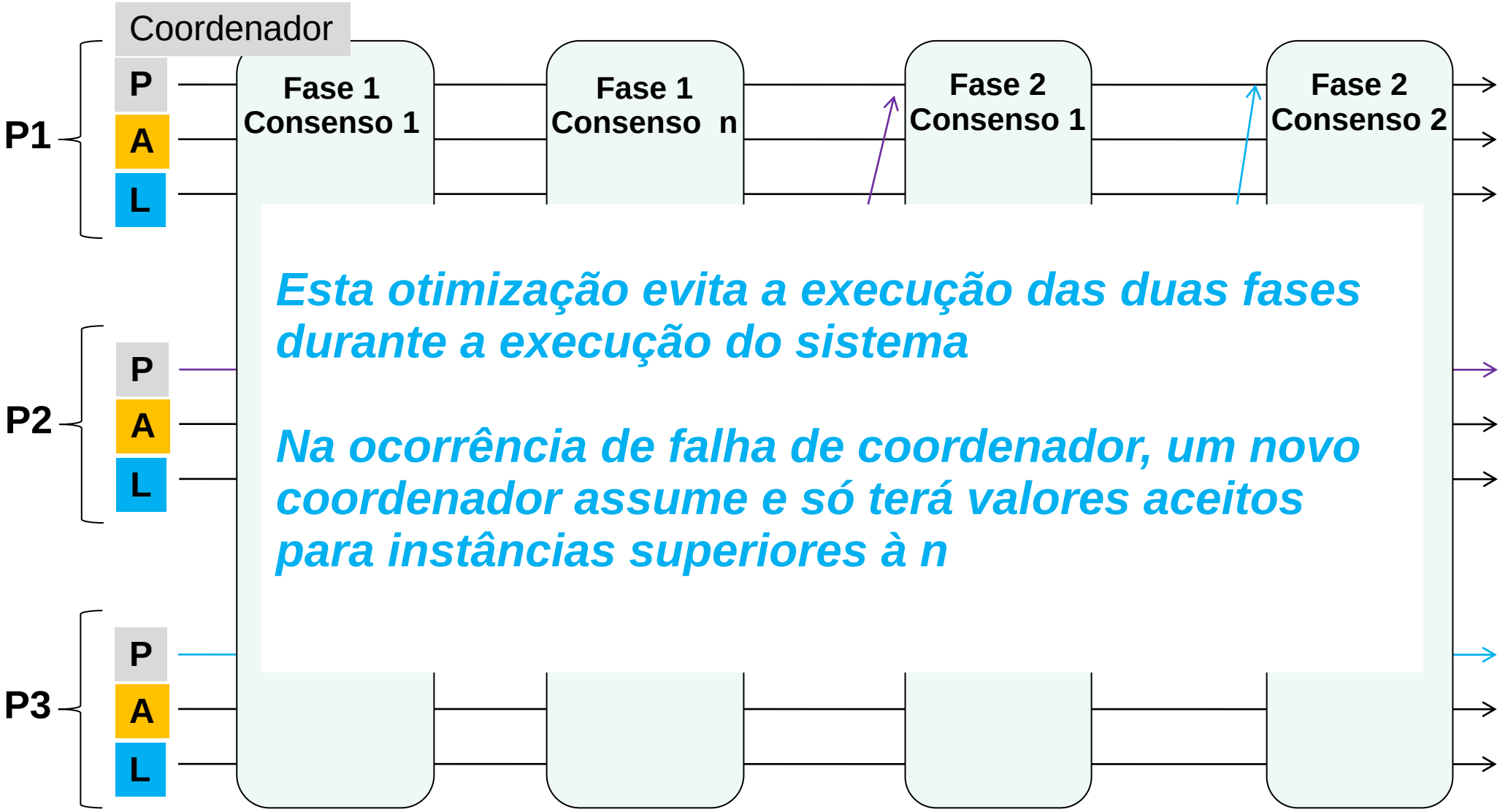
Paxos: Otimização com reserva de *ballots*



Paxos: Otimização com reserva de *ballots*



Paxos: Otimização com reserva de *ballots*

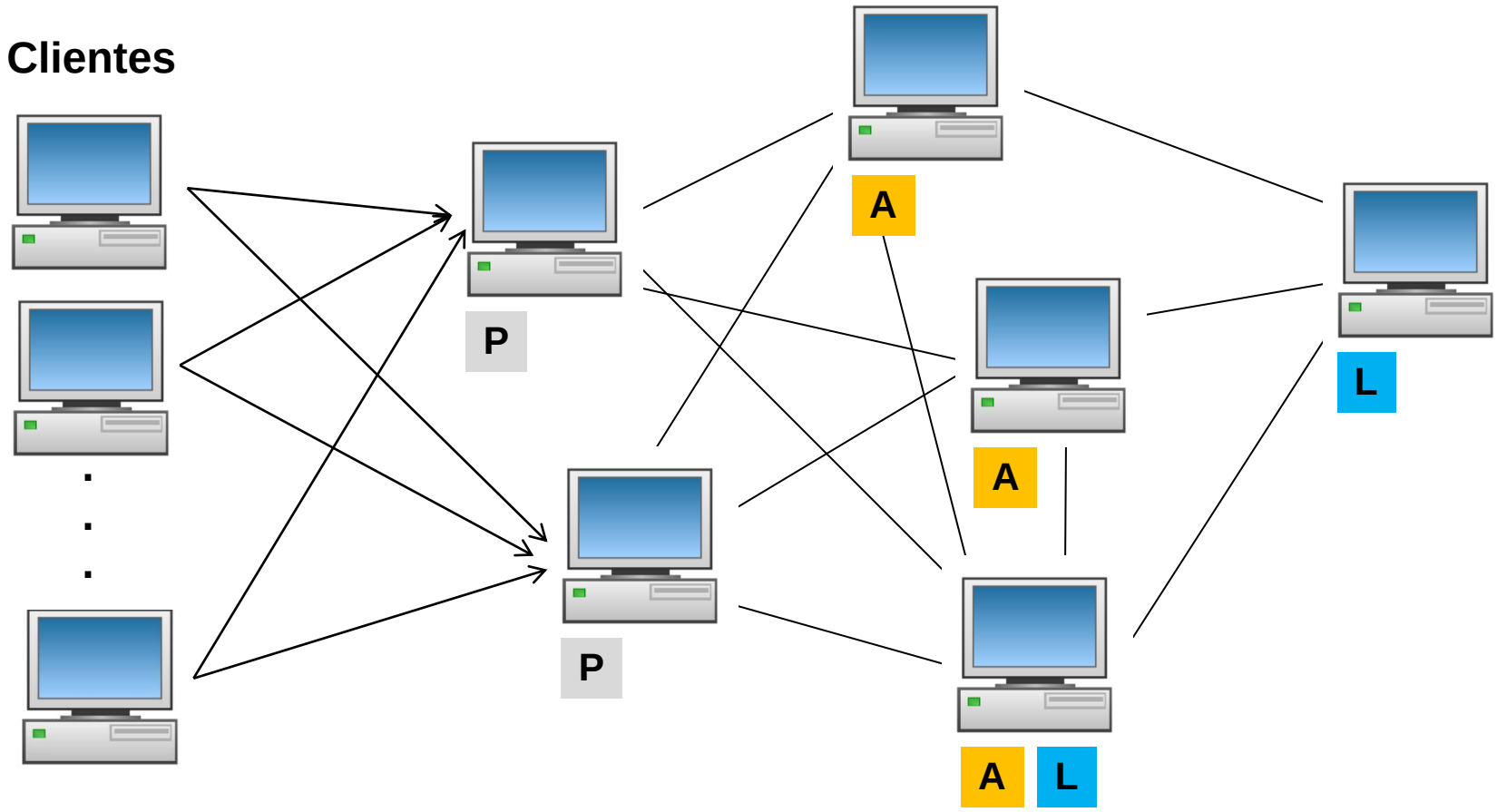


Paxos: Algumas considerações

Embora os exemplos mostrem cada processo executando os papéis de *proposer*, *acceptor* e *learner*, outras configurações podem ser adotadas

Porém, deve-se cuidar para as premissas sobre o número de falhas toleradas esteja de acordo com a configuração escolhida

Paxos: Algumas considerações



Para manter o serviço disponível com esta configuração:

Proposers *(np = f+1)*: no máximo 1 falha

Acceptors *(na = 2f+1)*: no máximo 1 falha

Learners *(nl = f+1)*: no máximo 1 falha

Paxos: Algumas considerações

Paxos ainda é o protocolo de consenso distribuído mais utilizado, dando origem para inúmeras variações e otimizações

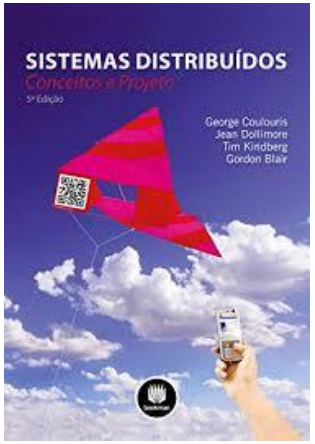
- Fast Paxos
- Disk Paxos
- Cheao Paxos
- S-Paxos
- Ring Paxos, Multi-Ring Paxos
- Generalized Paxos
- Egalitarian Paxos
- P4xos (Net Paxos)
- Kernel Paxos
- Byzantine Paxos
- ...

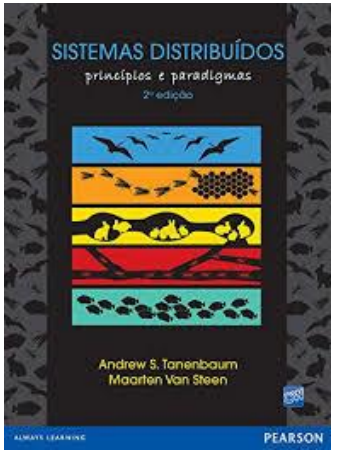
Outros protocolos surgem como alternativa ao Paxos, como o Raft e Zab

Referências

Parte destes slides são baseadas em material de aula dos livros:

- *Coulouris, George; Dollimore, Jean; Kindberg, Tim; Blair, Gordon. Sistemas Distribuídos: Conceitos e Projetos. Bookman; 5ª edição. 2013. ISBN: 8582600534*
- *Tanenbaum, Andrew S.; Van Steen, Maarten. Sistemas Distribuídos: Princípios e Paradigmas. 2007. Pearson Universidades; 2ª edição. ISBN: 8576051427*





- *Imagens e clip arts diversos:*
  https://free-icon-rainbow.com/
  https://www.gratispng.com/