https://youtu.be/UUQ8xYWR4do

Paxos in Pictures

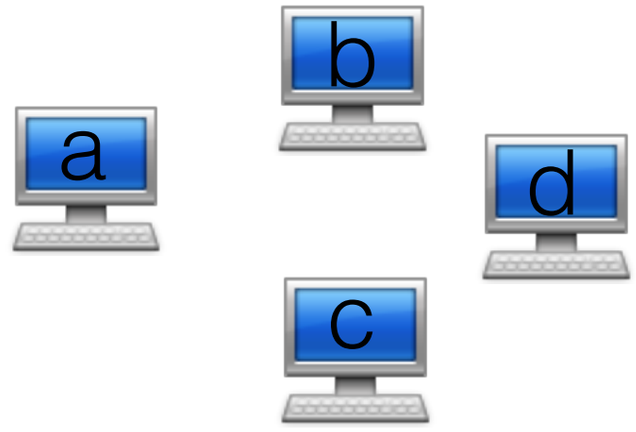
Quick tutorial of Lamport's Distributed Consensus Algorithm

Justin Pearson
May 25, 2017

Paxos: Problem Statement

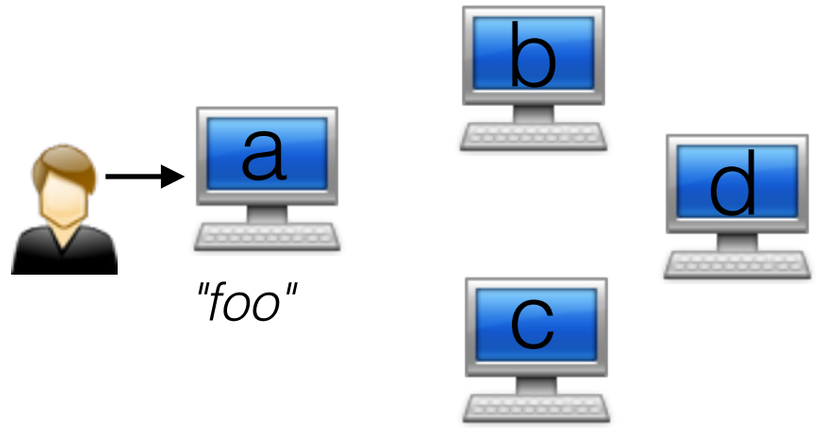
- Network of computers
- Client(s) gives computer(s) a value(s)
- Get them to agree on **some** value

Paxos: Problem Statement



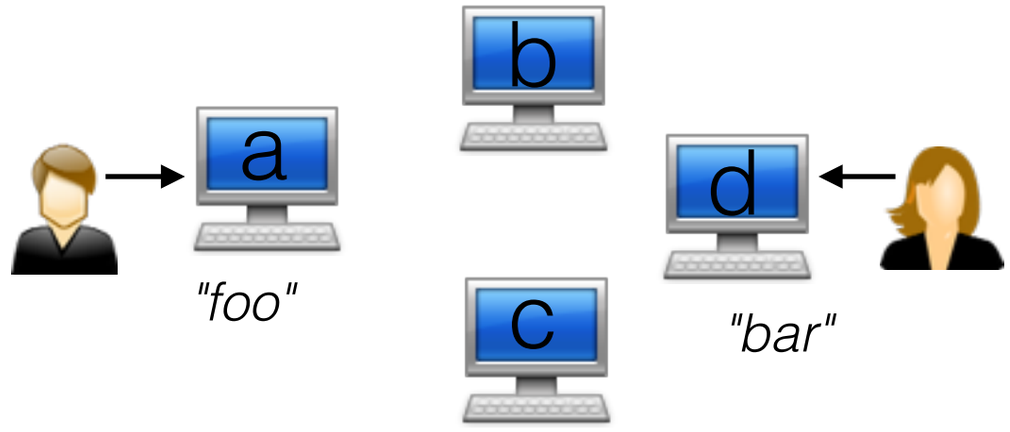
- Network of computers
- Client(s) gives computer(s) a value(s)
- Get them to agree on **some** value

Paxos: Problem Statement



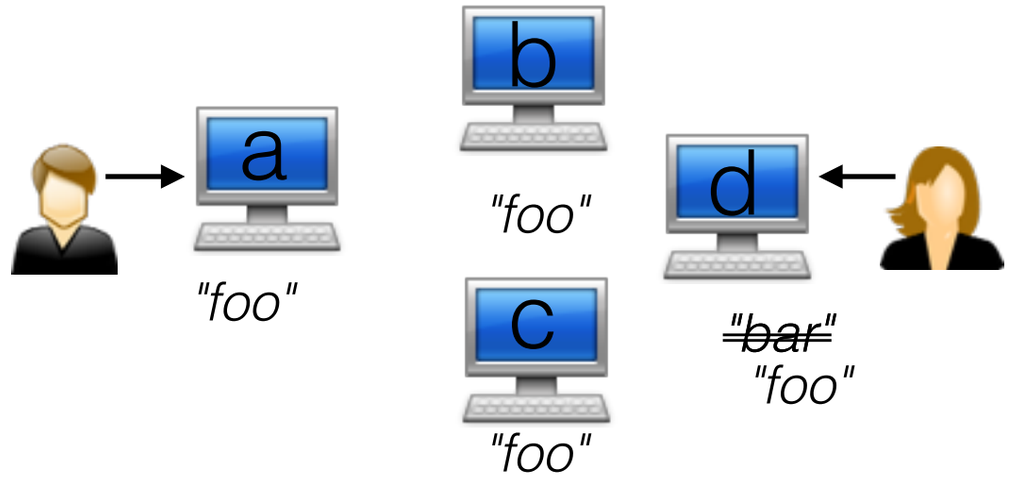
- Network of computers
- Client(s) gives computer(s) a value(s)
- Get them to agree on **some** value

Paxos: Problem Statement



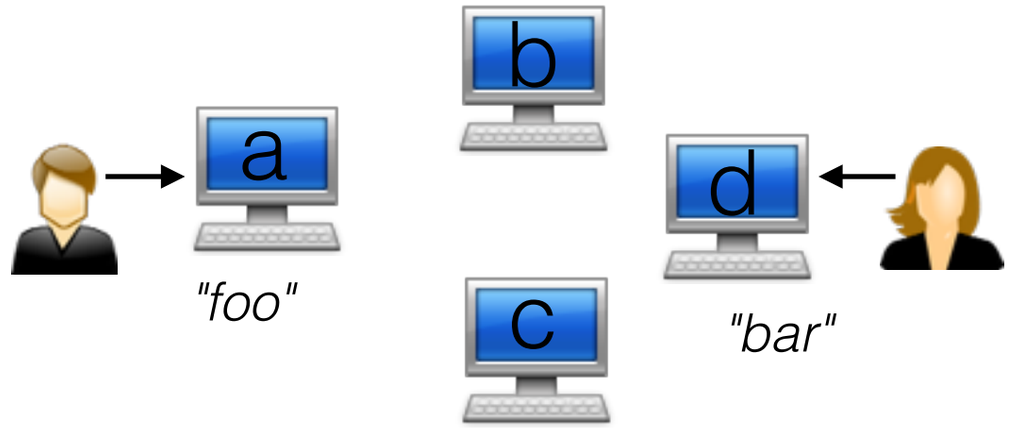
- Network of computers
- Client(s) gives computer(s) a value(s)
- Get them to agree on **some** value

Paxos: Problem Statement



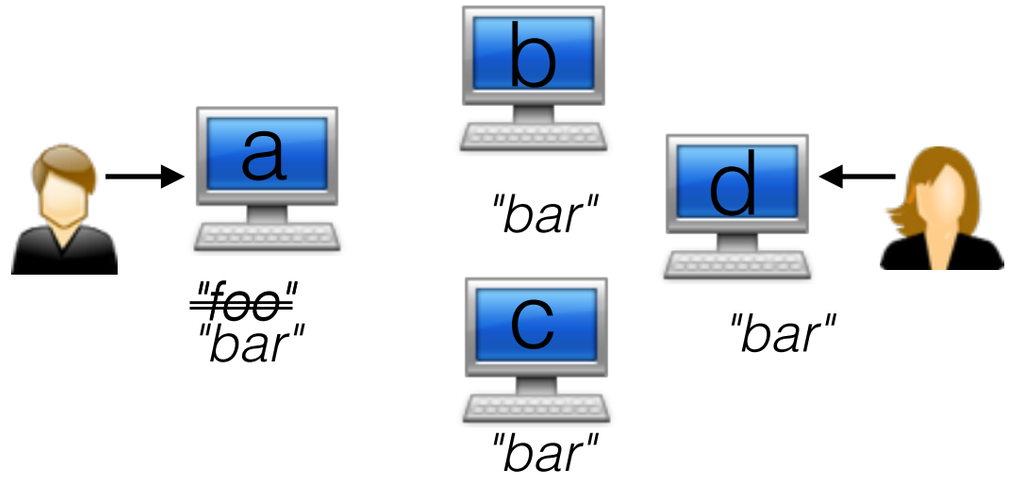
- Network of computers
- Client(s) gives computer(s) a value(s)
- Get them to agree on **some** value

Paxos: Problem Statement



- Network of computers
- Client(s) gives computer(s) a value(s)
- Get them to agree on **some** value

Paxos: Problem Statement



- Network of computers
- Client(s) gives computer(s) a value(s)
- Get them to agree on **some** value

Resources

Lamport, Leslie (2001). Paxos Made Simple ACM SIGACT News (Distributed Computing Column) 32, 4 (Whole Number 121, December 2001) 51-58.

http://research.microsoft.com/en-us/um/people/lamport/pubs/pubs.html#paxos-simple

https://en.wikipedia.org/wiki/Paxos_(computer_science)

"Paxos Made Simple"

A proposer issues a proposal by sending, to some set of acceptors, a request that the proposal be accepted. (This need not be the same set of acceptors that responded to the initial requests.) Let's call this an *accept* request.

This describes a proposer's algorithm. What about an acceptor? It can receive two kinds of requests from proposers: *prepare* requests and *accept* requests. An acceptor can ignore any request without compromising safety. So, we need to say only when it is allowed to respond to a request. It can always respond to a *prepare* request. It can respond to an *accept* request, accepting the proposal, iff it has not promised not to. In other words:

P1^a. An acceptor can accept a proposal numbered n iff it has not responded to a *prepare* request having a number greater than n.

Observe that P1^a subsumes P1.

We now have a complete algorithm for choosing a value that satisfies the required safety properties—assuming unique proposal numbers. The final algorithm is obtained by making one small optimization.

Suppose an acceptor receives a *prepare* request numbered n, but it has already responded to a *prepare* request numbered greater than n, thereby promising not to accept any new proposal numbered n. There is then no reason for the acceptor to respond to the new *prepare* request, since it will not accept the proposal numbered n that the proposer wants to issue. So we have the acceptor ignore such a *prepare* request. We also have it ignore a *prepare* request for a proposal it has already accepted.

With this optimization, an acceptor needs to remember only the highest-numbered proposal that it has ever accepted and the number of the highest-numbered *prepare* request to which it has responded. Because P2^c must be kept invariant regardless of failures, an acceptor must remember this information even if it fails and then restarts. Note that the proposer can always abandon a proposal and forget all about it—as long as it never tries to issue another proposal with the same number.

Putting the actions of the proposer and acceptor together, we see that the algorithm operates in the following two phases.

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.

A proposer can make multiple proposals, so long as it follows the algorithm for each one. It can abandon a proposal in the middle of the protocol at any time. (Correctness is maintained, even though requests and/or responses for the proposal may arrive at their destinations long after the proposal was abandoned.) It is probably a good idea to abandon a proposal if some proposer has begun trying to issue a higher-numbered one. Therefore, if an acceptor ignores a *prepare* or *accept* request because it has already received a *prepare* request with a higher number, then it should probably inform the proposer, who should then abandon its proposal. This is a performance optimization that does not affect correctness.

2.3 Learning a Chosen Value

To learn that a value has been chosen, a learner must find out that a proposal has been accepted by a majority of acceptors. The obvious algorithm is to have each acceptor, whenever it accepts a proposal, respond to all learners, sending them the proposal. This allows learners to find out about a chosen value as soon as possible, but it requires each acceptor to respond to each learner—a number of responses equal to the product of the number of acceptors and the number of learners.

The assumption of non-Byzantine failures makes it easy for one learner to find out from another learner that a value has been accepted. We can have the acceptors respond with their acceptances to a distinguished learner, which in turn informs the other learners when a value has been chosen. This approach requires an extra round for all the learners to discover the chosen value. It is also less reliable, since the distinguished learner could fail. But it requires a number of responses equal only to the sum of the number of acceptors and the number of learners.

More generally, the acceptors could respond with their acceptances to some set of distinguished learners, each of which can then inform all the learners when a value has been chosen. Using a larger set of distinguished

"Paxos Made Simple"

A proposer issues a proposal by sending, to some set of acceptors, a request that the proposal be accepted. (This need not be the same set of acceptors that responded to the initial requests.) Let's call this an *accept* request.

This describes a proposer's algorithm. What about an acceptor? It can receive two kinds of requests from proposers: *prepare* requests and *accept* requests. An acceptor can ignore any request without compromising safety. So, we need to say only when it is allowed to respond to a request. It can always respond to a *prepare* request. It can respond to an *accept* request, accepting the proposal, iff it has not promised not to. In other words:

P1^a. An acceptor can accept a proposal numbered n iff it has not responded to a *prepare* request having a number greater than n.

Observe that P1^a subsumes P1.

We now have a complete algorithm for choosing a value that satisfies the required safety properties—assuming unique proposal numbers. The final algorithm is obtained by making one small optimization.

Suppose an acceptor receives a *prepare* request numbered n, but it has already responded to a *prepare* request numbered greater than n, thereby promising not to accept any new proposal numbered n. There is then no reason for the acceptor to respond to the new *prepare* request, since it will not accept the proposal numbered n that the proposer wants to issue. So we have the acceptor ignore such a *prepare* request. We also have it ignore a *prepare* request for a proposal it has already accepted.

With this optimization, an acceptor needs to remember only the highest-numbered proposal that it has ever accepted and the number of the highest-numbered *prepare* request to which it has responded. Because P2^c must be kept invariant regardless of failures, an acceptor must remember this information even if it fails and then restarts. Note that the proposer can always abandon a proposal and forget all about it—as long as it never tries to issue another proposal with the same number.

Putting the actions of the proposer and acceptor together, we see that the algorithm operates in the following two phases.

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.

A proposer can make multiple proposals, so long as it follows the algorithm for each one. It can abandon a proposal in the middle of the protocol at any time. (Correctness is maintained, even though requests and/or responses for the proposal may arrive at their destinations long after the proposal was abandoned.) It is probably a good idea to abandon a proposal if some proposer has begun trying to issue a higher-numbered one. Therefore, if an acceptor ignores a *prepare* or *accept* request because it has already received a *prepare* request with a higher number, then it should probably inform the proposer, who should then abandon its proposal. This is a performance optimization that does not affect correctness.

2.3 Learning a Chosen Value

To learn that a value has been chosen, a learner must find out that a proposal has been accepted by a majority of acceptors. The obvious algorithm is to have each acceptor, whenever it accepts a proposal, respond to all learners, sending them the proposal. This allows learners to find out about a chosen value as soon as possible, but it requires each acceptor to respond to each learner—a number of responses equal to the product of the number of acceptors and the number of learners.

The assumption of non-Byzantine failures makes it easy for one learner to find out from another learner that a value has been accepted. We can have the acceptors respond with their acceptances to a distinguished learner, which in turn informs the other learners when a value has been chosen. This approach requires an extra round for all the learners to discover the chosen value. It is also less reliable, since the distinguished learner could fail. But it requires a number of responses equal only to the sum of the number of acceptors and the number of learners.

More generally, the acceptors could respond with their acceptances to some set of distinguished learners, each of which can then inform all the learners when a value has been chosen. Using a larger set of distinguished

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.

Phase 3.

To learn that a value has been chosen, a learner must find out that a proposal has been accepted by a majority of acceptors. The obvious algorithm is to have each acceptor, whenever it accepts a proposal, respond to all learners, sending them the proposal.

Outline

The Algorithm

- Example: how it works **initially**
- Example: how it handles **conflicts**
- Example: how it works **after consensus**

Proposer | Acceptor | Learner

"Want consensus! How about $v_{default}$?"

PrepareRequest[n]

"proposal number" → Proposal[n,v] ← "proposal value"

If n >= max rx'd PrepReq,

"New prepreq! Here is my highest Proposal's value."

ResponseToPrepareRequest[Proposal[m,w] or None]

If rx'd a majority,
v = max(rx'd Proposals).v or $v_{default}$

AcceptRequest[Proposal[n,v]]

"This Proposer gained majority support from As."

Decision[Proposal[n,v]]

If rx'd a majority,
value = v

Outline

- The Algorithm

- Example: how it works **initially**
- Example: how it handles **conflicts**
- Example: how it works **after consensus**

Alice

Bob

Carl

Alice P A L

Bob P A L

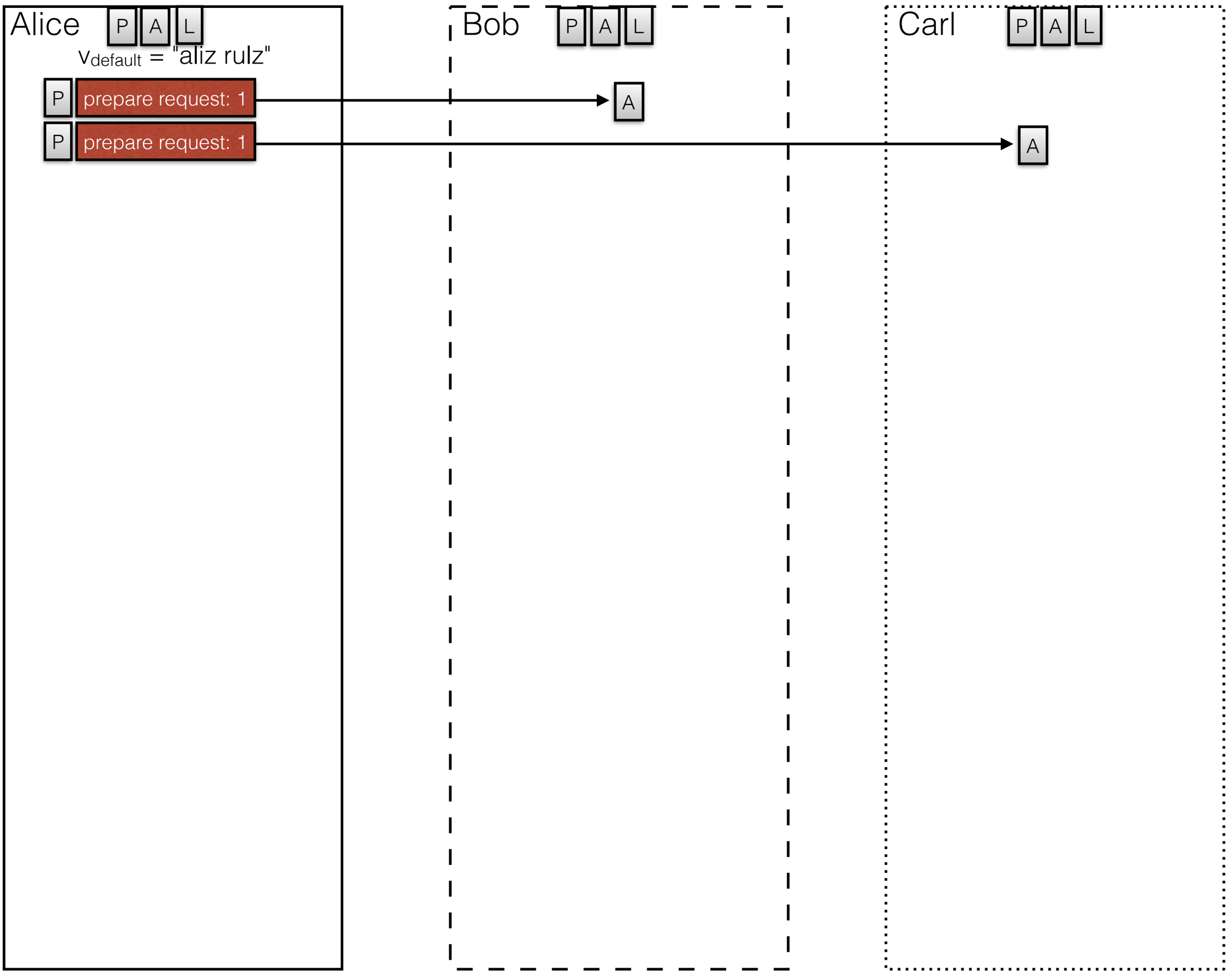
Carl P A L

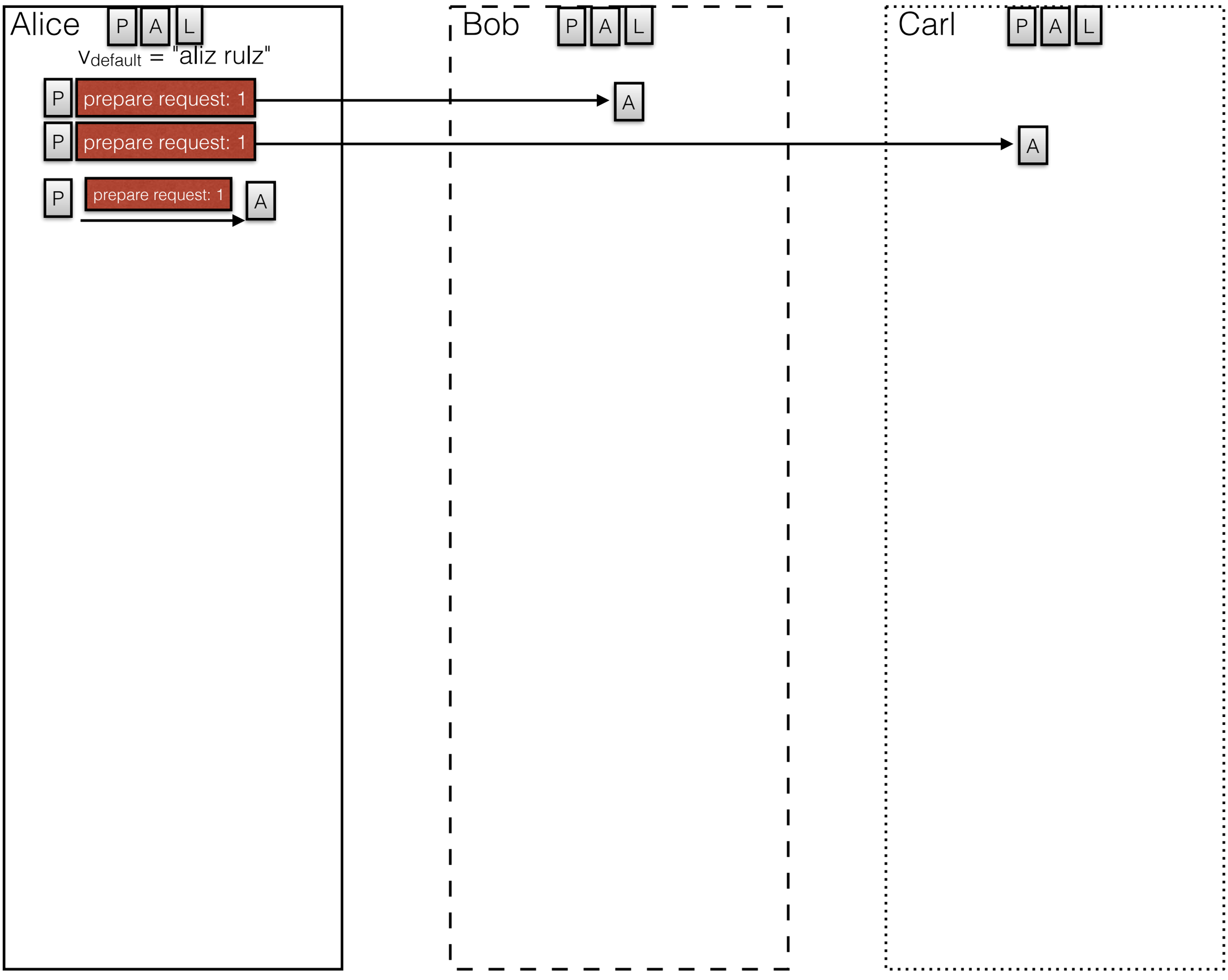
Alice P A L

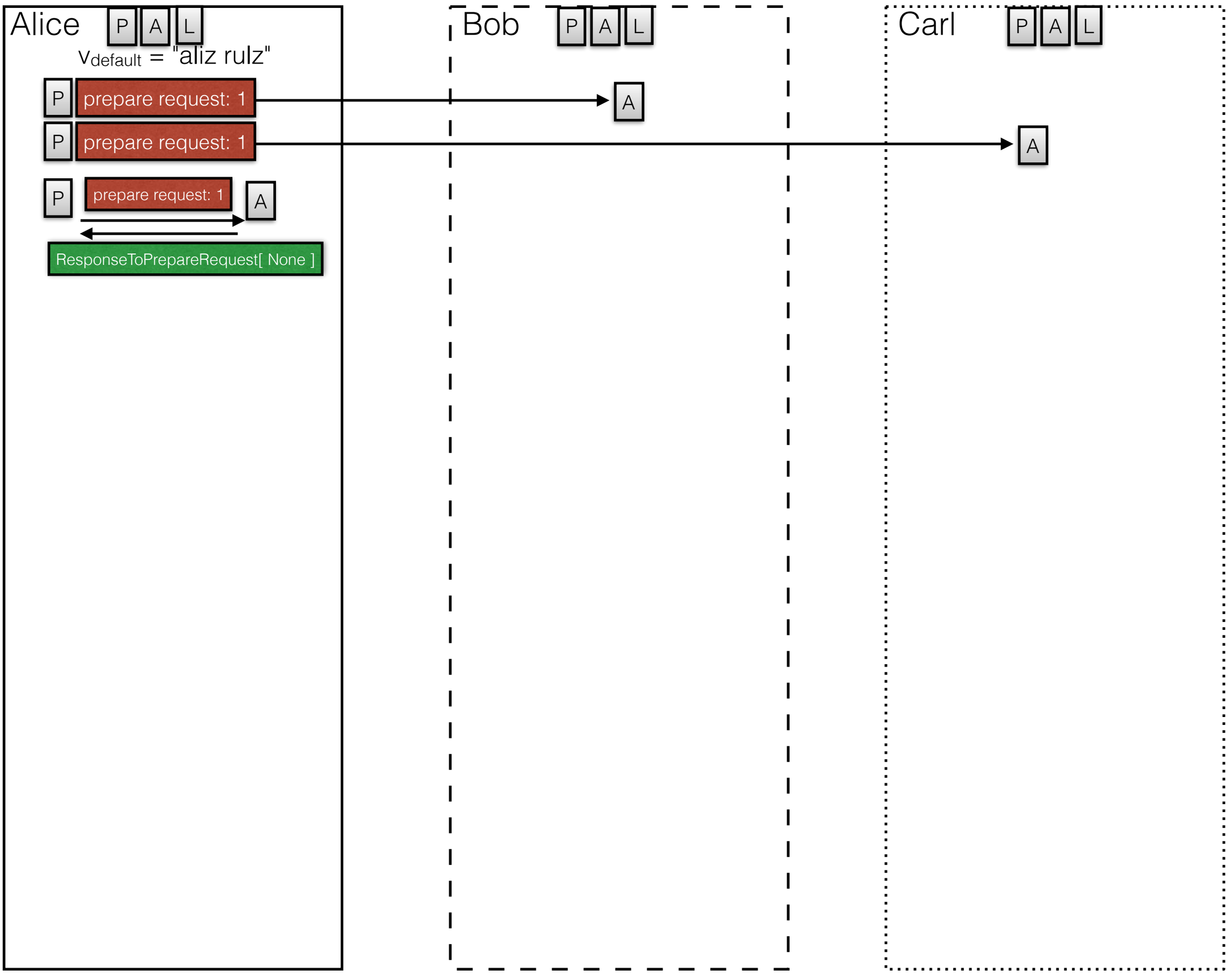
$v_{default}$ = "aliz rulz"

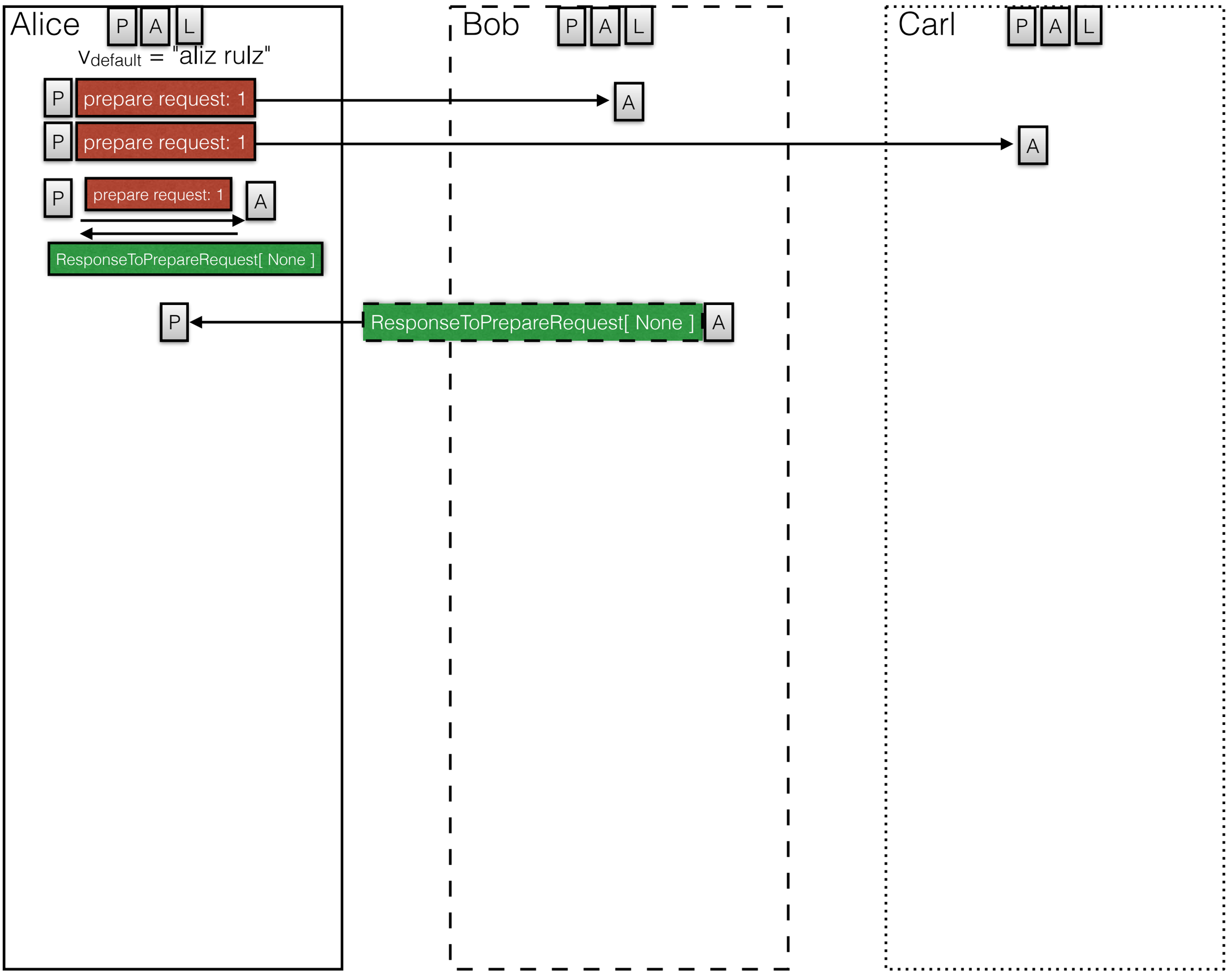
Bob P A L

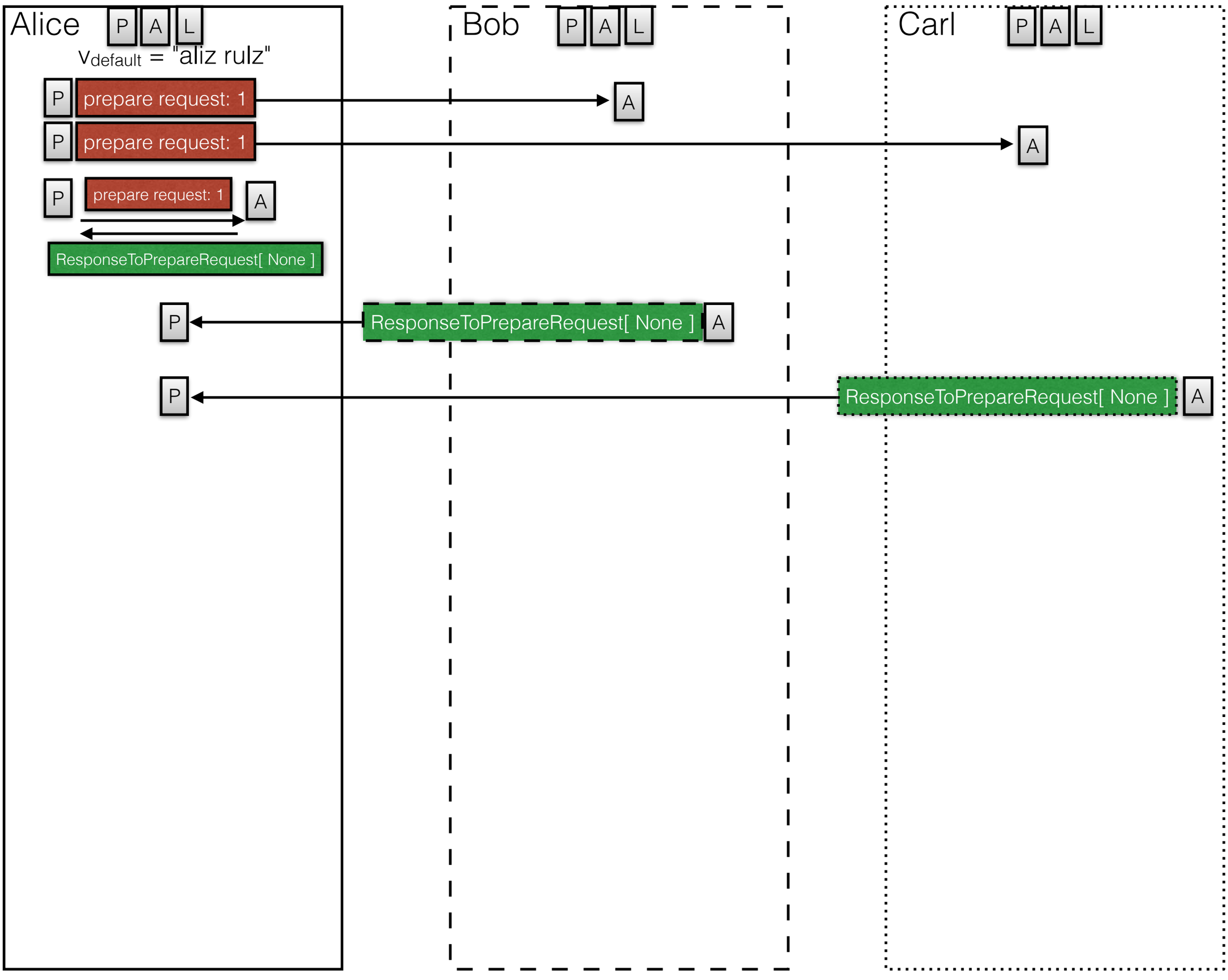
Carl P A L

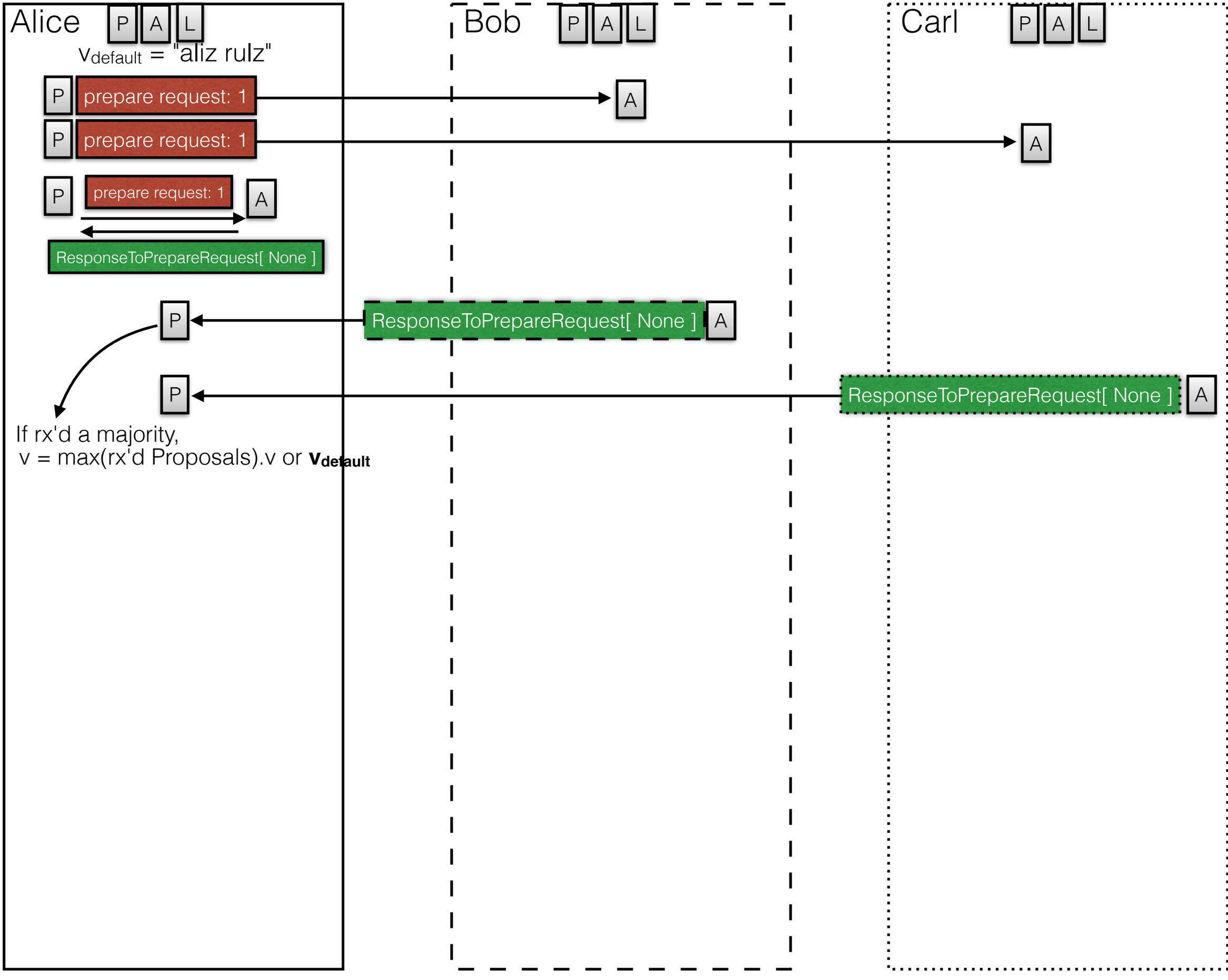

Alice
P
A
L
vdefault = "aliz rulz"
P
prepare request: 1
P
prepare request: 1
Bob
P
A
L
A
Carl
P
A
L
A


Alice
P
A
L
vdefault = "aliz rulz"
P
prepare request: 1
P
prepare request: 1
P
prepare request: 1
A
Bob
P
A
L
A
Carl
P
A
L
A


Alice
P
A
L
vdefault = "aliz rulz"
P
prepare request: 1
P
prepare request: 1
P
prepare request: 1
A
ResponseToPrepareRequest[None]
Bob
P
A
L
A
Carl
P
A
L
A


Alice
P
A
L
vdefault = "aliz rulz"
Bob
P
A
L
Carl
P
A
L
P
prepare request: 1
A
P
prepare request: 1
A
P
prepare request: 1
A
ResponseToPrepareRequest[None]
P
ResponseToPrepareRequest[None]
A


Alice
P
A
L
vdefault = "aliz rulz"
Bob
P
A
L
Carl
P
A
L
P
prepare request: 1
A
P
prepare request: 1
A
P
prepare request: 1
A
ResponseToPrepareRequest[None]
P
ResponseToPrepareRequest[None]
A
P
ResponseToPrepareRequest[None]
A


Alice
P
A
L
Bob
P
A
L
Carl
P
A
L
vdefault = "aliz rulz"
P
prepare request: 1
A
P
prepare request: 1
A
P
prepare request: 1
A
ResponseToPrepareRequest[None]
P
ResponseToPrepareRequest[None]
A
P
ResponseToPrepareRequest[None]
A
If rx'd a majority,
v = max(rx'd Proposals).v or vdefault

Alice P A L

Bob P A L

Carl P A L

$v_{default}$ = "aliz rulz"

P prepare request: 1 → A

P prepare request: 1 → A

P prepare request: 1 → A

ResponseToPrepareRequest[None]

P ← ResponseToPrepareRequest[None] A

P ← ResponseToPrepareRequest[None] A

If rx'd a majority,
v = max(rx'd Proposals).v or **$v_{default}$**

P AcceptRequest[Proposal[1,"aliz rulz"]] → A

P AcceptRequest[Proposal[1,"aliz rulz"]] → A

Alice P A L

Bob P A L

Carl P A L

$v_{default}$ = "aliz rulz"

P prepare request: 1 → A

P prepare request: 1 → A

P prepare request: 1 → A

ResponseToPrepareRequest[None]

P ← ResponseToPrepareRequest[None] A

P ← ResponseToPrepareRequest[None] A

If rx'd a majority,
v = max(rx'd Proposals).v or **$v_{default}$**

P AcceptRequest[Proposal[1,"aliz rulz"]] → A

P AcceptRequest[Proposal[1,"aliz rulz"]] → A

P AcceptRequest[Proposal[1,"aliz rulz"]] A

Alice P A L

Bob P A L

Carl P A L

$v_{default}$ = "aliz rulz"

P prepare request: 1 → A

P prepare request: 1 → A

P prepare request: 1 → A

ResponseToPrepareRequest[None]

P ← ResponseToPrepareRequest[None] A

P ← ResponseToPrepareRequest[None] A

If rx'd a majority,
v = max(rx'd Proposals).v or **$v_{default}$**

P AcceptRequest[Proposal[1,"aliz rulz"]] → A

P AcceptRequest[Proposal[1,"aliz rulz"]] → A

P AcceptRequest[Proposal[1,"aliz rulz"]] A → Decision[Proposal[n,v]] → L

L ← Decision[Proposal[n,v]]

Decision[Proposal[n,v]] → L

Alice P A L

Bob P A L

Carl P A L

$v_{default}$ = "aliz rulz"

P prepare request: 1 → A

P prepare request: 1 → A

P prepare request: 1 → A

ResponseToPrepareRequest[None]

P ← ResponseToPrepareRequest[None] A

P ← ResponseToPrepareRequest[None] A

If rx'd a majority,
v = max(rx'd Proposals).v or **$v_{default}$**

P AcceptRequest[Proposal[1,"aliz rulz"]] → A

P AcceptRequest[Proposal[1,"aliz rulz"]] → A

P AcceptRequest[Proposal[1,"aliz rulz"]] A → Decision[Proposal[n,v]] → L

L ← Decision[Proposal[n,v]]

Decision[Proposal[n,v]] → L

A

Alice P A L

Bob P A L

Carl P A L

$v_{default}$ = "aliz rulz"

P prepare request: 1 → A

P prepare request: 1 → A

P prepare request: 1 → A

ResponseToPrepareRequest[None]

P ← ResponseToPrepareRequest[None] A

P ← ResponseToPrepareRequest[None] A

If rx'd a majority,
v = max(rx'd Proposals).v or **$v_{default}$**

P AcceptRequest[Proposal[1,"aliz rulz"]] → A

P AcceptRequest[Proposal[1,"aliz rulz"]] → A

P AcceptRequest[Proposal[1,"aliz rulz"]] A → Decision[Proposal[n,v]] → L

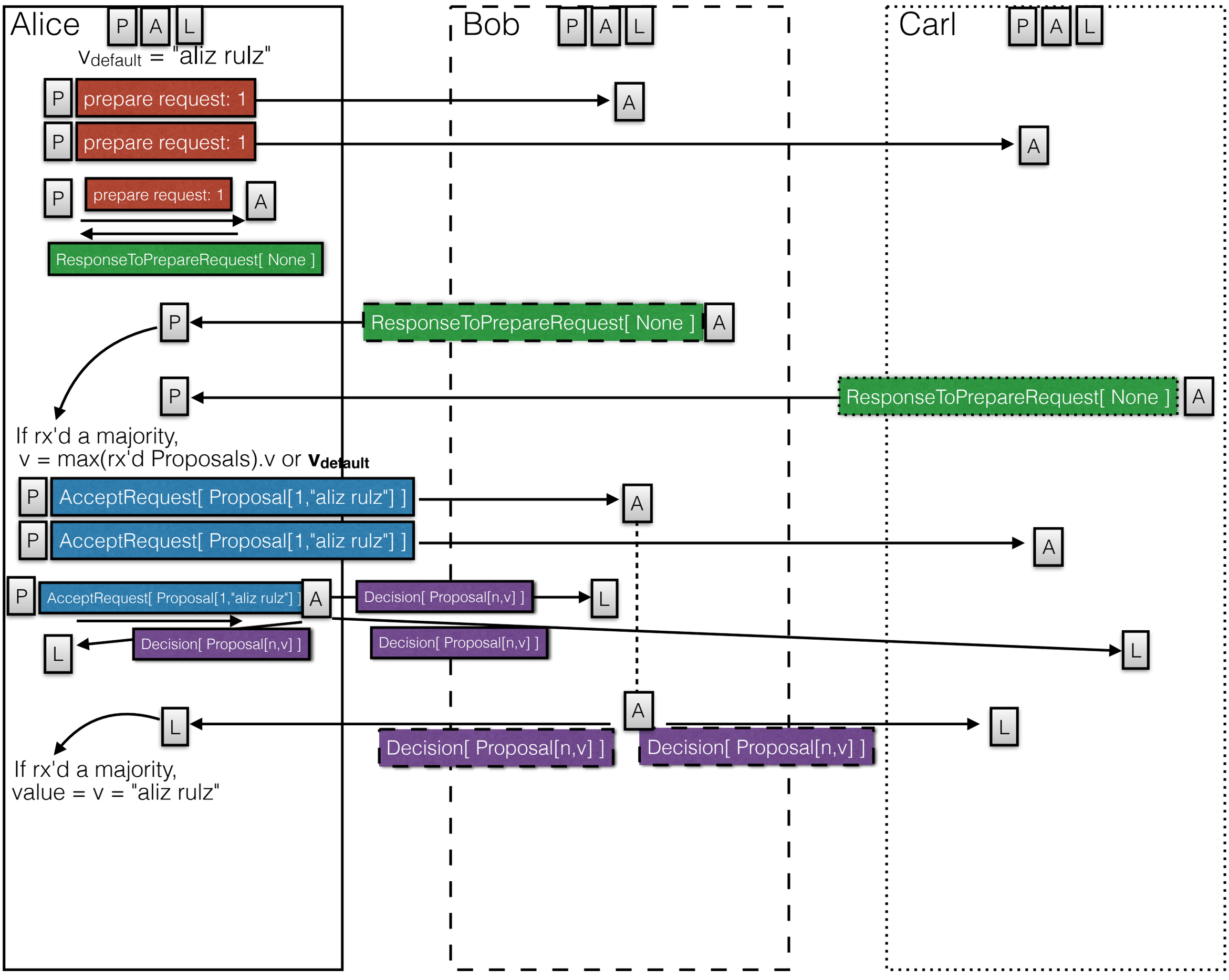
L ← Decision[Proposal[n,v]]

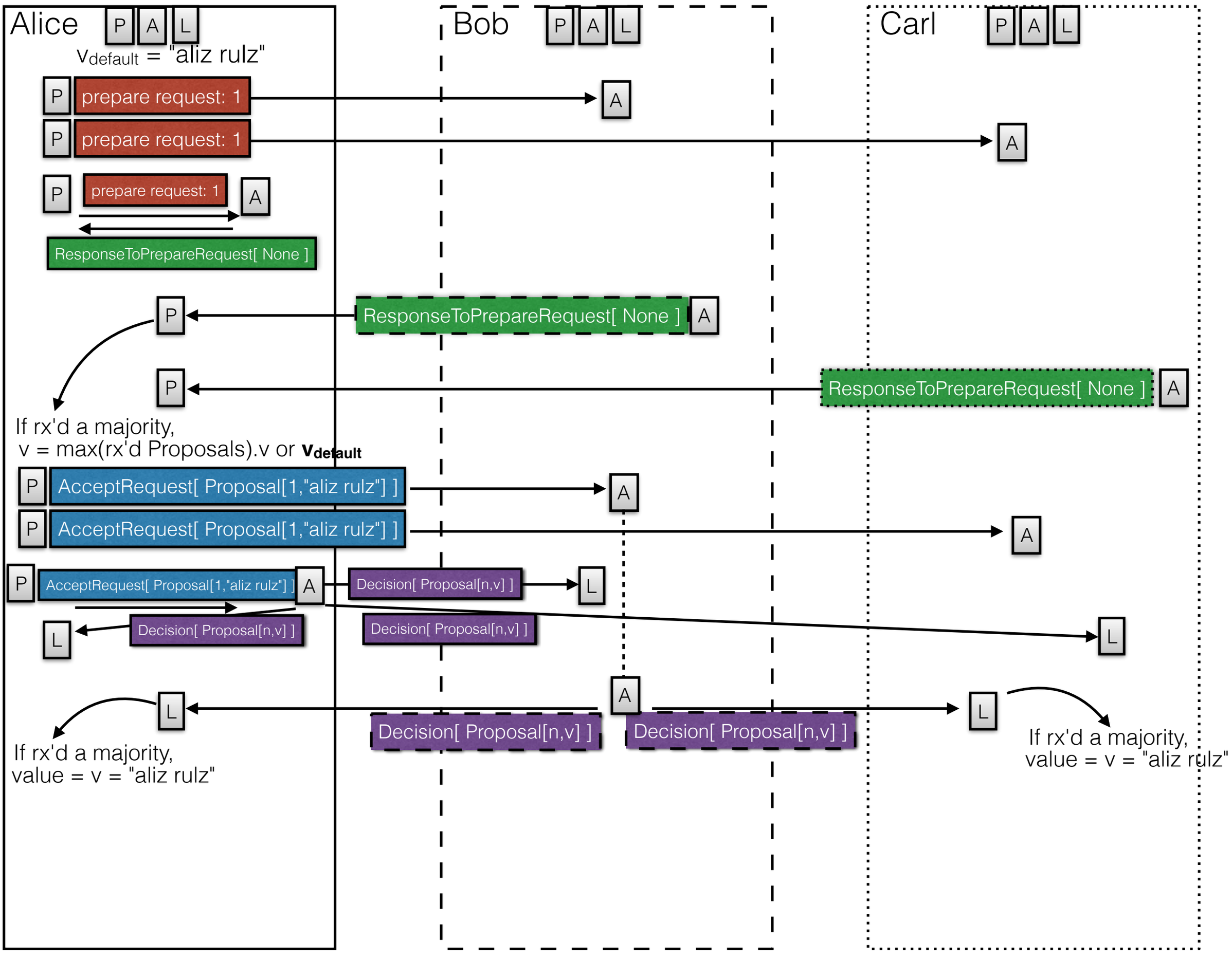
Decision[Proposal[n,v]] → L

L ← A → L

Decision[Proposal[n,v]]

Decision[Proposal[n,v]]


Alice
P A L
Bob
P A L
Carl
P A L
vdefault = "aliz rulz"
P prepare request: 1
A
P prepare request: 1
A
P prepare request: 1
A
ResponseToPrepareRequest[None]
P
ResponseToPrepareRequest[None]
A
P
ResponseToPrepareRequest[None]
A
If rx'd a majority,
v = max(rx'd Proposals).v or vdefault
P AcceptRequest[Proposal[1,"aliz rulz"]]
A
P AcceptRequest[Proposal[1,"aliz rulz"]]
A
P AcceptRequest[Proposal[1,"aliz rulz"]]
A
Decision[Proposal[n,v]]
L
L
Decision[Proposal[n,v]]
Decision[Proposal[n,v]]
L
L
A
Decision[Proposal[n,v]]
Decision[Proposal[n,v]]
L
If rx'd a majority,
value = v = "aliz rulz"


Alice
P
A
L
Bob
P
A
L
Carl
P
A
L
vdefault = "aliz rulz"
P
prepare request: 1
A
P
prepare request: 1
A
P
prepare request: 1
A
ResponseToPrepareRequest[None]
P
ResponseToPrepareRequest[None]
A
P
ResponseToPrepareRequest[None]
A
If rx'd a majority,
v = max(rx'd Proposals).v or vdefault
P
AcceptRequest[Proposal[1,"aliz rulz"]]
A
P
AcceptRequest[Proposal[1,"aliz rulz"]]
A
P
AcceptRequest[Proposal[1,"aliz rulz"]]
A
Decision[Proposal[n,v]]
L
L
Decision[Proposal[n,v]]
Decision[Proposal[n,v]]
L
L
Decision[Proposal[n,v]]
A
Decision[Proposal[n,v]]
L
If rx'd a majority,
value = v = "aliz rulz"
If rx'd a majority,
value = v = "aliz rulz"

Alice P A L

Bob P A L

Carl P A L

$v_{default}$ = "aliz rulz"

P prepare request: 1 → A

P prepare request: 1 → A

P prepare request: 1 → A

ResponseToPrepareRequest[None]

P ← ResponseToPrepareRequest[None] A

P ← ResponseToPrepareRequest[None] A

If rx'd a majority,
v = max(rx'd Proposals).v or **$v_{default}$**

P AcceptRequest[Proposal[1,"aliz rulz"]] → A

P AcceptRequest[Proposal[1,"aliz rulz"]] → A

P AcceptRequest[Proposal[1,"aliz rulz"]] A → Decision[Proposal[n,v]] → L

L ← Decision[Proposal[n,v]]

Decision[Proposal[n,v]]

L

L ← A → Decision[Proposal[n,v]] Decision[Proposal[n,v]] → L

Decision[Proposal[n,v]]

If rx'd a majority,
value = v = "aliz rulz"

If rx'd a majority,
value = v = "aliz rulz"

If rx'd a majority,
value = v = "aliz rulz"

Alice P A L

Bob P A L

Carl P A L

$v_{default}$ = "aliz rulz"

P prepare request: 1 → A

P prepare request: 1 → A

P prepare request: 1 → A

ResponseToPrepareRequest[None]

P ← ResponseToPrepareRequest[None] A

P ← ResponseToPrepareRequest[None] A

If rx'd a majority,
v = max(rx'd Proposals).v or **$v_{default}$**

P AcceptRequest[Proposal[1,"aliz rulz"]] → A

P AcceptRequest[Proposal[1,"aliz rulz"]] → A

P AcceptRequest[Proposal[1,"aliz rulz"]] A → Decision[Proposal[n,v]] → L

L ← Decision[Proposal[n,v]]

Decision[Proposal[n,v]]

L

A

Decision[Proposal[n,v]] Decision[Proposal[n,v]]

Decision[Proposal[n,v]]

L ← A → L

If rx'd a majority,
value = v = "aliz rulz"

L

If rx'd a majority,
value = v = "aliz rulz"

A

If rx'd a majority,
value = v = "aliz rulz"

Alice P A L

Bob P A L

Carl P A L

$v_{default}$ = "aliz rulz"

P prepare request: 1 → A

P prepare request: 1 → A

P prepare request: 1 → A

ResponseToPrepareRequest[None]

P ← ResponseToPrepareRequest[None] A

P ← ResponseToPrepareRequest[None] A

If rx'd a majority,
v = max(rx'd Proposals).v or **$v_{default}$**

P AcceptRequest[Proposal[1,"aliz rulz"]] → A

P AcceptRequest[Proposal[1,"aliz rulz"]] → A

P AcceptRequest[Proposal[1,"aliz rulz"]] A → Decision[Proposal[n,v]] → L

L ← Decision[Proposal[n,v]]

Decision[Proposal[n,v]]

→ L

L ← A → Decision[Proposal[n,v]] Decision[Proposal[n,v]] → L

If rx'd a majority,
value = v = "aliz rulz"

Decision[Proposal[n,v]]

L

If rx'd a majority,
value = v = "aliz rulz"

If rx'd a majority,
value = v = "aliz rulz"

A

L ← Decision[Proposal[n,v]]

Decision[Proposal[n,v]]

L

Alice P A L

Bob P A L

Carl P A L

$v_{default}$ = "aliz rulz"

P prepare request: 1 → A

P prepare request: 1 → A

P prepare request: 1 A

ResponseToPrepareRequest[None]

P ← ResponseToPrepareRequest[None] A

P ← ResponseToPrepareRequest[None] A

If rx'd a majority,
v = max(rx'd Proposals).v or **$v_{default}$**

P AcceptRequest[Proposal[1,"aliz rulz"]] → A

P AcceptRequest[Proposal[1,"aliz rulz"]] → A

P AcceptRequest[Proposal[1,"aliz rulz"]] A → Decision[Proposal[n,v]] → L

L ← Decision[Proposal[n,v]]

Decision[Proposal[n,v]]

→ L

L

If rx'd a majority,
value = v = "aliz rulz"

A

Decision[Proposal[n,v]] Decision[Proposal[n,v]]

Decision[Proposal[n,v]]

L → If rx'd a majority,
value = v = "aliz rulz"

L

If rx'd a majority,
value = v = "aliz rulz"

L

If rx'd a majority,
value = v = "aliz rulz"

A

Decision[Proposal[n,v]]

Decision[Proposal[n,v]]

L

If rx'd a majority,
value = v = "aliz rulz"

Alice P A L

Bob P A L

Carl P A L

$v_{default}$ = "aliz rulz"

P prepare request: 1 → A

P prepare request: 1 → A

P prepare request: 1 A

ResponseToPrepareRequest[None]

P ← ResponseToPrepareRequest[None] A

P ← ResponseToPrepareRequest[None] A

If rx'd a majority,
v = max(rx'd Proposals).v or **$v_{default}$**

P AcceptRequest[Proposal[1,"aliz rulz"]] → A

P AcceptRequest[Proposal[1,"aliz rulz"]] → A

P AcceptRequest[Proposal[1,"aliz rulz"]] A Decision[Proposal[n,v]] → L

L ← Decision[Proposal[n,v]]

Decision[Proposal[n,v]]

→ L

L

If rx'd a majority,
value = v = "aliz rulz"

A

Decision[Proposal[n,v]] Decision[Proposal[n,v]]

Decision[Proposal[n,v]]

L

If rx'd a majority,
value = v = "aliz rulz"

L

If rx'd a majority,
value = v = "aliz rulz"

L

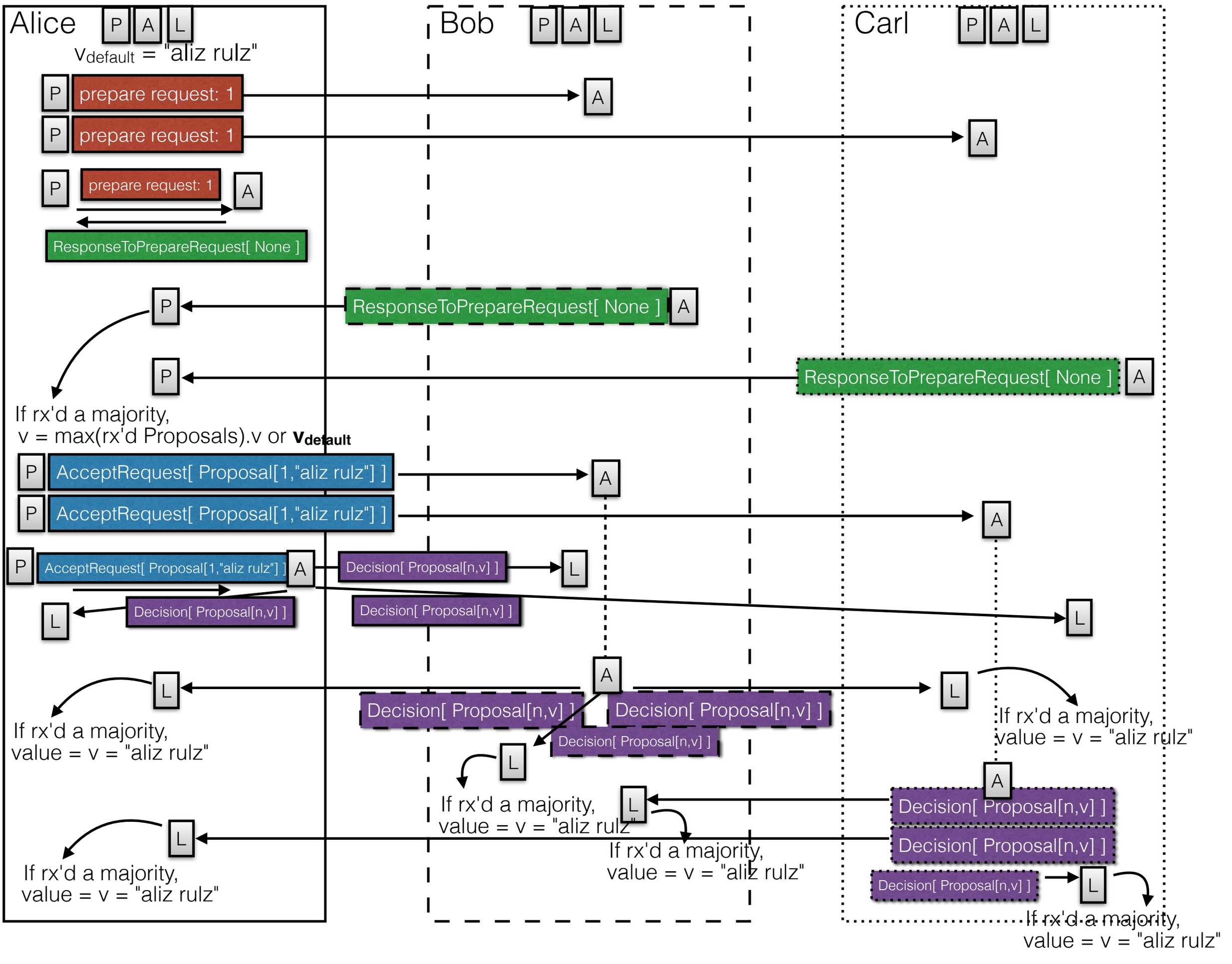
If rx'd a majority,
value = v = "aliz rulz"

A

Decision[Proposal[n,v]]

Decision[Proposal[n,v]]

Decision[Proposal[n,v]] → L


Alice
P
A
L
Bob
P
A
L
Carl
P
A
L
vdefault = "aliz rulz"
P
prepare request: 1
A
P
prepare request: 1
A
P
prepare request: 1
A
ResponseToPrepareRequest[None]
P
ResponseToPrepareRequest[None]
A
P
ResponseToPrepareRequest[None]
A
If rx'd a majority,
v = max(rx'd Proposals).v or vdefault
P
AcceptRequest[Proposal[1,"aliz rulz"]]
A
P
AcceptRequest[Proposal[1,"aliz rulz"]]
A
P
AcceptRequest[Proposal[1,"aliz rulz"]]
A
Decision[Proposal[n,v]]
L
L
Decision[Proposal[n,v]]
Decision[Proposal[n,v]]
L
L
A
Decision[Proposal[n,v]]
Decision[Proposal[n,v]]
Decision[Proposal[n,v]]
L
L
If rx'd a majority,
value = v = "aliz rulz"
If rx'd a majority,
value = v = "aliz rulz"
L
If rx'd a majority,
value = v = "aliz rulz"
A
Decision[Proposal[n,v]]
Decision[Proposal[n,v]]
Decision[Proposal[n,v]]
L
If rx'd a majority,
value = v = "aliz rulz"
If rx'd a majority,
value = v = "aliz rulz"

Outline

- The Algorithm
- Example: how it works **initially**

- Example: how it handles **conflicts**

- Example: how it works **after consensus**

Alice P A L

Bob P A L

Carl P A L

Alice P A L

$v_{default}$ = "aliz rulz"

Bob P A L

Carl P A L

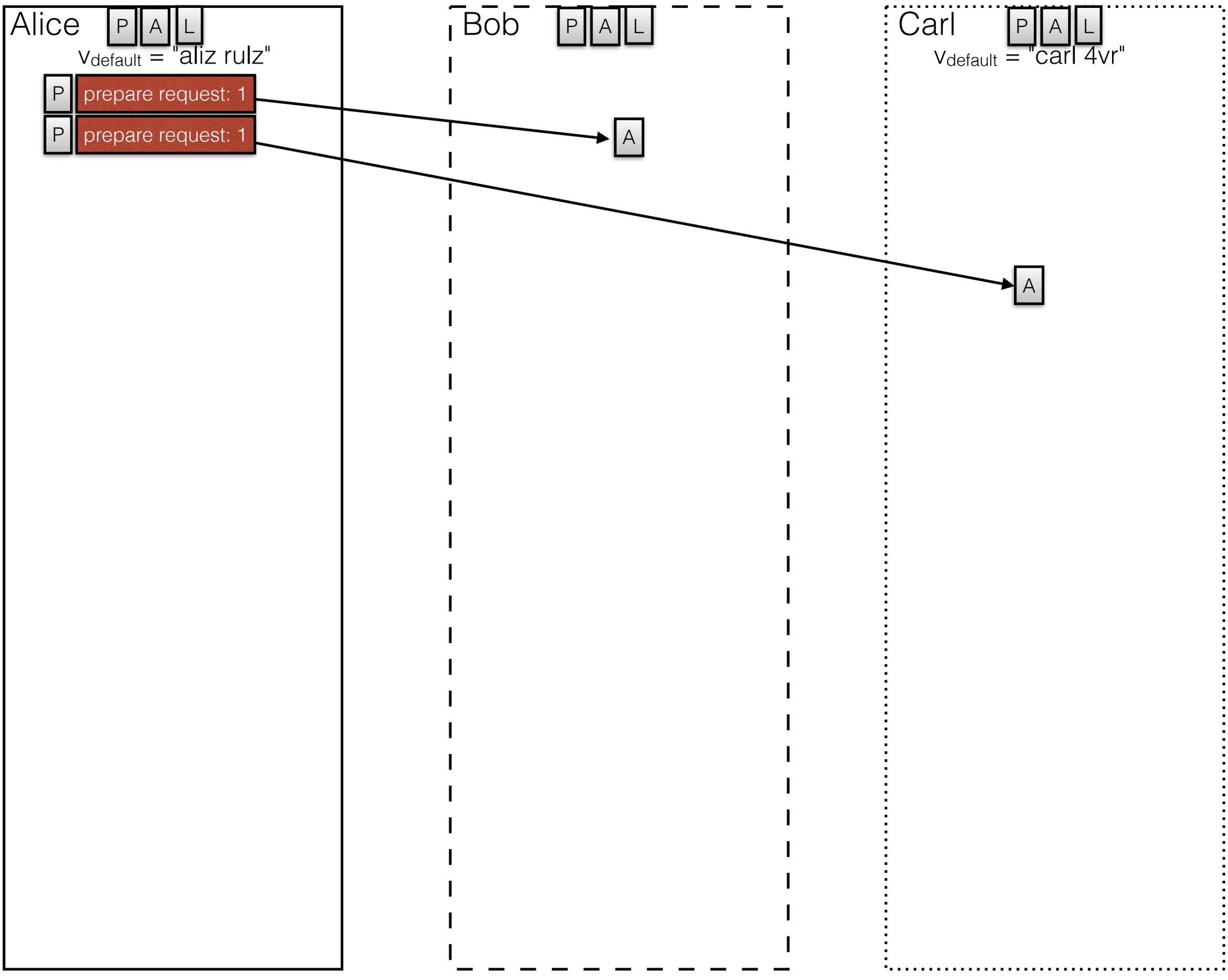
Alice P A L

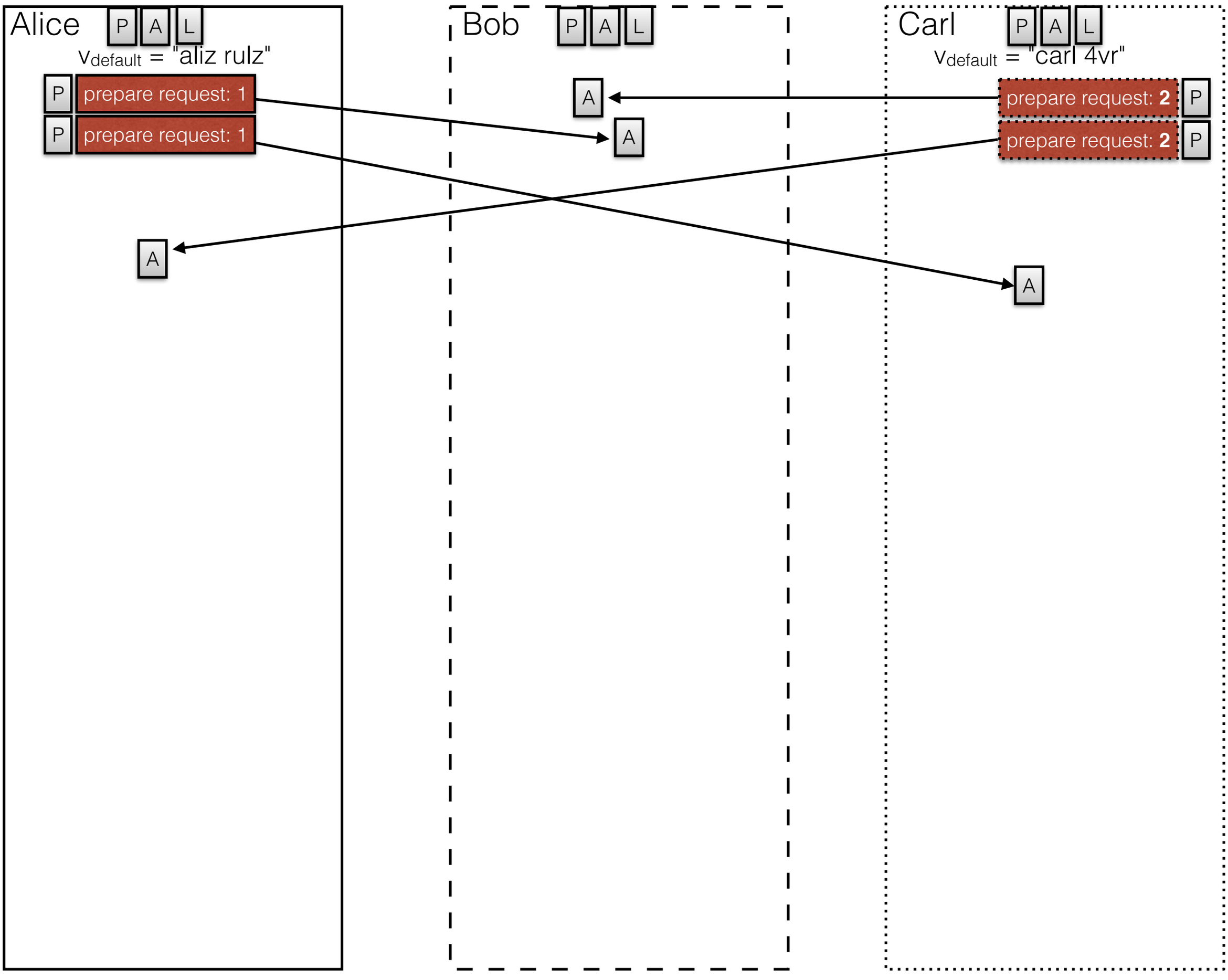
$v_{default}$ = "aliz rulz"

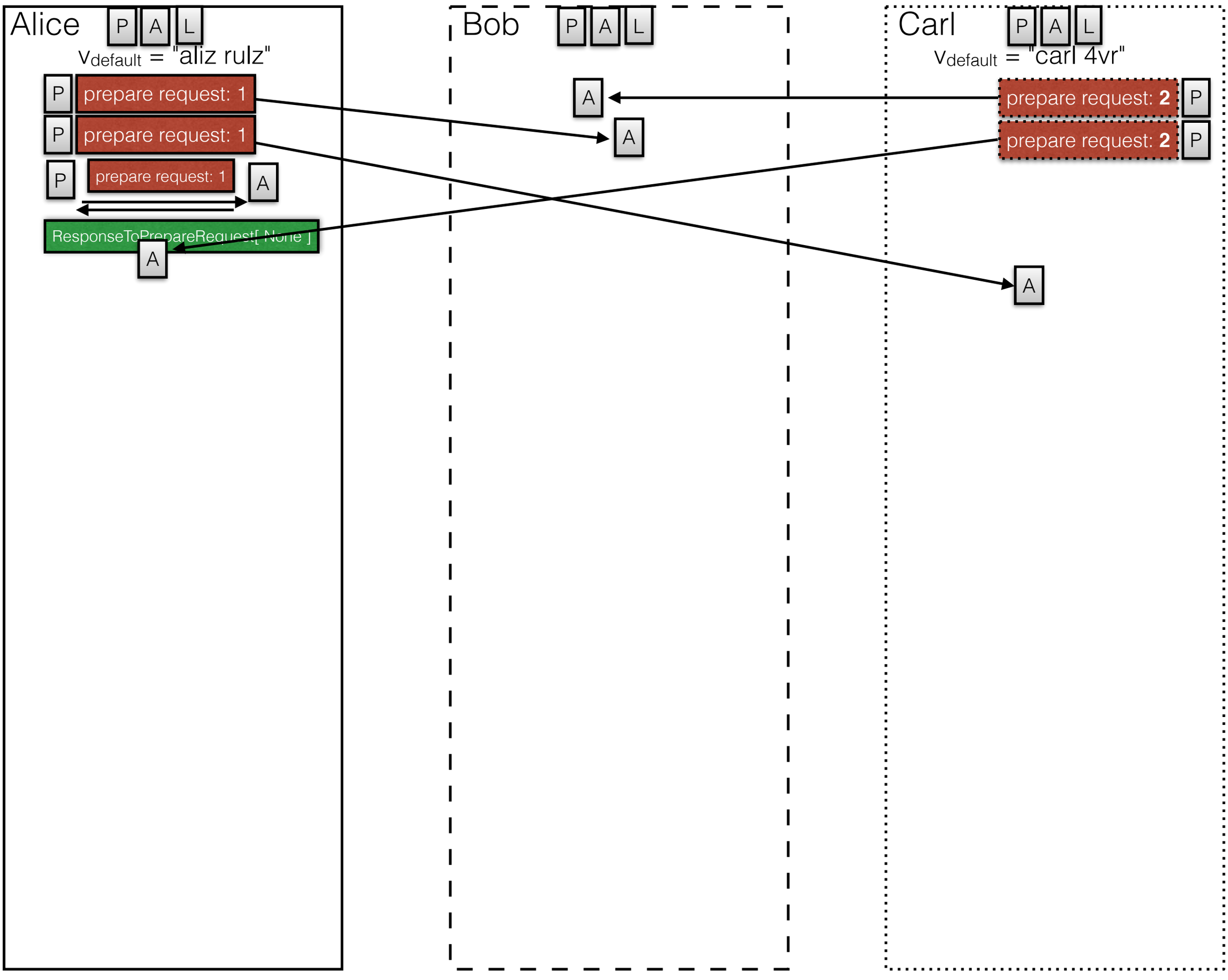
Bob P A L

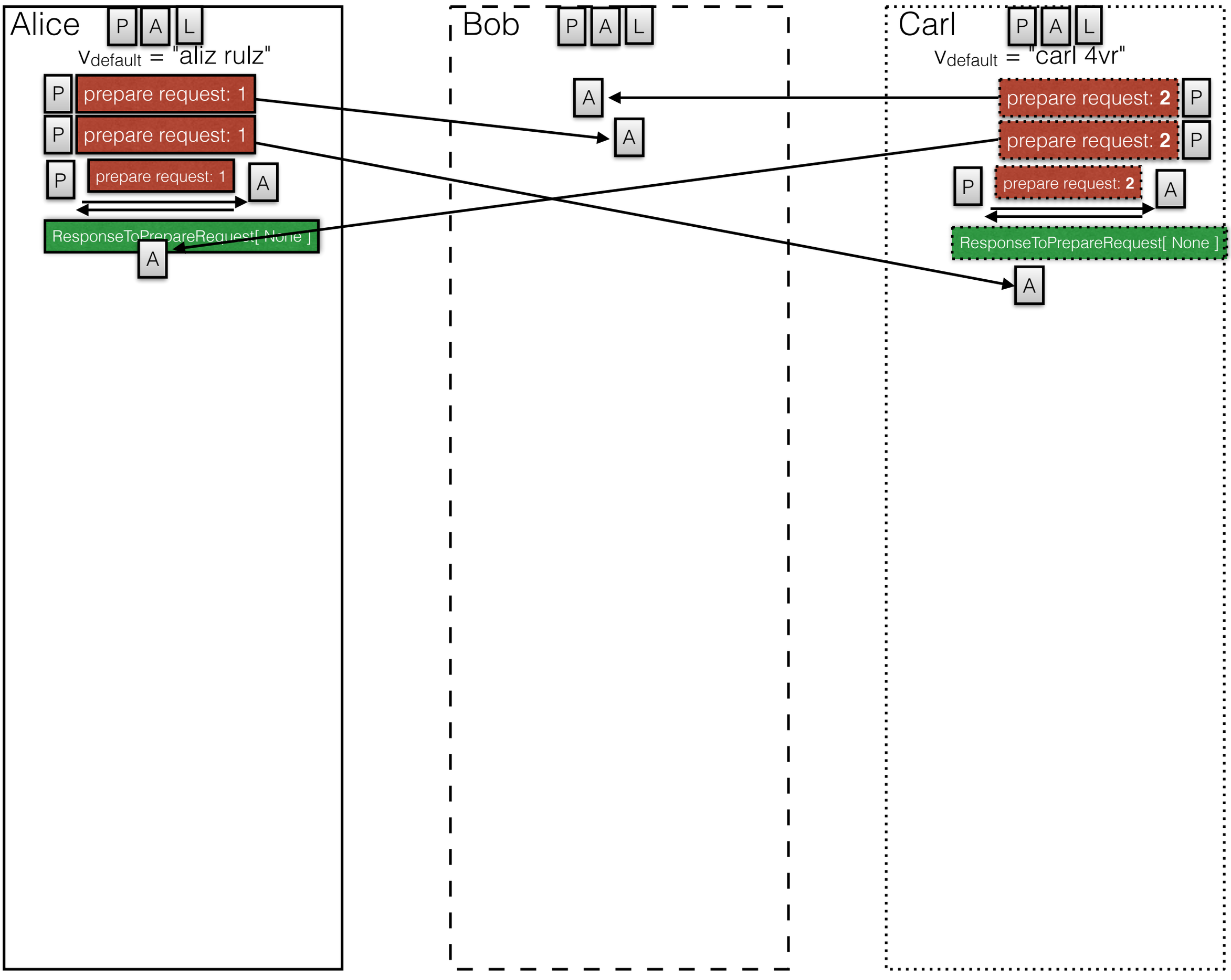
Carl P A L

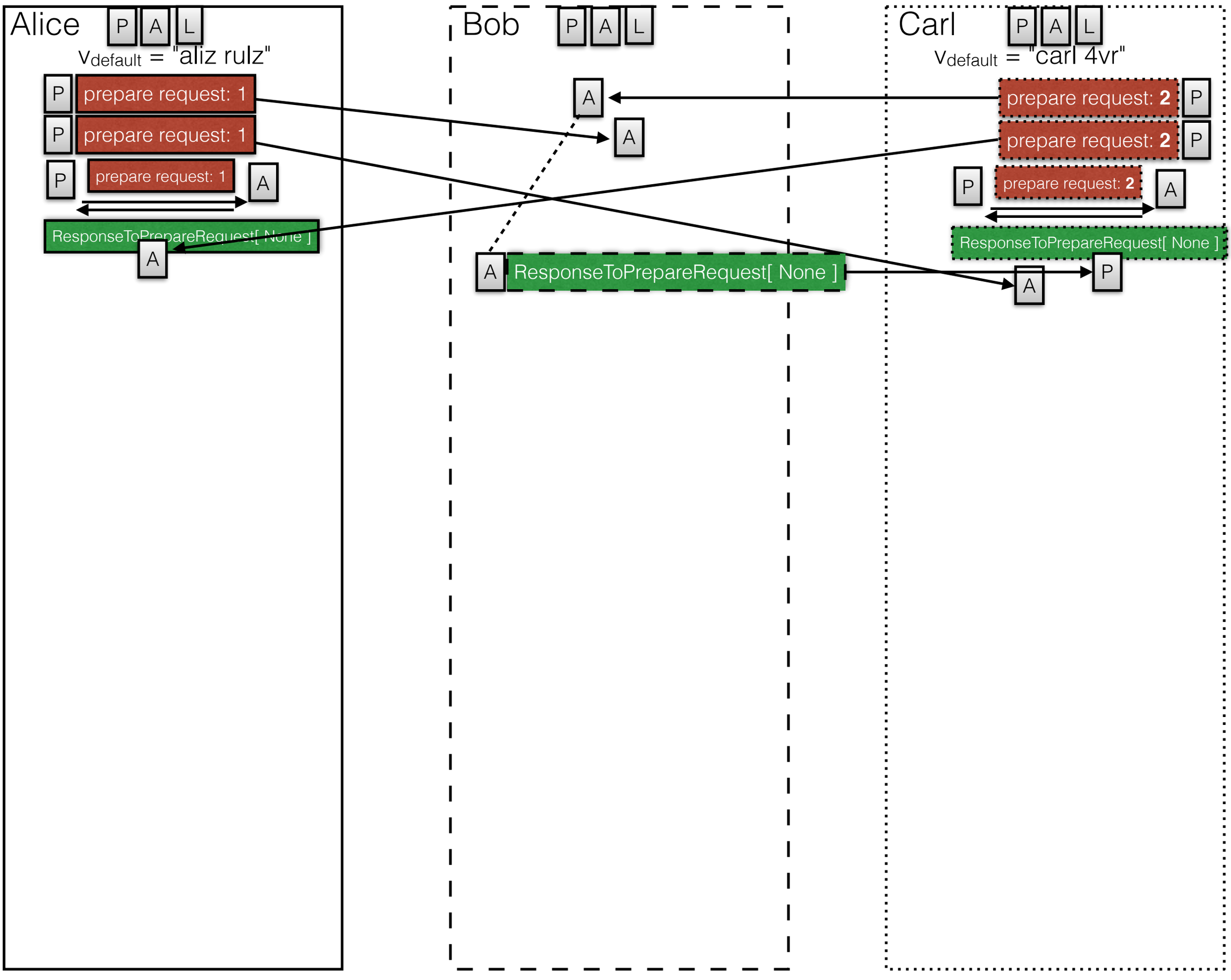
$v_{default}$ = "carl 4vr"

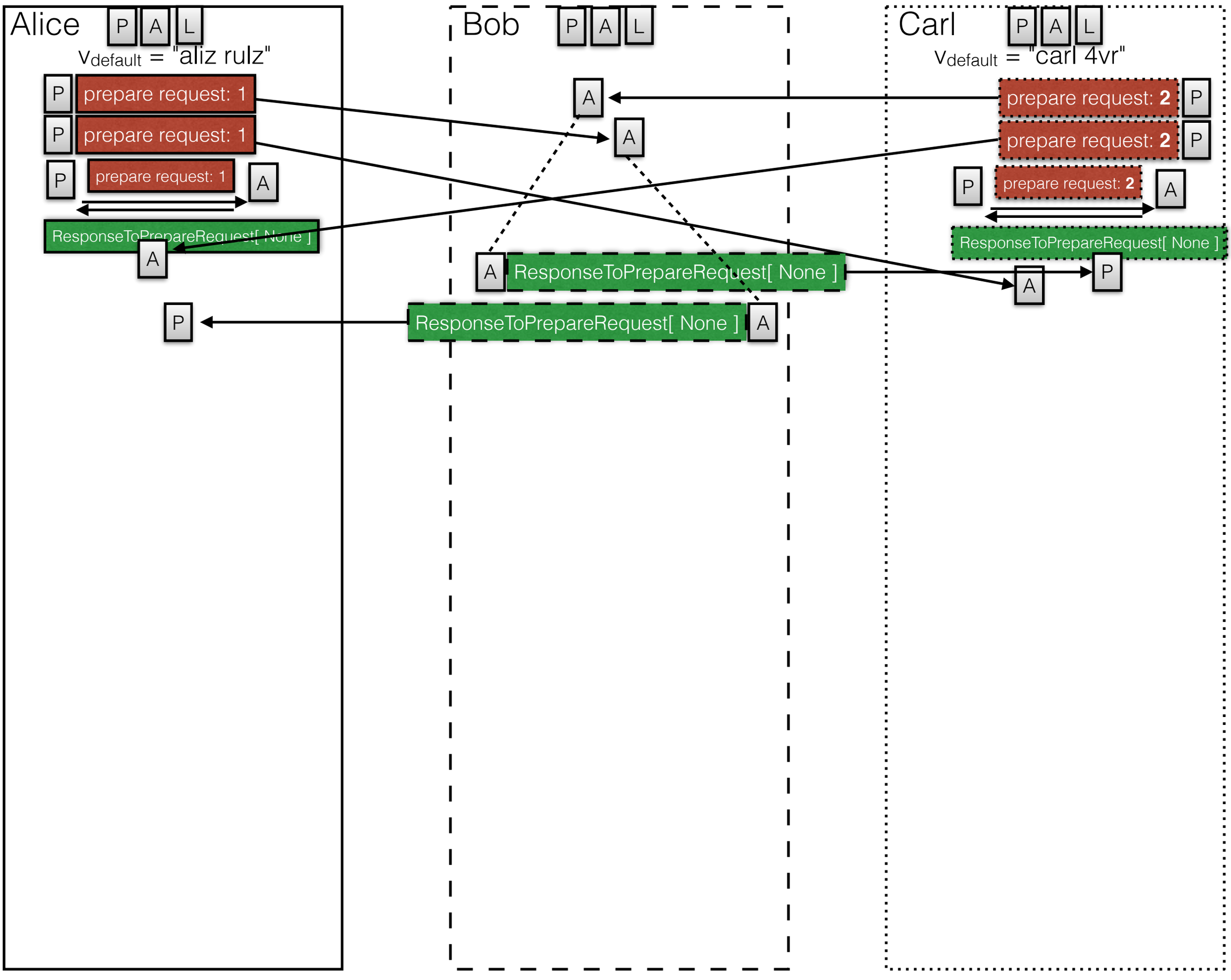

Alice
P
A
L
vdefault = "aliz rulz"
P
prepare request: 1
P
prepare request: 1
Bob
P
A
L
A
Carl
P
A
L
vdefault = "carl 4vr"
A

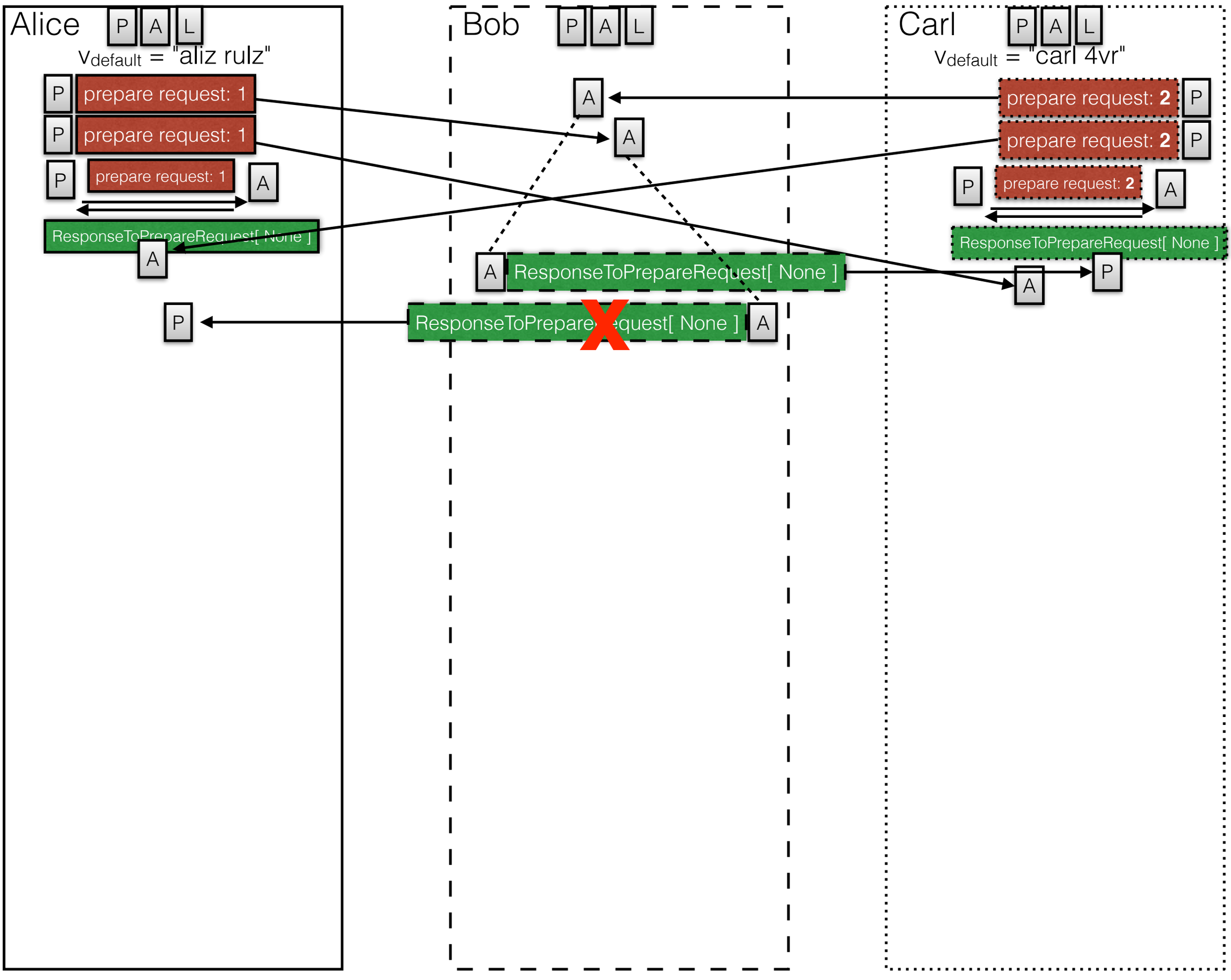

Alice
P
A
L
vdefault = "aliz rulz"
P
prepare request: 1
P
prepare request: 1
A
Bob
P
A
L
A
A
Carl
P
A
L
vdefault = "carl 4vr"
prepare request: 2
P
prepare request: 2
P
A

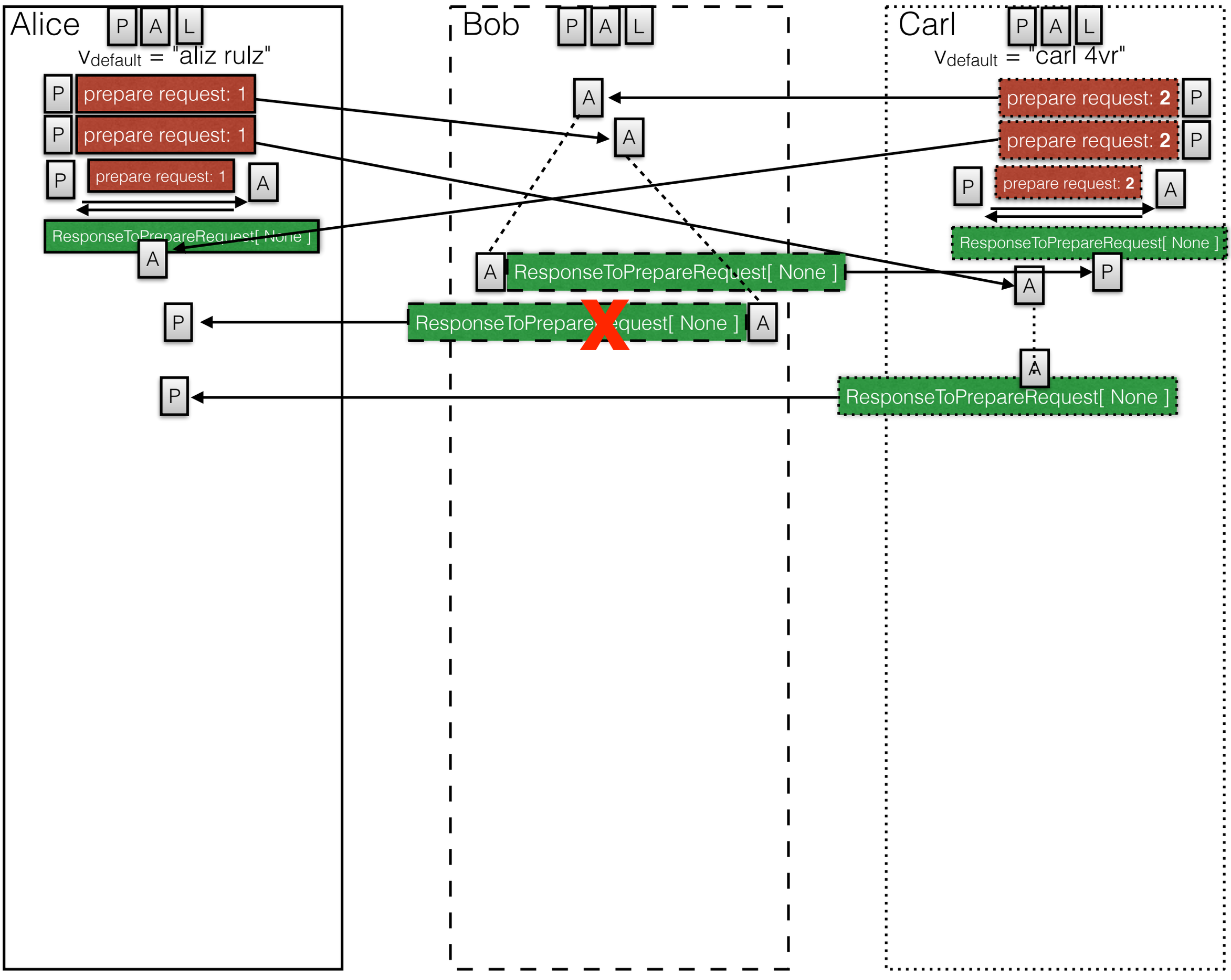

Alice
P
A
L
vdefault = "aliz rulz"
P
prepare request: 1
P
prepare request: 1
P
prepare request: 1
A
ResponseToPrepareRequest[None]
A
Bob
P
A
L
A
A
Carl
P
A
L
vdefault = "carl 4vr"
prepare request: 2
P
prepare request: 2
P
A

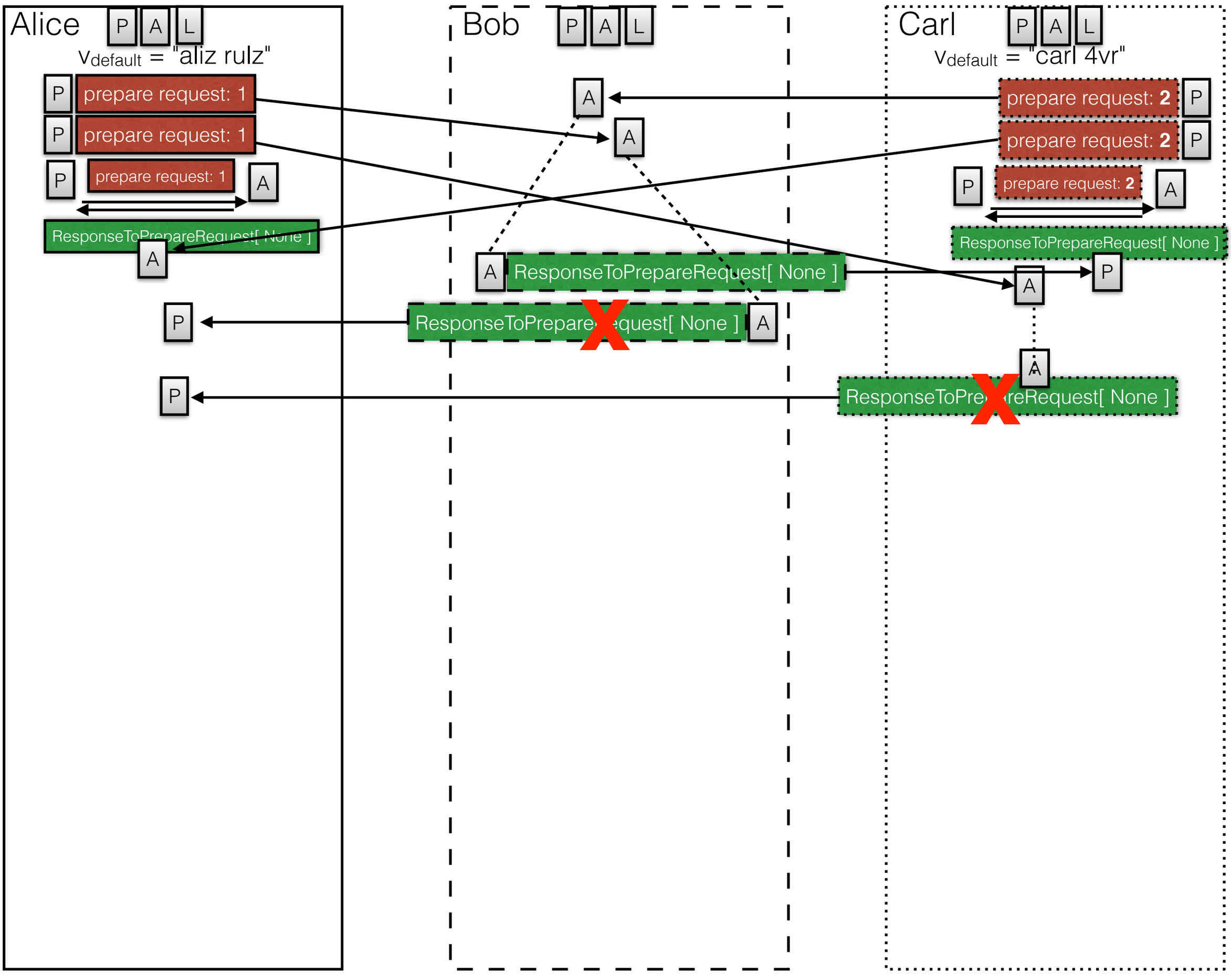

Alice
P
A
L
vdefault = "aliz rulz"
P
prepare request: 1
P
prepare request: 1
P
prepare request: 1
A
ResponseToPrepareRequest[None]
A
Bob
P
A
L
A
A
Carl
P
A
L
vdefault = "carl 4vr"
prepare request: 2
P
prepare request: 2
P
P
prepare request: 2
A
ResponseToPrepareRequest[None]
A

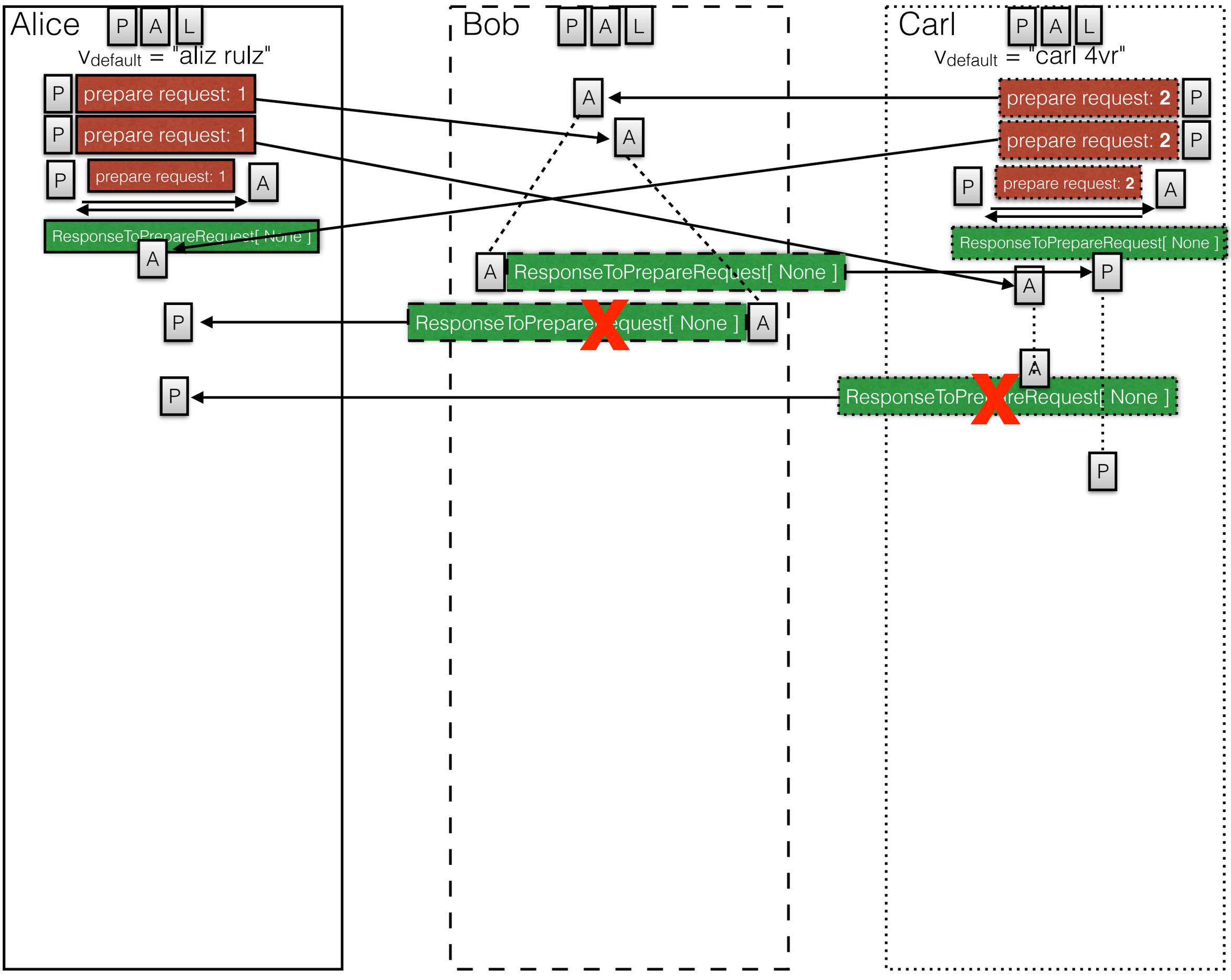

Alice
P
A
L
vdefault = "aliz rulz"
P
prepare request: 1
P
prepare request: 1
P
prepare request: 1
A
ResponseToPrepareRequest[None]
A
Bob
P
A
L
A
A
A
ResponseToPrepareRequest[None]
Carl
P
A
L
vdefault = "carl 4vr"
prepare request: 2
P
prepare request: 2
P
P
prepare request: 2
A
ResponseToPrepareRequest[None]
A
P

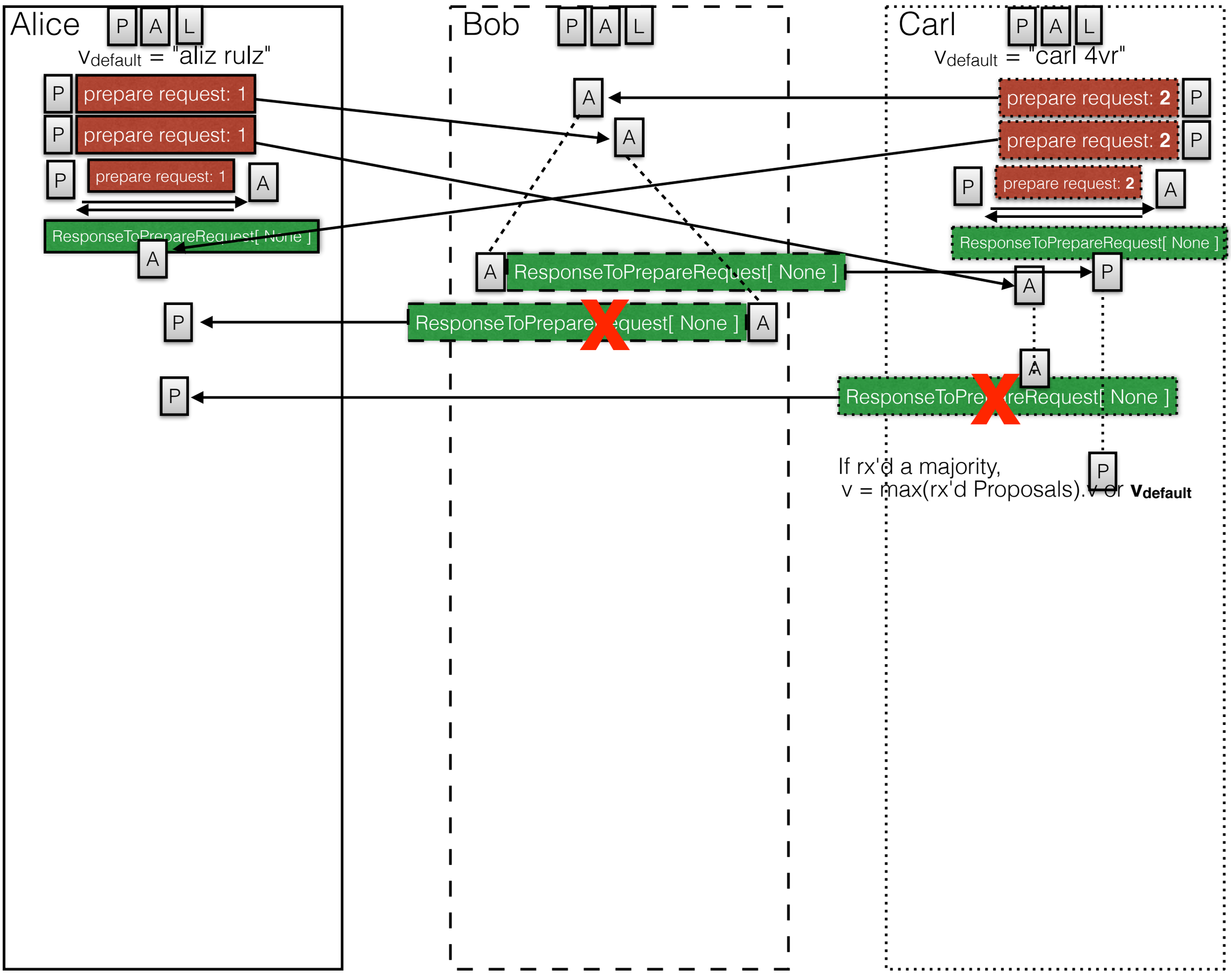

Alice
P
A
L
vdefault = "aliz rulz"
Bob
P
A
L
Carl
P
A
L
vdefault = "carl 4vr"
P
prepare request: 1
P
prepare request: 1
P
prepare request: 1
A
ResponseToPrepareRequest[None]
A
P
A
A
prepare request: 2
P
prepare request: 2
P
P
prepare request: 2
A
ResponseToPrepareRequest[None]
A
ResponseToPrepareRequest[None]
P
A
P
ResponseToPrepareRequest[None]
A


Alice
P
A
L
vdefault = "aliz rulz"
Bob
Carl
vdefault = "carl 4vr"
prepare request: 1
prepare request: 2
ResponseToPrepareRequest[None]


Alice
P
A
L
vdefault = "aliz rulz"
Bob
P
A
L
Carl
P
A
L
vdefault = "carl 4vr"
P
prepare request: 1
P
prepare request: 1
P
prepare request: 1
A
ResponseToPrepareRequest[None]
A
A
A
A
prepare request: 2
P
prepare request: 2
P
P
prepare request: 2
A
ResponseToPrepareRequest[None]
A
ResponseToPrepareRequest[None]
P
A
P
ResponseToPrepare equest[None]
A
A
P
A
ResponseToPrepareRequest[None]


Alice
P
A
L
vdefault = "aliz rulz"
Bob
P
A
L
Carl
P
A
L
vdefault = "carl 4vr"
P
prepare request: 1
P
prepare request: 1
P
prepare request: 1
A
ResponseToPrepareRequest[None]
A
A
A
prepare request: 2
P
prepare request: 2
P
P
prepare request: 2
A
ResponseToPrepareRequest[None]
A
ResponseToPrepareRequest[None]
P
A
P
ResponseToPrepare equest[None]
A
X
P
A
ResponseToPre reRequest[None]
X


Alice
P
A
L
vdefault = "aliz rulz"
Bob
Carl
vdefault = "carl 4vr"
prepare request: 1
prepare request: 2
ResponseToPrepareRequest[None]


Alice
P A L
vdefault = "aliz rulz"
Bob
P A L
Carl
P A L
vdefault = "carl 4vr"
P prepare request: 1
P prepare request: 1
P prepare request: 1
A
ResponseToPrepareRequest[None]
A
A
A
prepare request: 2 P
prepare request: 2 P
P prepare request: 2
A
ResponseToPrepareRequest[None]
A ResponseToPrepareRequest[None]
P
A
P
ResponseToPrepare equest[None] A
X
A
P
ResponseToPre reRequest[None]
X
P
If rx'd a majority,
v = max(rx'd Proposals).v or vdefault

Alice P A L

$v_{default}$ = "aliz rulz"

Bob P A L

Carl P A L

$v_{default}$ = "carl 4vr"

P prepare request: 1

P prepare request: 1

P prepare request: 1 A

ResponseToPrepareRequest[None] A

A

A

prepare request: **2** P

prepare request: **2** P

P prepare request: **2** A

ResponseToPrepareRequest[None]

A ResponseToPrepareRequest[None]

P

ResponseToPrepareRequest[None] A

A P

P

A

ResponseToPrepareRequest[None]

If rx'd a majority,
v = max(rx'd Proposals).v or **$v_{default}$**

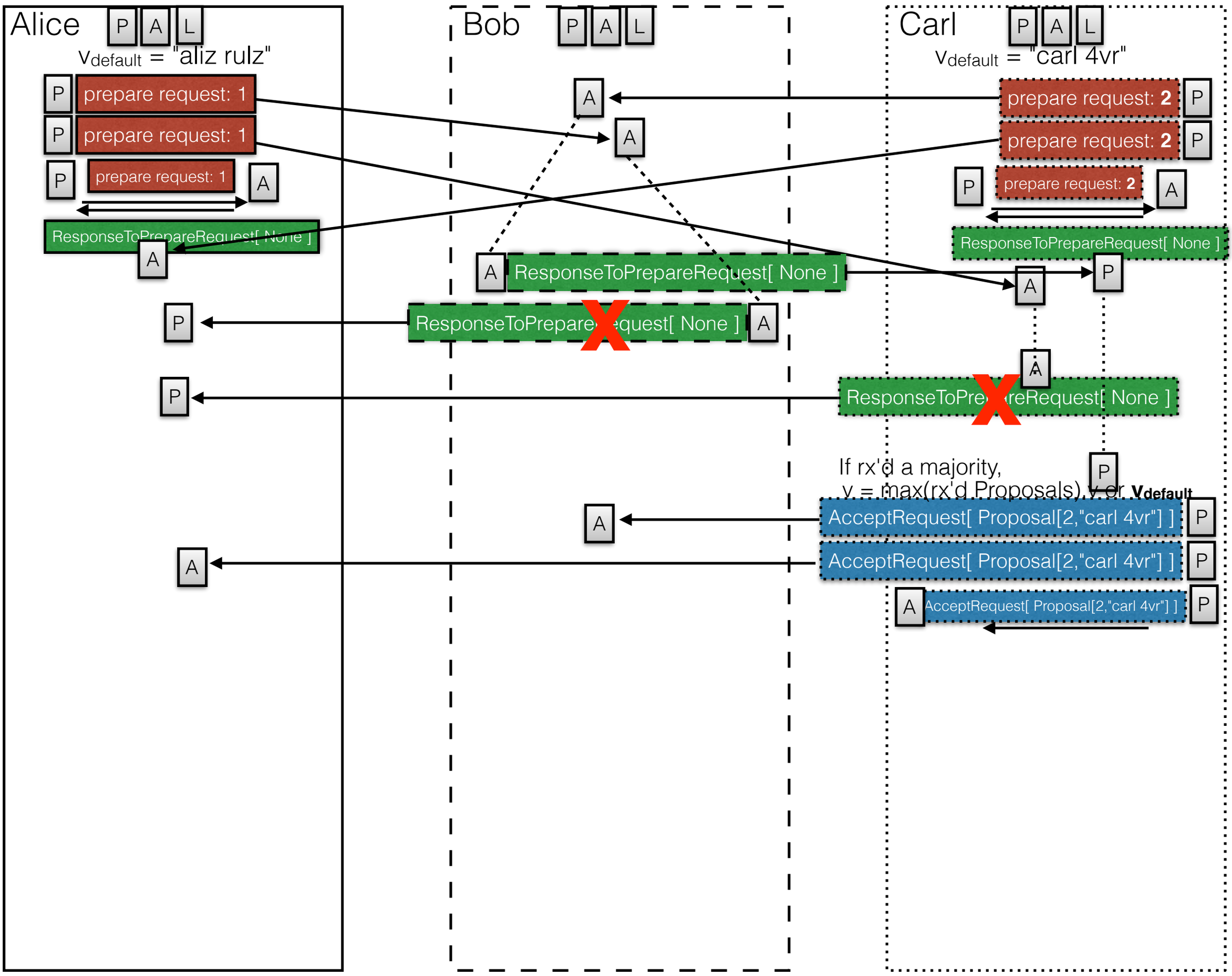
P

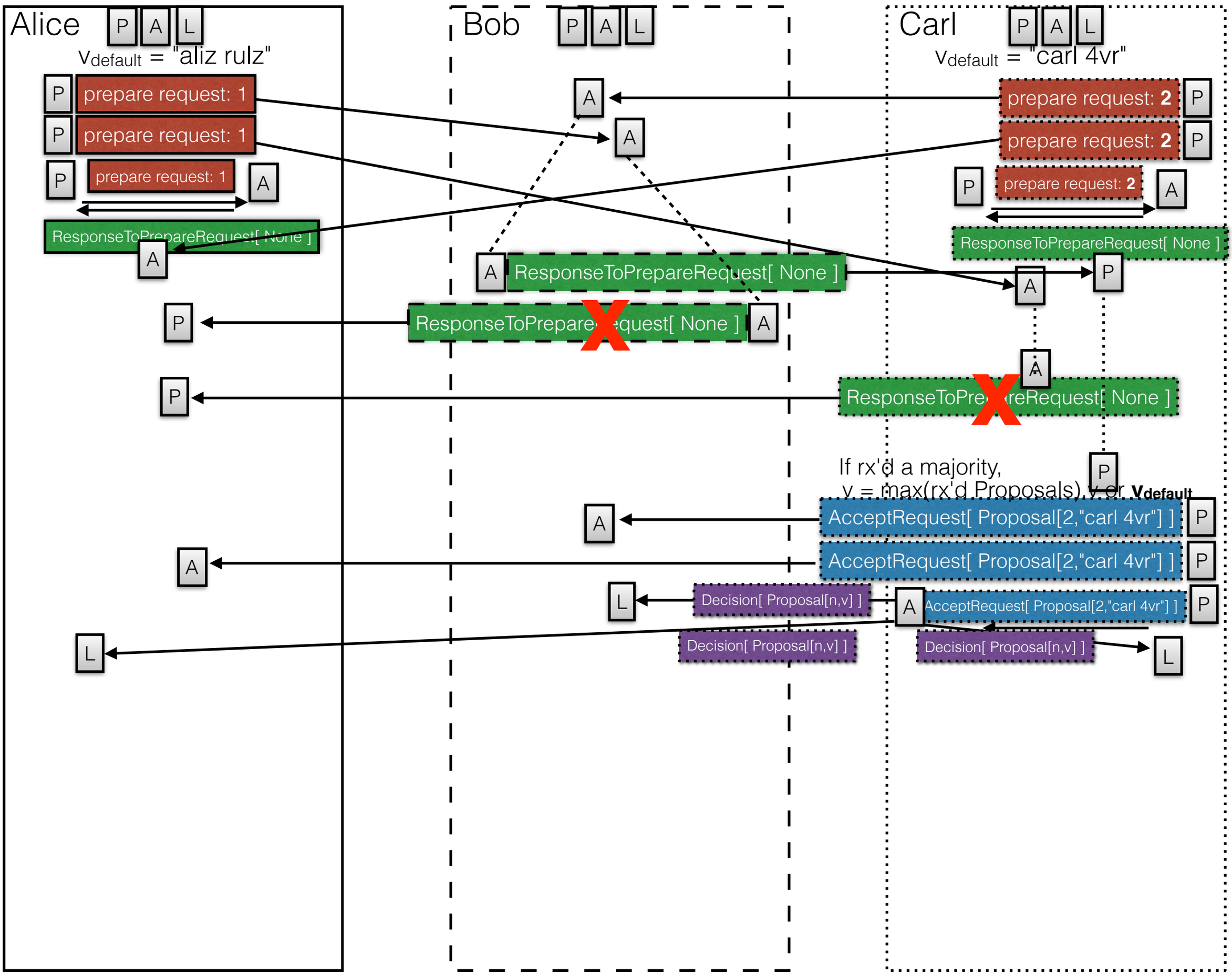
A

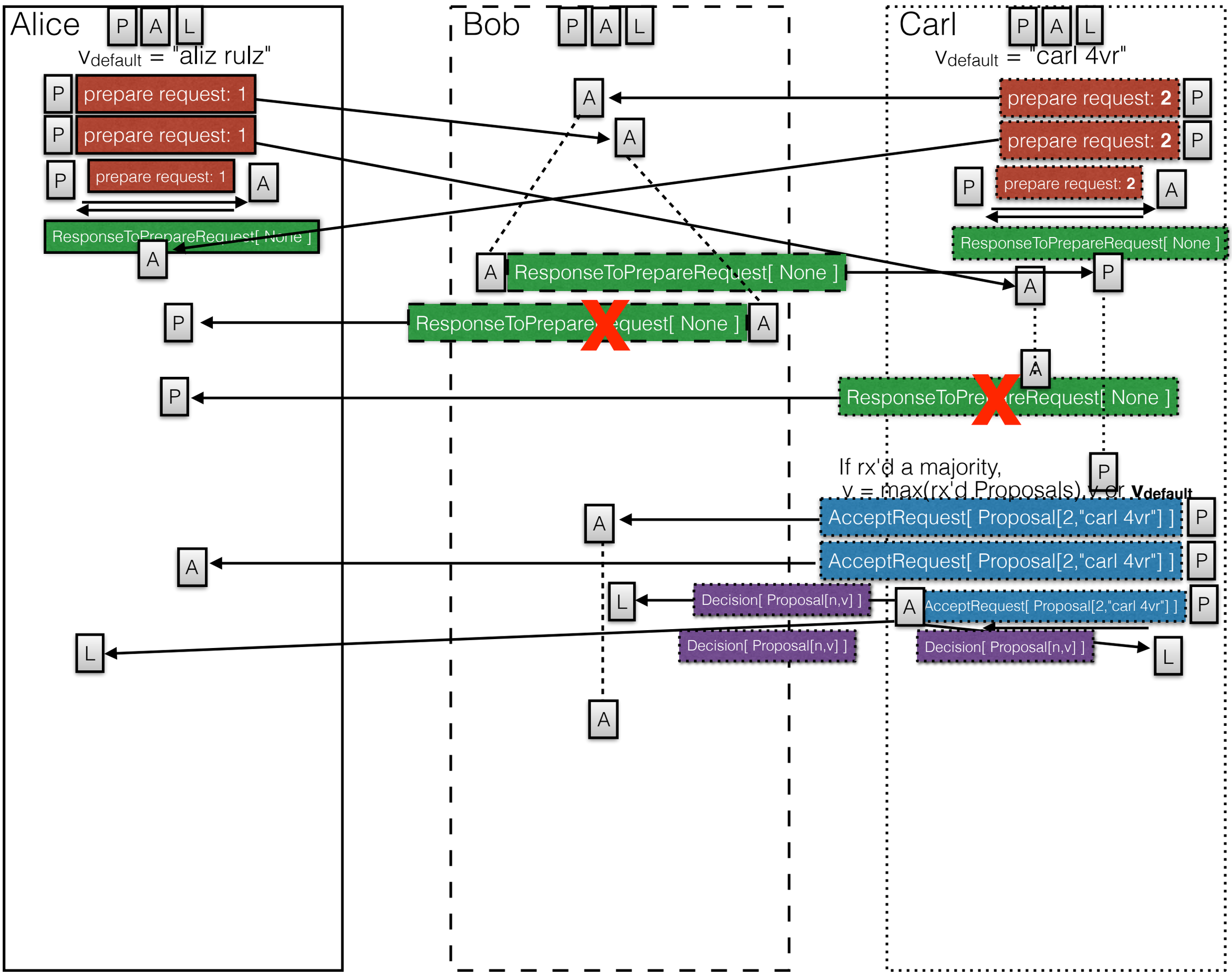
AcceptRequest[Proposal[2,"carl 4vr"]] P

A

AcceptRequest[Proposal[2,"carl 4vr"]] P


Alice
P A L
vdefault = "aliz rulz"
Bob
P A L
Carl
P A L
vdefault = "carl 4vr"
P prepare request: 1
P prepare request: 1
P prepare request: 1
A
ResponseToPrepareRequest[None]
A
A
A
prepare request: 2 P
prepare request: 2 P
P prepare request: 2 A
ResponseToPrepareRequest[None]
A ResponseToPrepareRequest[None]
P
A
P
ResponseToPrepare equest[None] A
A
P
A
ResponseToPre reRequest[None]
P
If rx'd a majority,
v = max(rx'd Proposals).v or vdefault
A
AcceptRequest[Proposal[2,"carl 4vr"]] P
A
AcceptRequest[Proposal[2,"carl 4vr"]] P
A AcceptRequest[Proposal[2,"carl 4vr"]] P


Alice
P A L
vdefault = "aliz rulz"
Bob
P A L
Carl
P A L
vdefault = "carl 4vr"
P prepare request: 1
P prepare request: 1
P prepare request: 1
A
ResponseToPrepareRequest[None]
A
A
A
prepare request: 2 P
prepare request: 2 P
P prepare request: 2
A
ResponseToPrepareRequest[None]
A ResponseToPrepareRequest[None]
P
A
P
ResponseToPrepare equest[None] A
X
P
A
ResponseToPre reRequest[None]
X
If rx'd a majority,
v = max(rx'd Proposals).v or vdefault
P
A
AcceptRequest[Proposal[2,"carl 4vr"]] P
A
AcceptRequest[Proposal[2,"carl 4vr"]] P
L
Decision[Proposal[n,v]]
A
AcceptRequest[Proposal[2,"carl 4vr"]] P
L
Decision[Proposal[n,v]]
Decision[Proposal[n,v]]
L


Alice
P A L
vdefault = "aliz rulz"
Bob
P A L
Carl
P A L
vdefault = "carl 4vr"
P prepare request: 1
P prepare request: 1
P prepare request: 1
A
ResponseToPrepareRequest[None]
A
A
A
prepare request: 2 P
prepare request: 2 P
P prepare request: 2
A
ResponseToPrepareRequest[None]
A ResponseToPrepareRequest[None]
P
A
P
ResponseToPrepare equest[None] A
X
P
A
ResponseToPre reRequest[None]
X
If rx'd a majority,
v = max(rx'd Proposals).v or vdefault
P
A
AcceptRequest[Proposal[2,"carl 4vr"]] P
A
AcceptRequest[Proposal[2,"carl 4vr"]] P
L
Decision[Proposal[n,v]]
A
AcceptRequest[Proposal[2,"carl 4vr"]] P
L
Decision[Proposal[n,v]]
Decision[Proposal[n,v]]
L
A

Alice P A L

$v_{default}$ = "aliz rulz"

Bob P A L

Carl P A L

$v_{default}$ = "carl 4vr"

P prepare request: 1

P prepare request: 1

P prepare request: 1 A

ResponseToPrepareRequest[None] A

A

A

prepare request: **2** P

prepare request: **2** P

P prepare request: **2** A

ResponseToPrepareRequest[None] P

A ResponseToPrepareRequest[None]

A

P

ResponseToPrepareRequest[None] A

X

A

P

ResponseToPrepareRequest[None]

X

If rx'd a majority,
v = max(rx'd Proposals).v or **$v_{default}$**

P

A AcceptRequest[Proposal[2,"carl 4vr"]] P

A AcceptRequest[Proposal[2,"carl 4vr"]] P

L Decision[Proposal[n,v]]

A AcceptRequest[Proposal[2,"carl 4vr"]] P

Decision[Proposal[n,v]]

Decision[Proposal[n,v]]

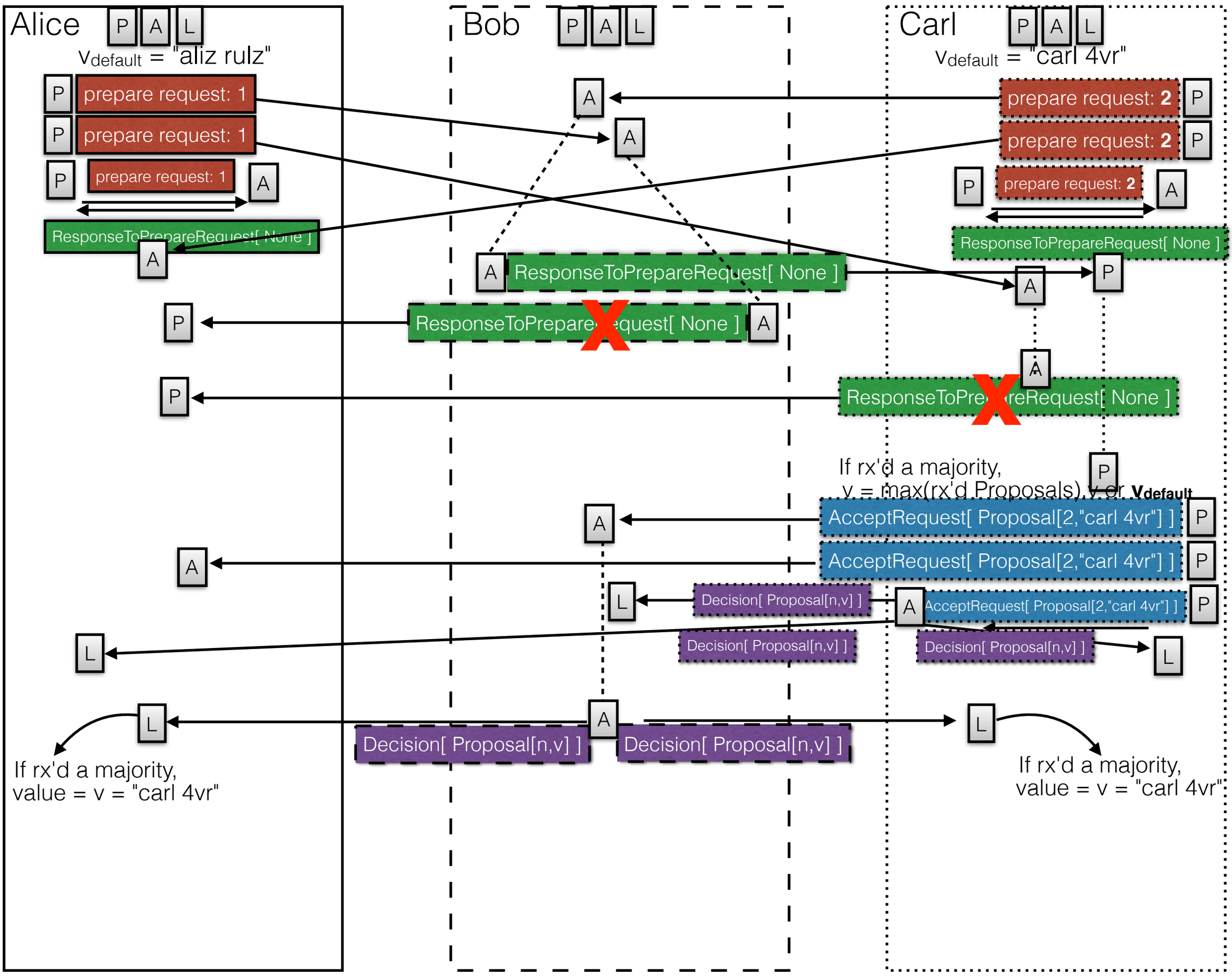
L

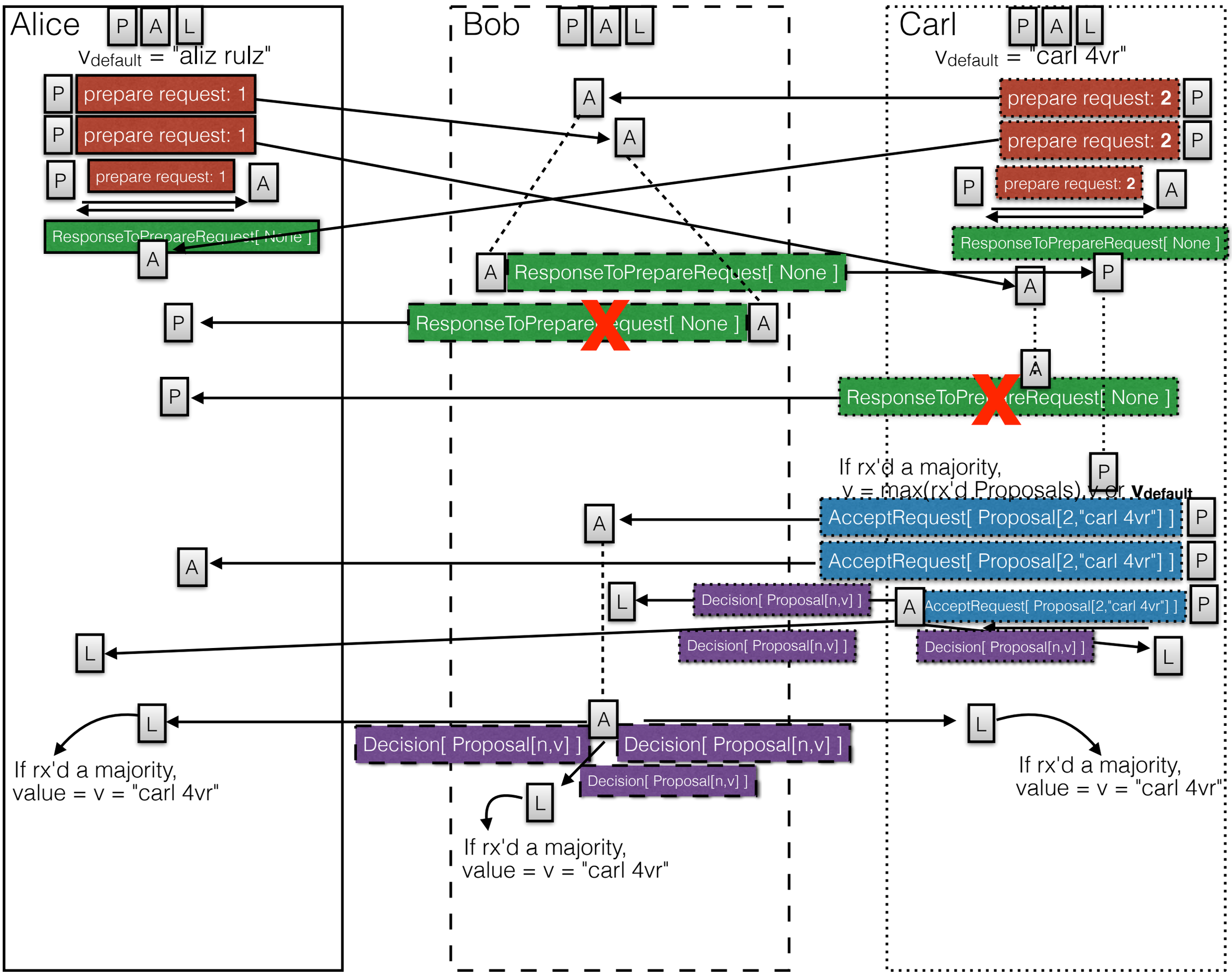
L

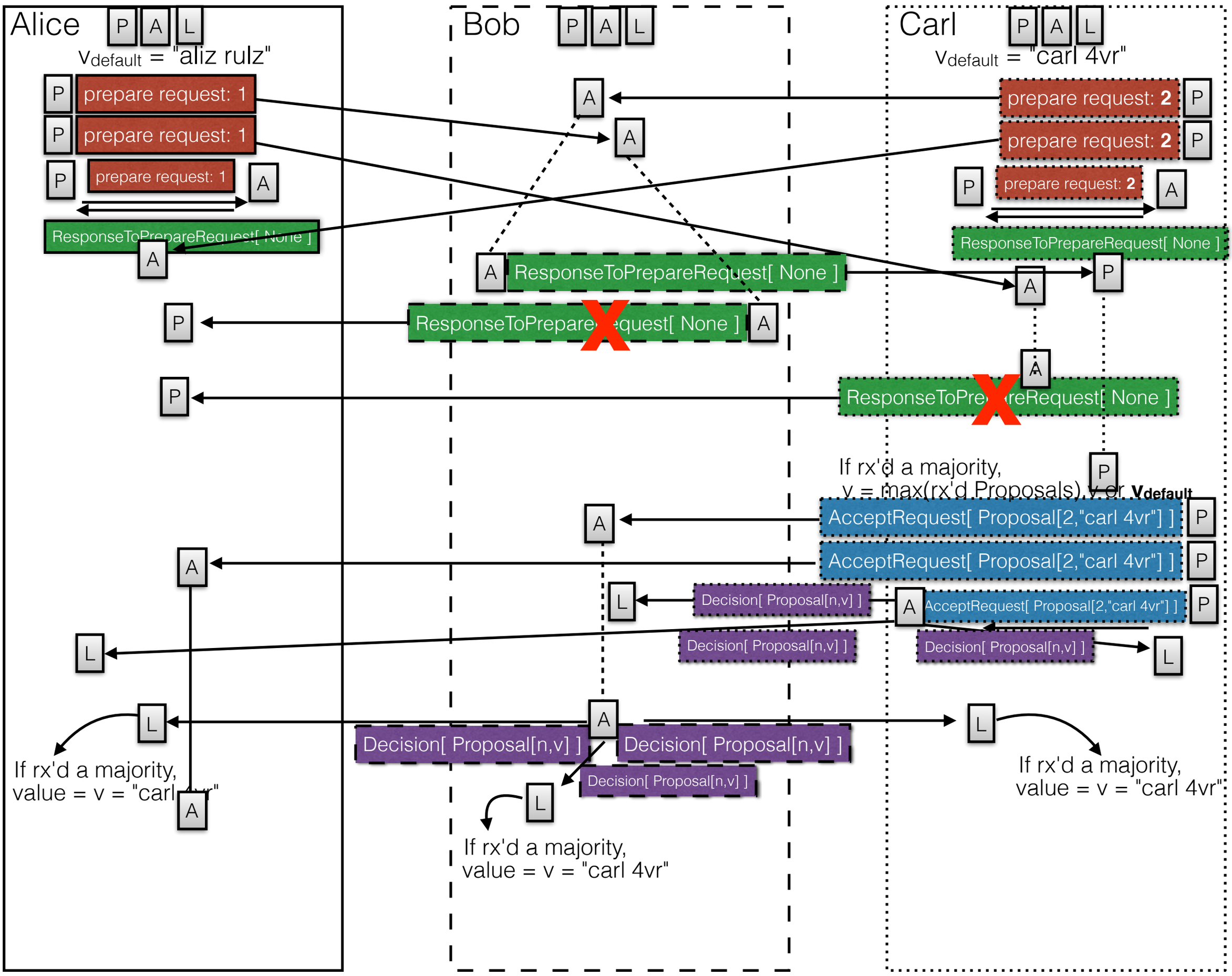
L A L

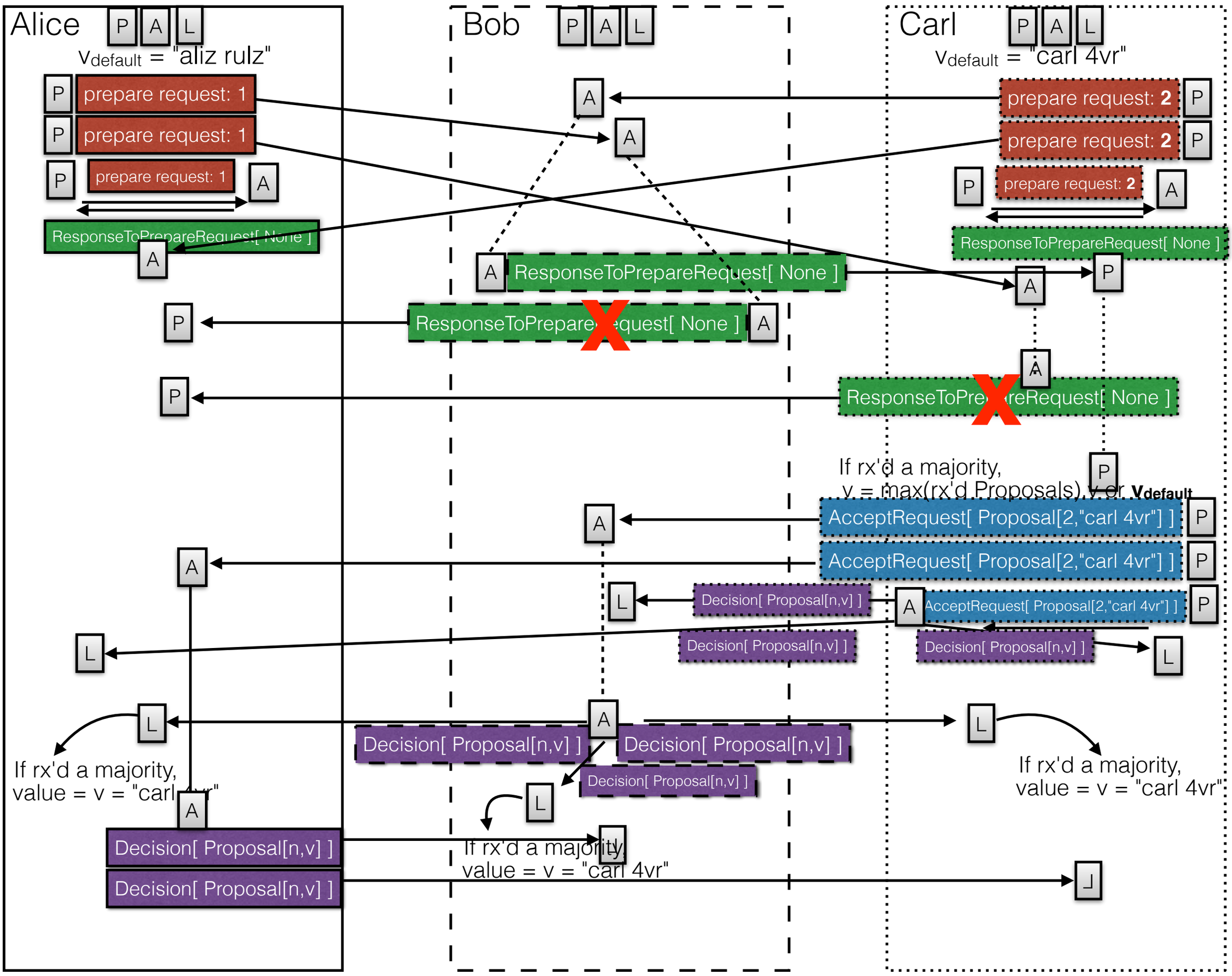
Decision[Proposal[n,v]]

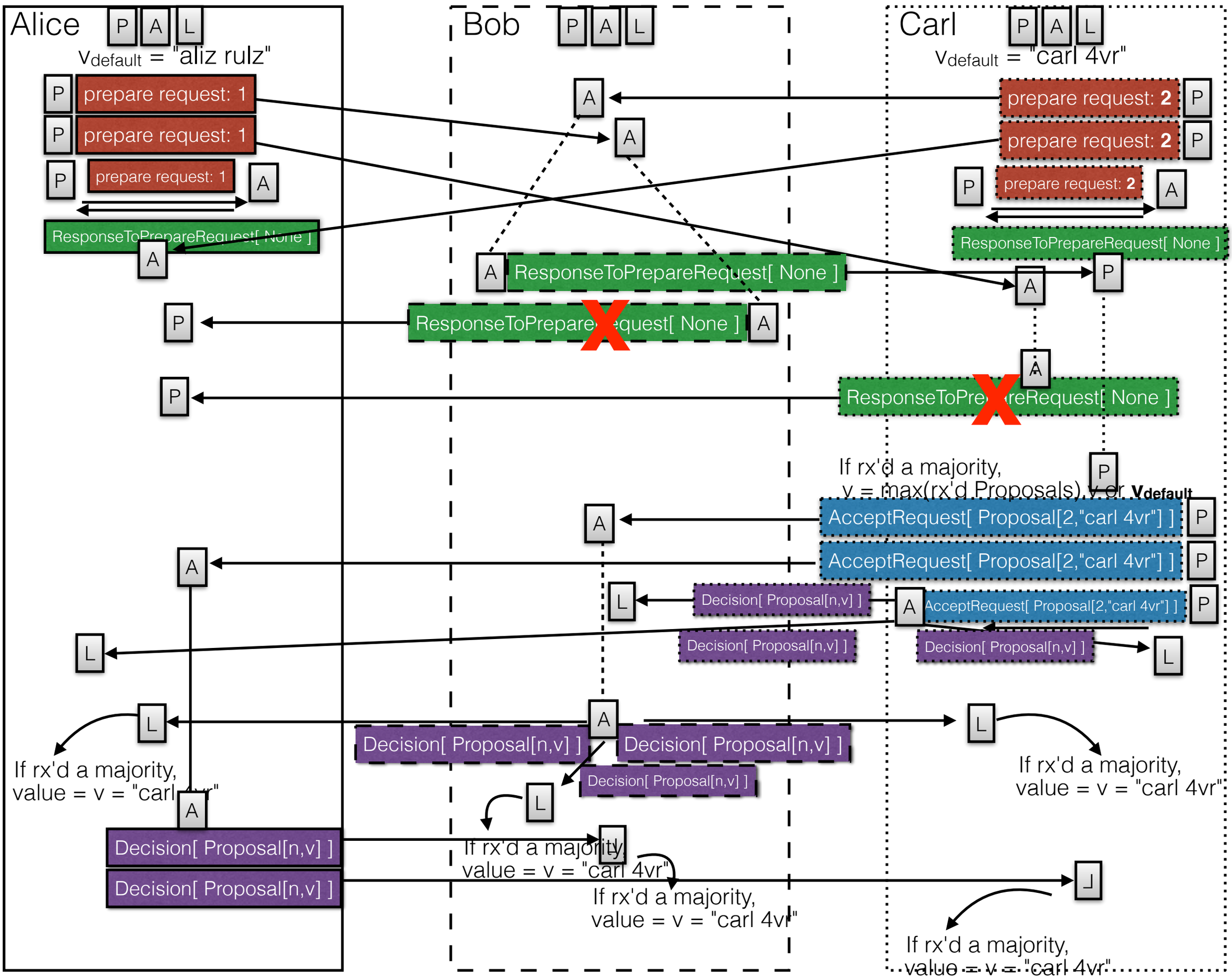
Decision[Proposal[n,v]]

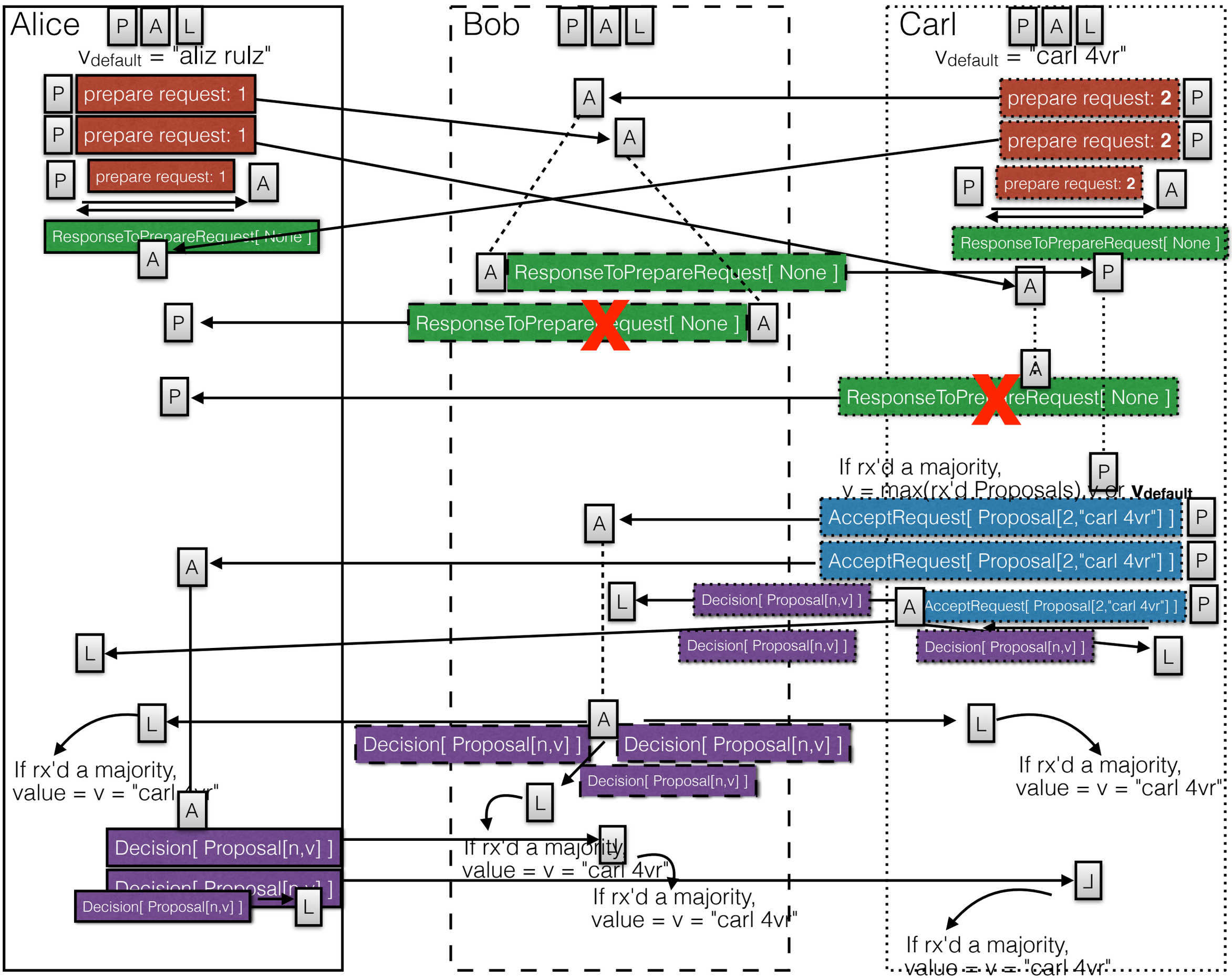

Alice
P A L
vdefault = "aliz rulz"
Bob
P A L
Carl
P A L
vdefault = "carl 4vr"
P prepare request: 1
P prepare request: 1
P prepare request: 1
A
ResponseToPrepareRequest[None]
A
prepare request: 2 P
prepare request: 2 P
P prepare request: 2 A
ResponseToPrepareRequest[None]
A ResponseToPrepareRequest[None]
P
A
P
ResponseToPrepare equest[None] A
A
P
ResponseToPre reRequest[None]
If rx'd a majority,
v = max(rx'd Proposals), v or vdefault
A
AcceptRequest[Proposal[2,"carl 4vr"]] P
A
AcceptRequest[Proposal[2,"carl 4vr"]] P
L
Decision[Proposal[n,v]]
A
AcceptRequest[Proposal[2,"carl 4vr"]] P
Decision[Proposal[n,v]]
Decision[Proposal[n,v]]
L
L
L
A
L
Decision[Proposal[n,v]]
Decision[Proposal[n,v]]
L
If rx'd a majority,
value = v = "carl 4vr"
If rx'd a majority,
value = v = "carl 4vr"

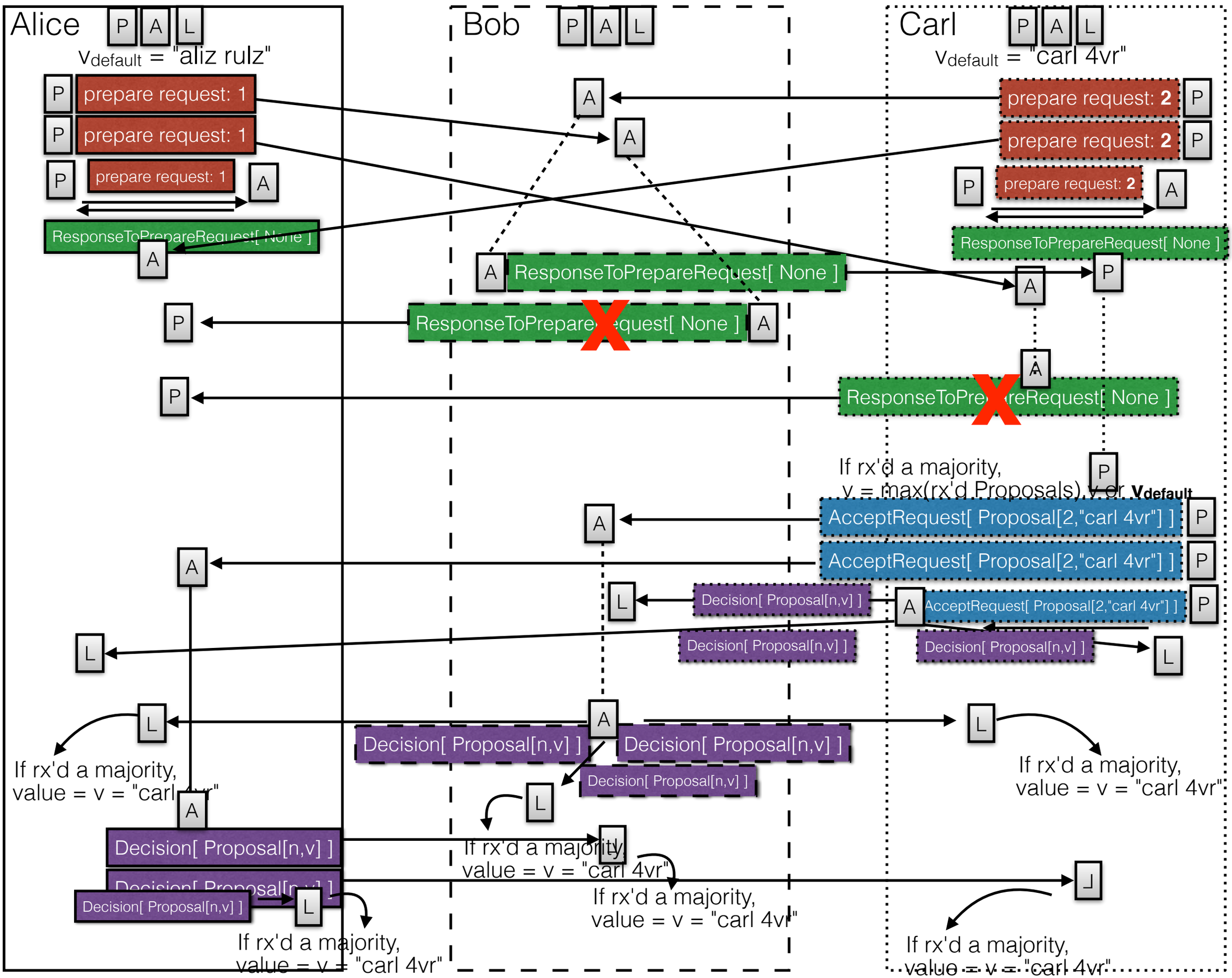

Alice
P A L
vdefault = "aliz rulz"
Bob
P A L
Carl
P A L
vdefault = "carl 4vr"
P prepare request: 1
P prepare request: 1
P prepare request: 1 A
ResponseToPrepareRequest[None]
A
prepare request: 2 P
prepare request: 2 P
P prepare request: 2 A
ResponseToPrepareRequest[None]
A ResponseToPrepareRequest[None]
P
ResponseToPrepareRequest[None] A
A
P
A
ResponseToPrepareRequest[None]
P
If rx'd a majority,
v = max(rx'd Proposals), v or vdefault
AcceptRequest[Proposal[2,"carl 4vr"]] P
AcceptRequest[Proposal[2,"carl 4vr"]] P
A
A
L
Decision[Proposal[n,v]]
A AcceptRequest[Proposal[2,"carl 4vr"]] P
Decision[Proposal[n,v]]
Decision[Proposal[n,v]]
L
L
L
A
L
Decision[Proposal[n,v]]
Decision[Proposal[n,v]]
Decision[Proposal[n,v]]
L
If rx'd a majority,
value = v = "carl 4vr"
L
If rx'd a majority,
value = v = "carl 4vr"
If rx'd a majority,
value = v = "carl 4vr"


Alice
P A L
vdefault = "aliz rulz"
Bob
P A L
Carl
P A L
vdefault = "carl 4vr"
P prepare request: 1
P prepare request: 1
P prepare request: 1
A
ResponseToPrepareRequest[None]
A
A
A
prepare request: 2 P
prepare request: 2 P
P prepare request: 2
A
ResponseToPrepareRequest[None]
A ResponseToPrepareRequest[None]
P
A
P
ResponseToPrepare equest[None]
A
X
A
P
ResponseToPre reRequest[None]
X
If rx'd a majority,
v = max(rx'd Proposals), v or vdefault
P
A
AcceptRequest[Proposal[2,"carl 4vr"]] P
A
AcceptRequest[Proposal[2,"carl 4vr"]] P
L
Decision[Proposal[n,v]]
A
AcceptRequest[Proposal[2,"carl 4vr"]] P
Decision[Proposal[n,v]]
Decision[Proposal[n,v]]
L
L
L
A
L
Decision[Proposal[n,v]]
Decision[Proposal[n,v]]
Decision[Proposal[n,v]]
If rx'd a majority,
value = v = "carl 4vr"
L
If rx'd a majority,
value = v = "carl 4vr"
A
If rx'd a majority,
value = v = "carl 4vr"


Alice
P A L
vdefault = "aliz rulz"
Bob
P A L
Carl
P A L
vdefault = "carl 4vr"
P prepare request: 1
P prepare request: 1
P prepare request: 1
A
ResponseToPrepareRequest[None]
A
prepare request: 2 P
prepare request: 2 P
P prepare request: 2 A
ResponseToPrepareRequest[None]
A ResponseToPrepareRequest[None]
P
A
P
ResponseToPrepare equest[None] A
A
P
ResponseToPre reRequest[None]
If rx'd a majority,
v = max(rx'd Proposals).v or vdefault
P
A
AcceptRequest[Proposal[2,"carl 4vr"]] P
AcceptRequest[Proposal[2,"carl 4vr"]] P
A
L
Decision[Proposal[n,v]]
A AcceptRequest[Proposal[2,"carl 4vr"]] P
L
Decision[Proposal[n,v]]
Decision[Proposal[n,v]]
L
L
A
L
Decision[Proposal[n,v]]
Decision[Proposal[n,v]]
Decision[Proposal[n,v]]
If rx'd a majority,
value = v = "carl 4vr"
L
If rx'd a majority,
value = v = "carl 4vr"
A
Decision[Proposal[n,v]]
Decision[Proposal[n,v]]
L
If rx'd a majority,
value = v = "carl 4vr"
L


Alice
P A L
vdefault = "aliz rulz"
Bob
P A L
Carl
P A L
vdefault = "carl 4vr"
P prepare request: 1
P prepare request: 1
P prepare request: 1
A
ResponseToPrepareRequest[None]
A
prepare request: 2 P
prepare request: 2 P
P prepare request: 2
A
ResponseToPrepareRequest[None]
P
A ResponseToPrepareRequest[None]
P
ResponseToPrepare equest[None] A
A
P
A
ResponseToPre reRequest[None]
P
If rx'd a majority,
v = max(rx'd Proposals), v or vdefault
AcceptRequest[Proposal[2,"carl 4vr"]] P
AcceptRequest[Proposal[2,"carl 4vr"]] P
A
A
L
Decision[Proposal[n,v]]
A AcceptRequest[Proposal[2,"carl 4vr"]] P
L
Decision[Proposal[n,v]]
Decision[Proposal[n,v]]
L
L
A
L
Decision[Proposal[n,v]]
Decision[Proposal[n,v]]
Decision[Proposal[n,v]]
If rx'd a majority,
value = v = "carl 4vr"
L
If rx'd a majority,
value = v = "carl 4vr"
A
Decision[Proposal[n,v]]
Decision[Proposal[n,v]]
L
If rx'd a majority,
value = v = "carl 4vr"
If rx'd a majority,
value = v = "carl 4vr"
If rx'd a majority,
value = v = "carl 4vr"


Alice
P A L
vdefault = "aliz rulz"
Bob
P A L
Carl
P A L
vdefault = "carl 4vr"
P prepare request: 1
P prepare request: 1
P prepare request: 1
A
ResponseToPrepareRequest[None]
A
prepare request: 2 P
prepare request: 2 P
P prepare request: 2
A
ResponseToPrepareRequest[None]
A ResponseToPrepareRequest[None]
P
A
P
ResponseToPrepare quest[None] A
A
P
ResponseToPre reRequest[None]
If rx'd a majority,
v = max(rx'd Proposals), v or vdefault
P
A
AcceptRequest[Proposal[2,"carl 4vr"]] P
A
AcceptRequest[Proposal[2,"carl 4vr"]] P
L
Decision[Proposal[n,v]]
A
AcceptRequest[Proposal[2,"carl 4vr"]] P
L
Decision[Proposal[n,v]]
Decision[Proposal[n,v]]
L
L
A
L
Decision[Proposal[n,v]]
Decision[Proposal[n,v]]
Decision[Proposal[n,v]]
If rx'd a majority,
value = v = "carl 4vr"
L
If rx'd a majority,
value = v = "carl 4vr"
A
Decision[Proposal[n,v]]
Decision[Proposal[n,v]]
Decision[Proposal[n,v]]
L
If rx'd a majority,
value = v = "carl 4vr"
If rx'd a majority,
value = v = "carl 4vr"
If rx'd a majority,
value = v = "carl 4vr"


Alice
P A L
vdefault = "aliz rulz"
Bob
P A L
Carl
P A L
vdefault = "carl 4vr"
P prepare request: 1
P prepare request: 1
P prepare request: 1
A
ResponseToPrepareRequest[None]
A
prepare request: 2 P
prepare request: 2 P
P prepare request: 2
A
ResponseToPrepareRequest[None]
A ResponseToPrepareRequest[None]
P
A
P
ResponseToPrepare equest[None] A
X
A
P
ResponseToPre reRequest[None]
X
If rx'd a majority,
v = max(rx'd Proposals), v or vdefault
AcceptRequest[Proposal[2,"carl 4vr"]] P
AcceptRequest[Proposal[2,"carl 4vr"]] P
L
Decision[Proposal[n,v]]
A AcceptRequest[Proposal[2,"carl 4vr"]] P
Decision[Proposal[n,v]]
Decision[Proposal[n,v]]
L
L
A
L
Decision[Proposal[n,v]]
Decision[Proposal[n,v]]
Decision[Proposal[n,v]]
L
L
If rx'd a majority,
value = v = "carl 4vr"
If rx'd a majority,
value = v = "carl 4vr"
A
Decision[Proposal[n,v]]
Decision[Proposal[n,v]]
Decision[Proposal[n,v]]
L
If rx'd a majority,
value = v = "carl 4vr"
If rx'd a majority,
value = v = "carl 4vr"
If rx'd a majority,
value = v = "carl 4vr"

Outline

- The Algorithm
- Example: how it works **initially**
- Example: how it handles **conflicts**

- Example: how it works **after consensus**

Alice

Bob

Carl

Alice P A L

Bob P A L

Carl P A L

Alice P A L

Proposal[2,"carl 4vr"]

Bob P A L

Proposal[2,"carl 4vr"]

Carl P A L

Proposal[2,"carl 4vr"]

Alice P A L

$v_{default}$ = "aliz rulz"

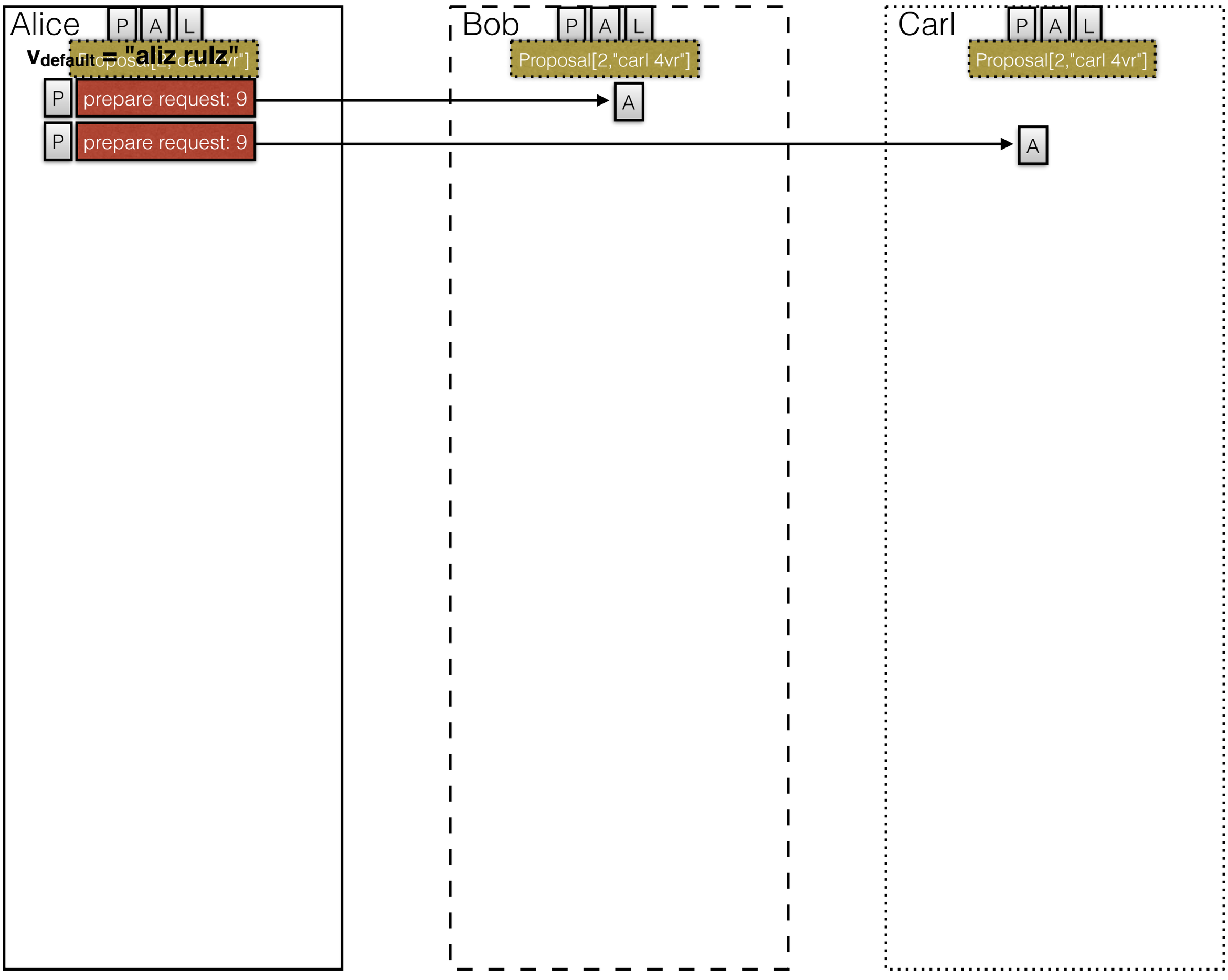
Proposal[2,"carl 4vr"]

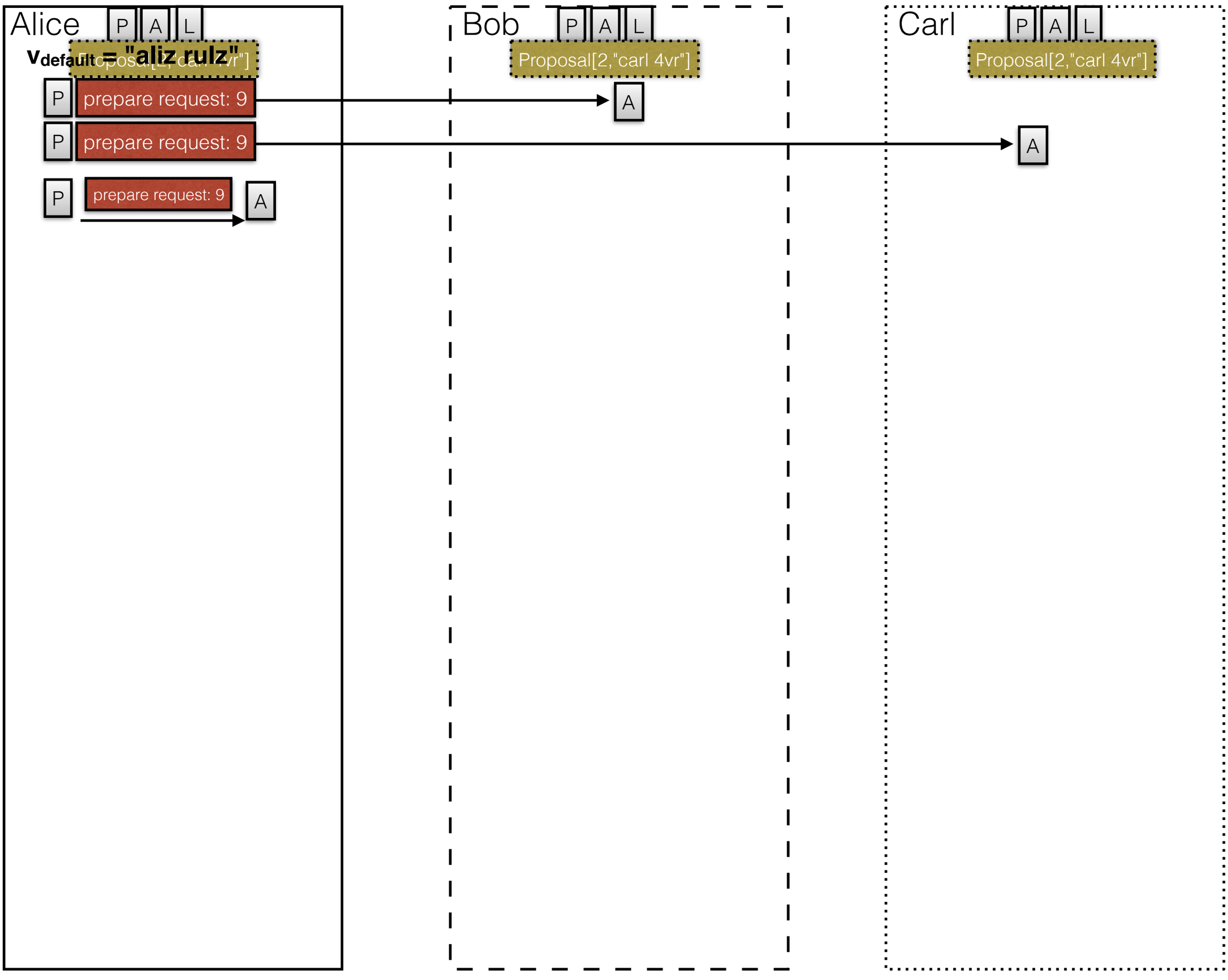
Bob P A L

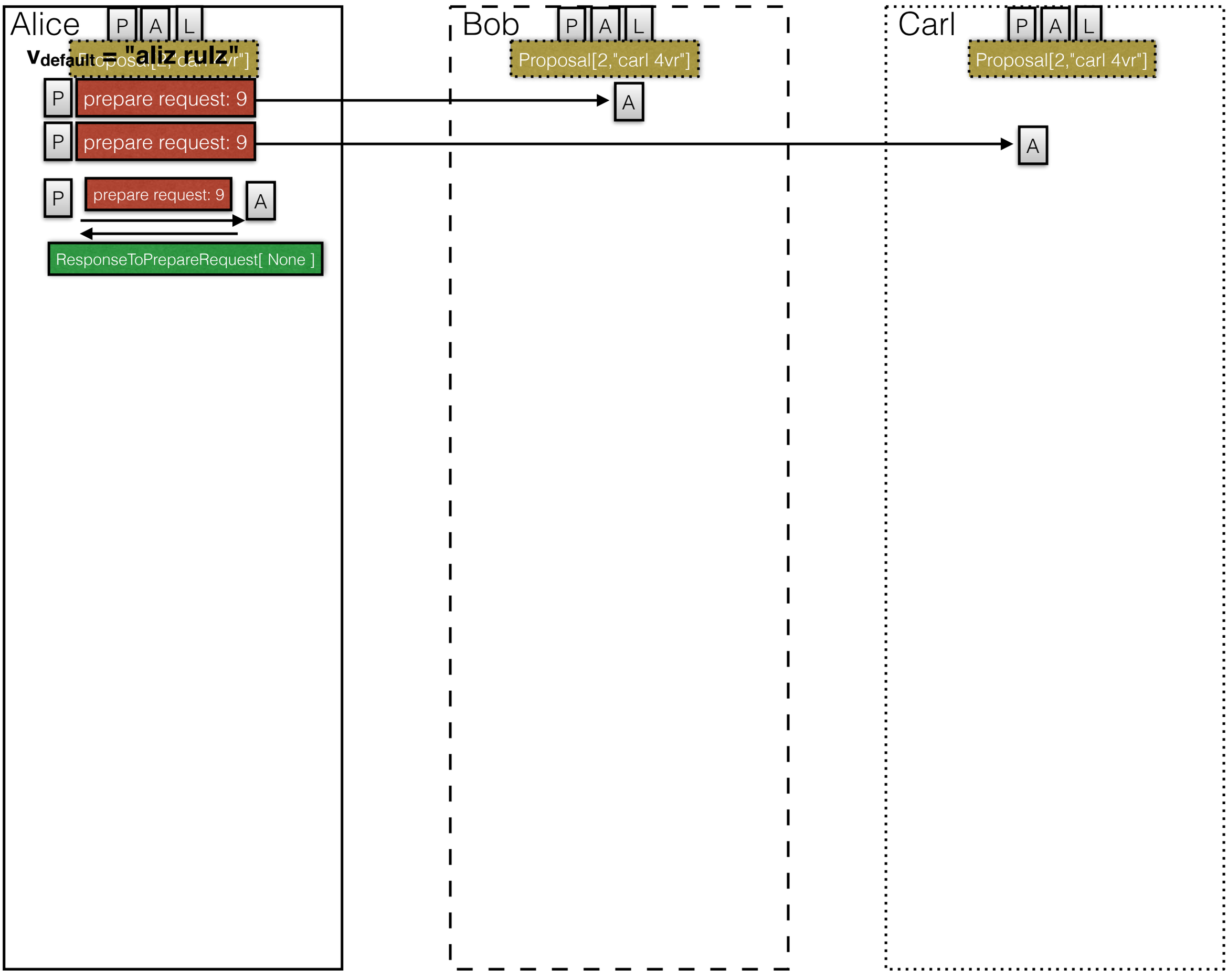
Proposal[2,"carl 4vr"]

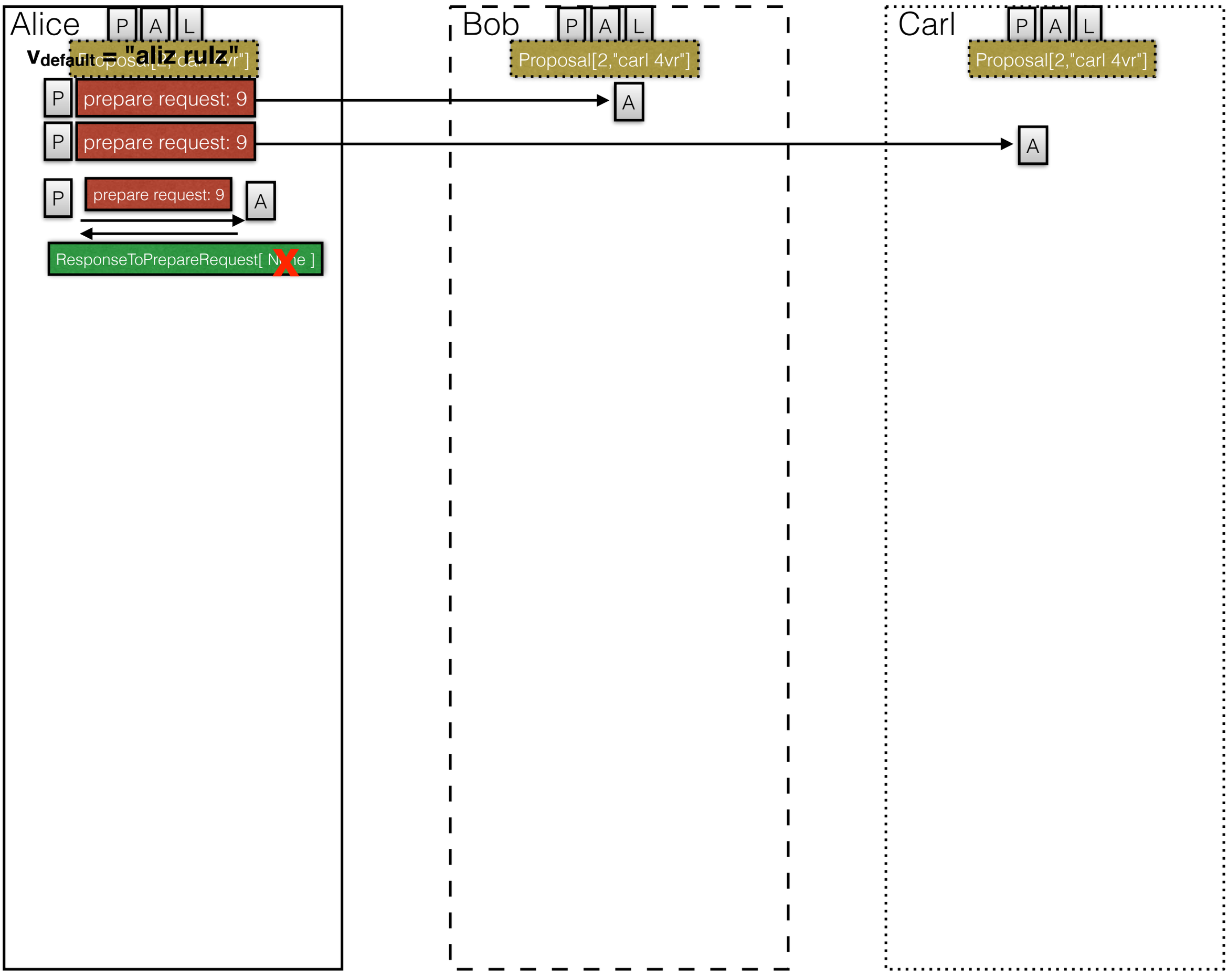
Carl P A L

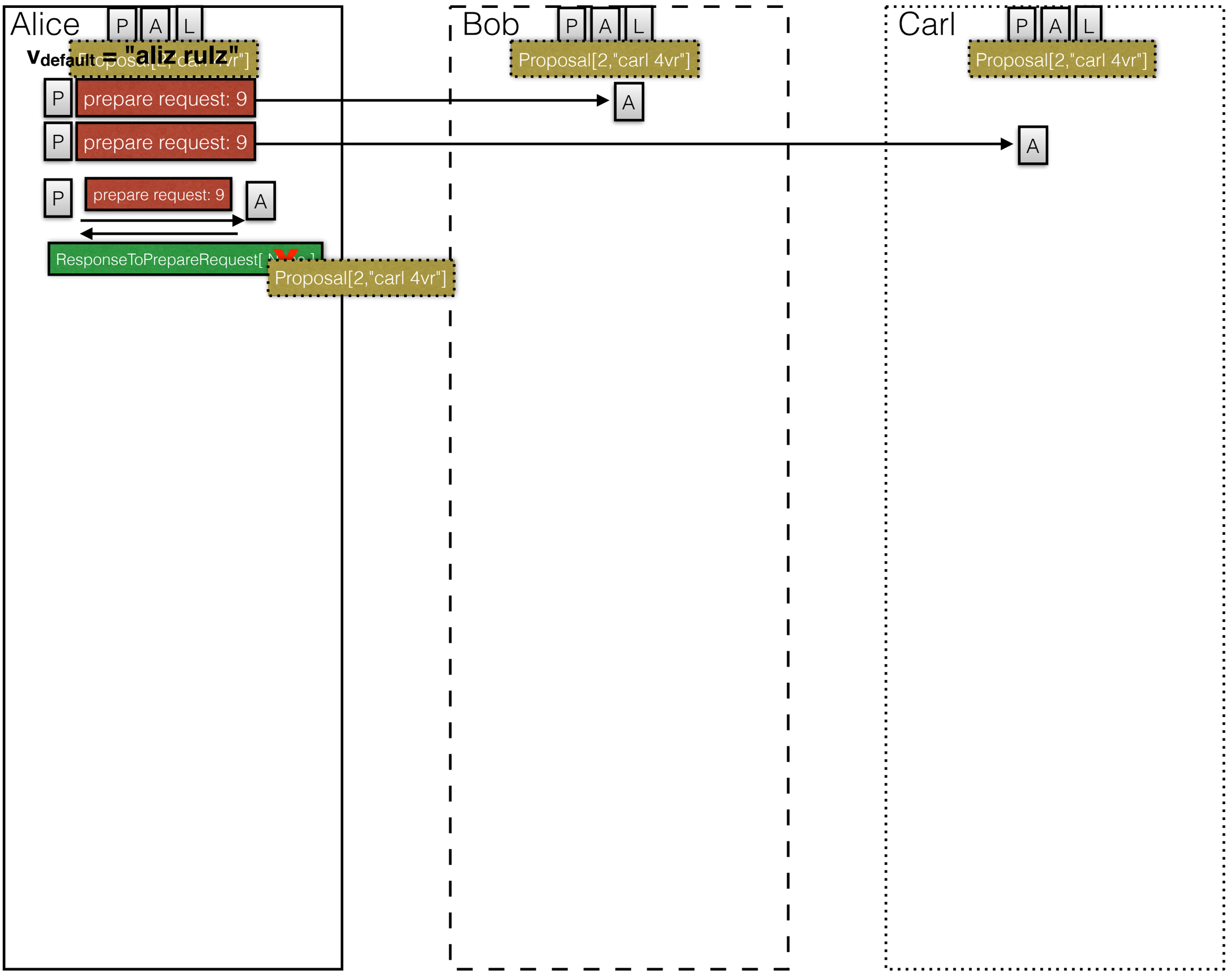
Proposal[2,"carl 4vr"]

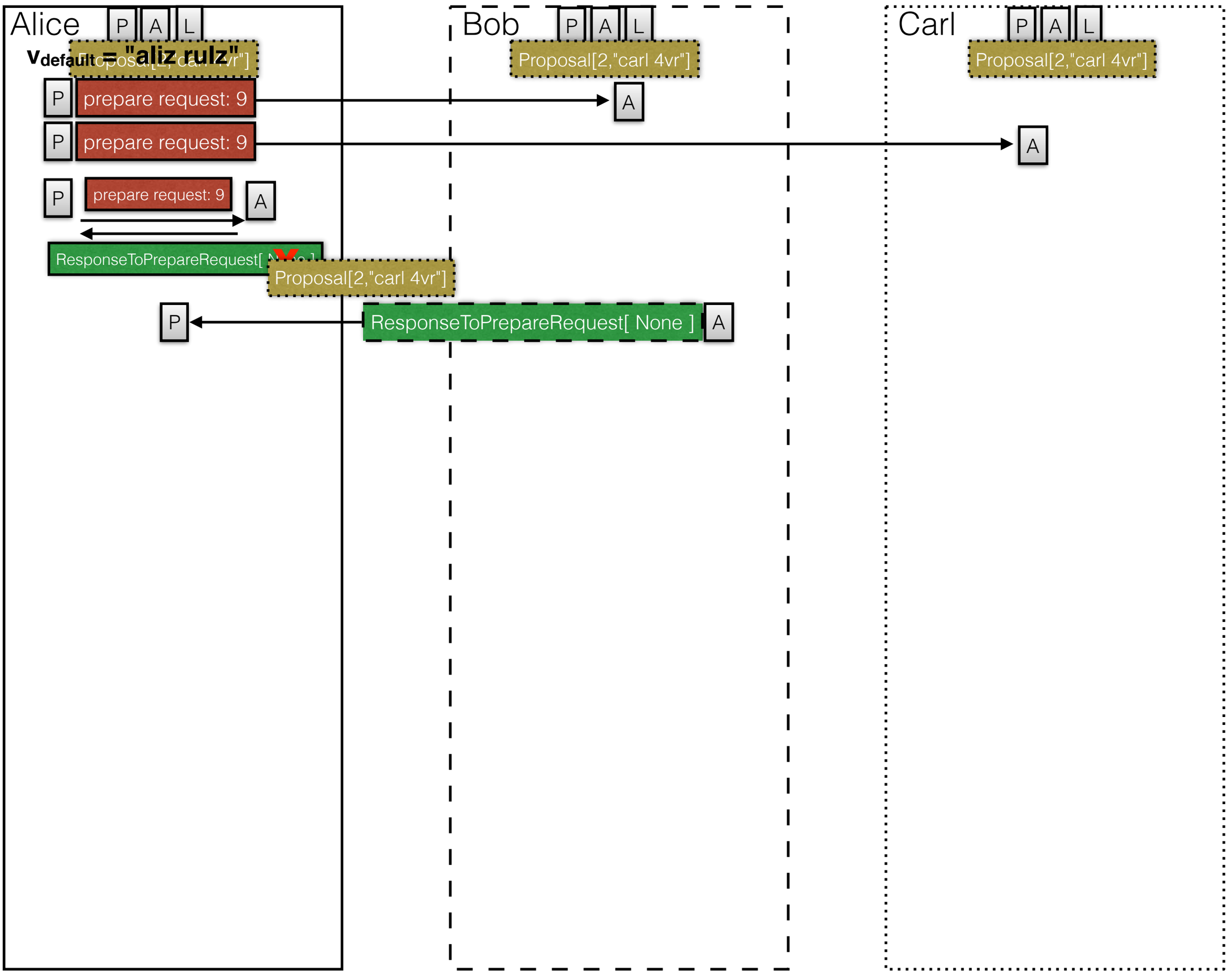

Alice
P
A
L
Vdefault = "aliz rulz"
Proposal[2,"carl 4vr"]
P
prepare request: 9
P
prepare request: 9
Bob
P
A
L
Proposal[2,"carl 4vr"]
A
Carl
P
A
L
Proposal[2,"carl 4vr"]
A

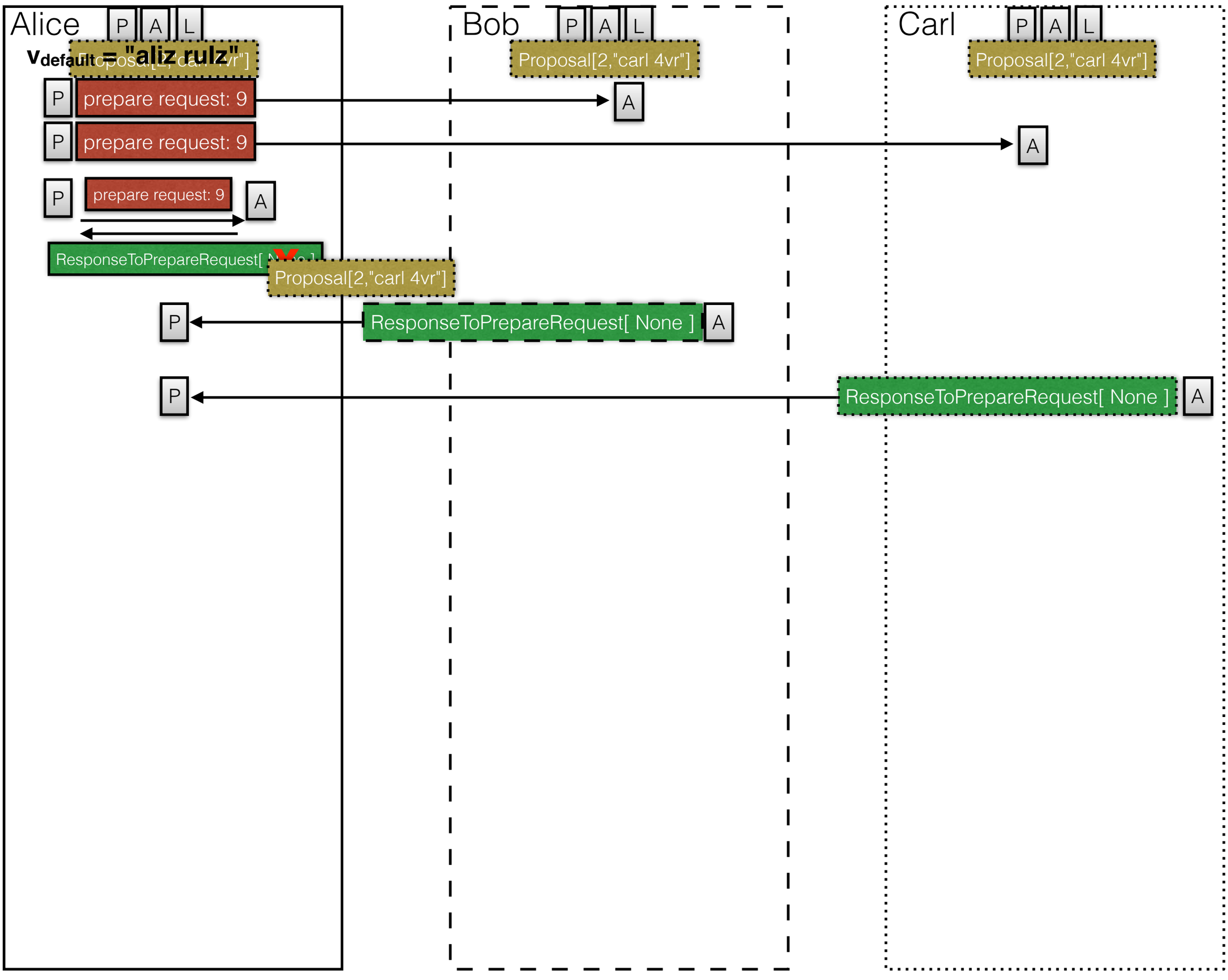

Alice
P
A
L
Vdefault = "aliz rulz"
Proposal[2,"carl 4vr"]
P
prepare request: 9
P
prepare request: 9
P
prepare request: 9
A
Bob
P
A
L
Proposal[2,"carl 4vr"]
A
Carl
P
A
L
Proposal[2,"carl 4vr"]
A

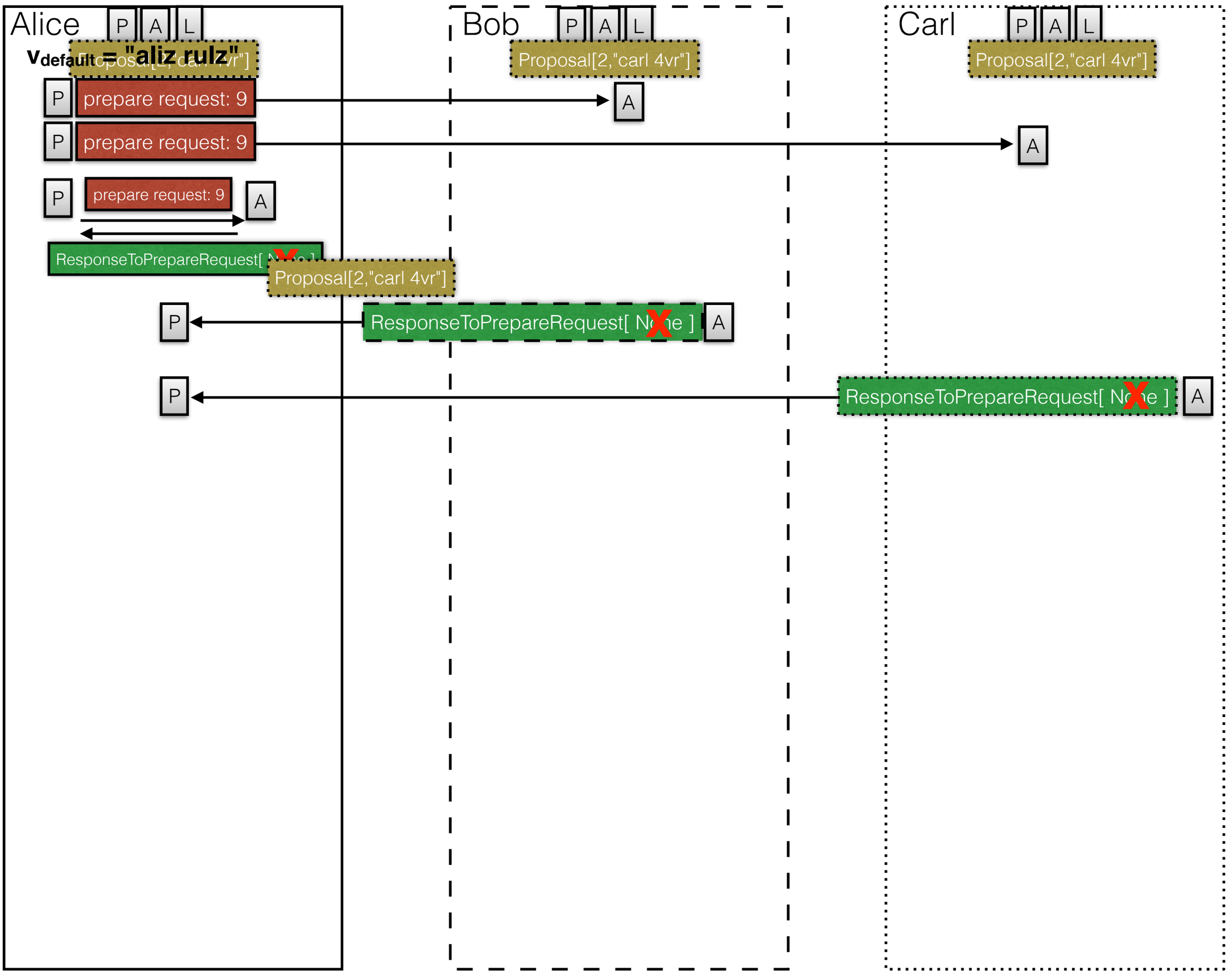

Alice
P
A
L
Vdefault = "aliz rulz"
Proposal[2,"carl 4vr"]
P
prepare request: 9
P
prepare request: 9
P
prepare request: 9
A
ResponseToPrepareRequest[None]
Bob
P
A
L
Proposal[2,"carl 4vr"]
A
Carl
P
A
L
Proposal[2,"carl 4vr"]
A

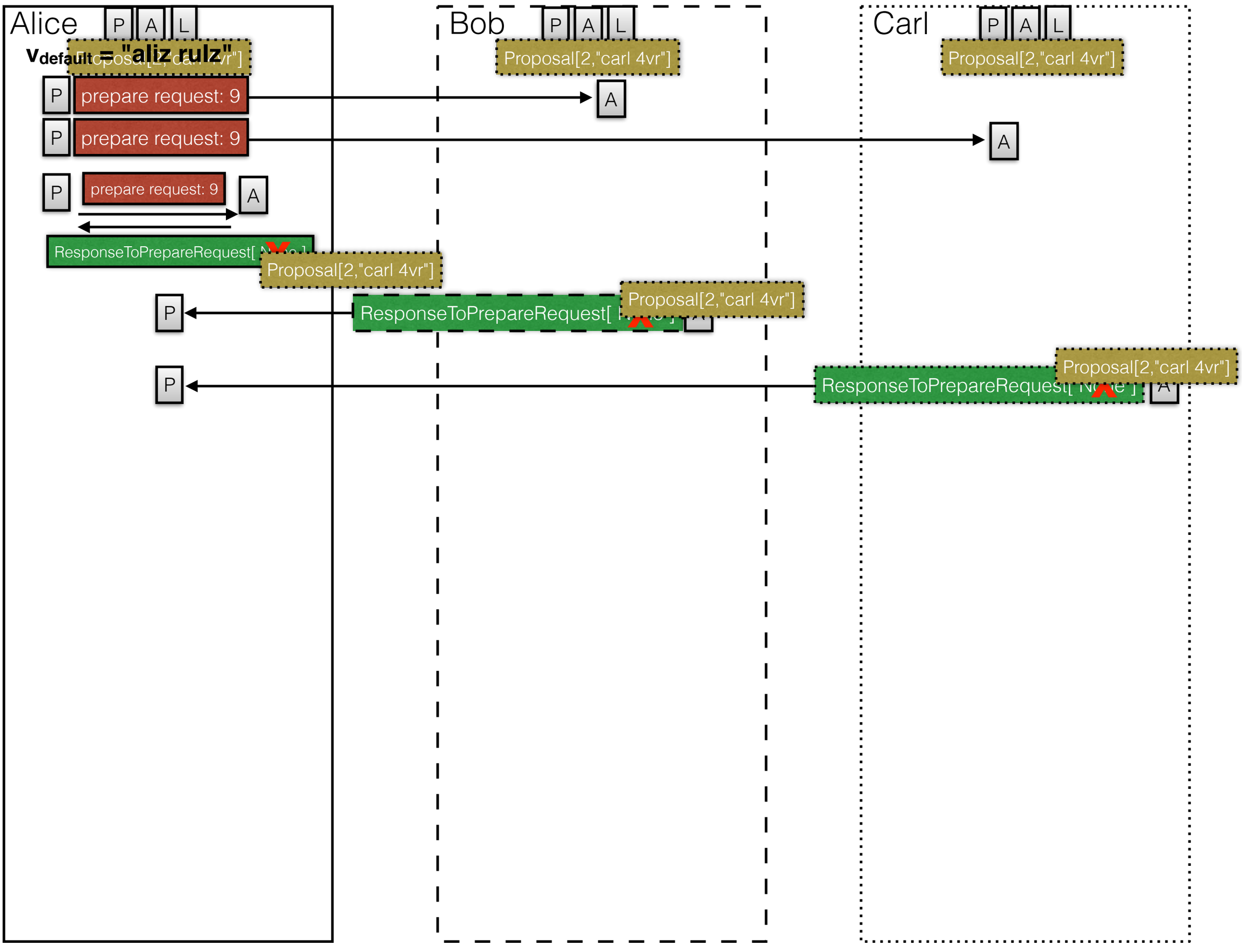

Alice
P
A
L
Vdefault = "aliz rulz"
Proposal[2,"carl 4vr"]
P
prepare request: 9
P
prepare request: 9
P
prepare request: 9
A
ResponseToPrepareRequest[None]
Bob
P
A
L
Proposal[2,"carl 4vr"]
A
Carl
P
A
L
Proposal[2,"carl 4vr"]
A

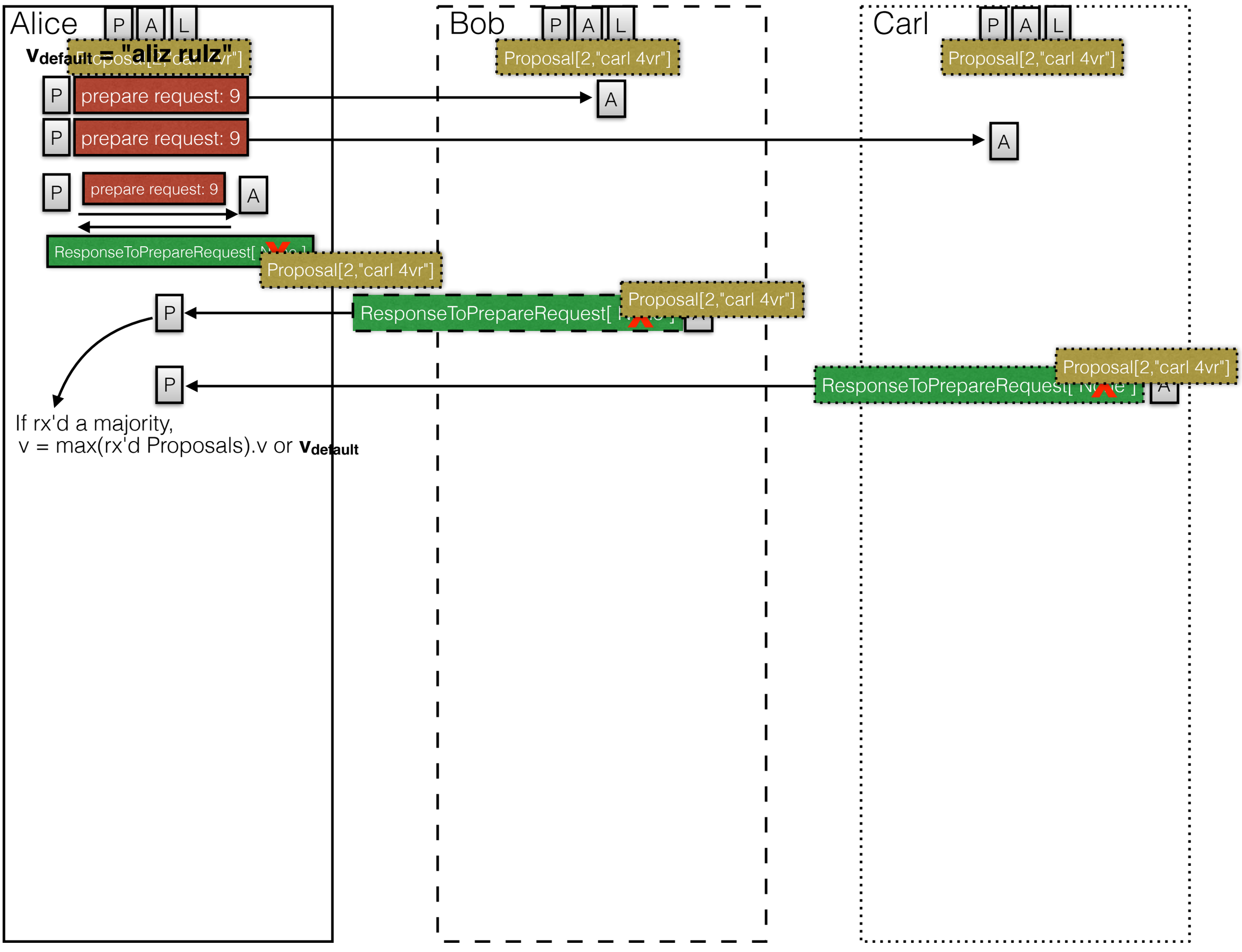

Alice
P
A
L
vdefault = "aliz rulz"
Proposal[2,"carl 4vr"]
P
prepare request: 9
P
prepare request: 9
P
prepare request: 9
A
ResponseToPrepareRequest[
Proposal[2,"carl 4vr"]
Bob
P
A
L
Proposal[2,"carl 4vr"]
A
Carl
P
A
L
Proposal[2,"carl 4vr"]
A

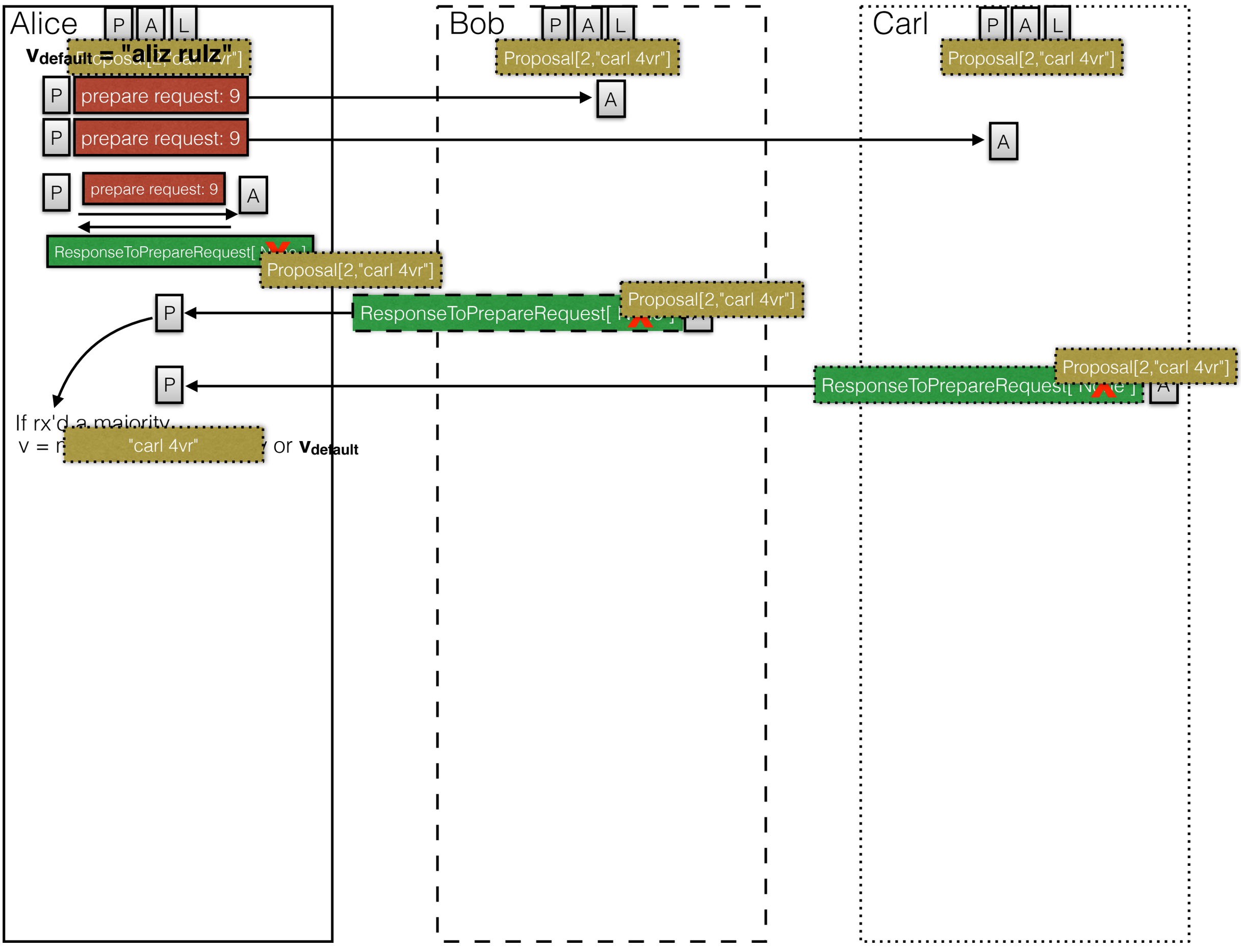

Alice
P
A
L
vdefault = "aliz rulz"
Proposal[2,"carl 4vr"]
P
prepare request: 9
P
prepare request: 9
P
prepare request: 9
A
ResponseToPrepareRequest[
Proposal[2,"carl 4vr"]
P
Bob
P
A
L
Proposal[2,"carl 4vr"]
A
ResponseToPrepareRequest[None]
A
Carl
P
A
L
Proposal[2,"carl 4vr"]
A


Alice
P
A
L
Vdefault = "aliz rulz"
Proposal[2,"carl 4vr"]
P
prepare request: 9
P
prepare request: 9
P
prepare request: 9
A
ResponseToPrepareRequest[
Proposal[2,"carl 4vr"]
P
Bob
P
A
L
Proposal[2,"carl 4vr"]
A
A
ResponseToPrepareRequest[None]
A
P
Carl
P
A
L
Proposal[2,"carl 4vr"]
A
ResponseToPrepareRequest[None]
A
P


Alice
P
A
L
Vdefault = "aliz rulz"
Proposal[2,"carl 4vr"]
P
prepare request: 9
P
prepare request: 9
P
prepare request: 9
A
ResponseToPrepareRequest[None]
Proposal[2,"carl 4vr"]
P
Bob
P
A
L
Proposal[2,"carl 4vr"]
A
ResponseToPrepareRequest[None]
A
P
Carl
P
A
L
Proposal[2,"carl 4vr"]
A
ResponseToPrepareRequest[None]
A
P


Alice
P
A
L
Vdefault = "aliz rulz"
Proposal[2,"carl 4vr"]
P
prepare request: 9
P
prepare request: 9
P
prepare request: 9
A
ResponseToPrepareRequest[
Proposal[2,"carl 4vr"]
P
Proposal[2,"carl 4vr"]
P
Bob
P
A
L
Proposal[2,"carl 4vr"]
A
Proposal[2,"carl 4vr"]
Carl
P
A
L
Proposal[2,"carl 4vr"]
A
Proposal[2,"carl 4vr"]
A


Alice
P
A
L
Bob
P
A
L
Carl
P
A
L
vdefault = "aliz rulz"
Proposal[2,"carl 4vr"]
Proposal[2,"carl 4vr"]
Proposal[2,"carl 4vr"]
P
prepare request: 9
A
P
prepare request: 9
A
P
prepare request: 9
A
Proposal[2,"carl 4vr"]
Proposal[2,"carl 4vr"]
P
Proposal[2,"carl 4vr"]
P
If rx'd a majority,
v = max(rx'd Proposals).v or vdefault


Alice
P
A
L
Vdefault = "aliz rulz"
Proposal[2,"carl 4vr"]
P
prepare request: 9
P
prepare request: 9
P
prepare request: 9
A
ResponseToPrepareRequest[
Proposal[2,"carl 4vr"]
P
P
If rx'd a majority
v = r
"carl 4vr"
or Vdefault
Bob
P
A
L
Proposal[2,"carl 4vr"]
A
Proposal[2,"carl 4vr"]
ResponseToPrepareRequest[
Carl
P
A
L
Proposal[2,"carl 4vr"]
A
Proposal[2,"carl 4vr"]
ResponseToPrepareRequest[
A

Alice P A L

Bob P A L

Carl P A L

$v_{default}$ = "aliz rulz"

Proposal[2,"carl 4vr"]

P prepare request: 9 → A

P prepare request: 9 → A

P prepare request: 9 → A

ResponseToPrepareRequest[None]

Proposal[2,"carl 4vr"]

P ← ResponseToPrepareRequest[None]

Proposal[2,"carl 4vr"]

P ← ResponseToPrepareRequest[None] → A

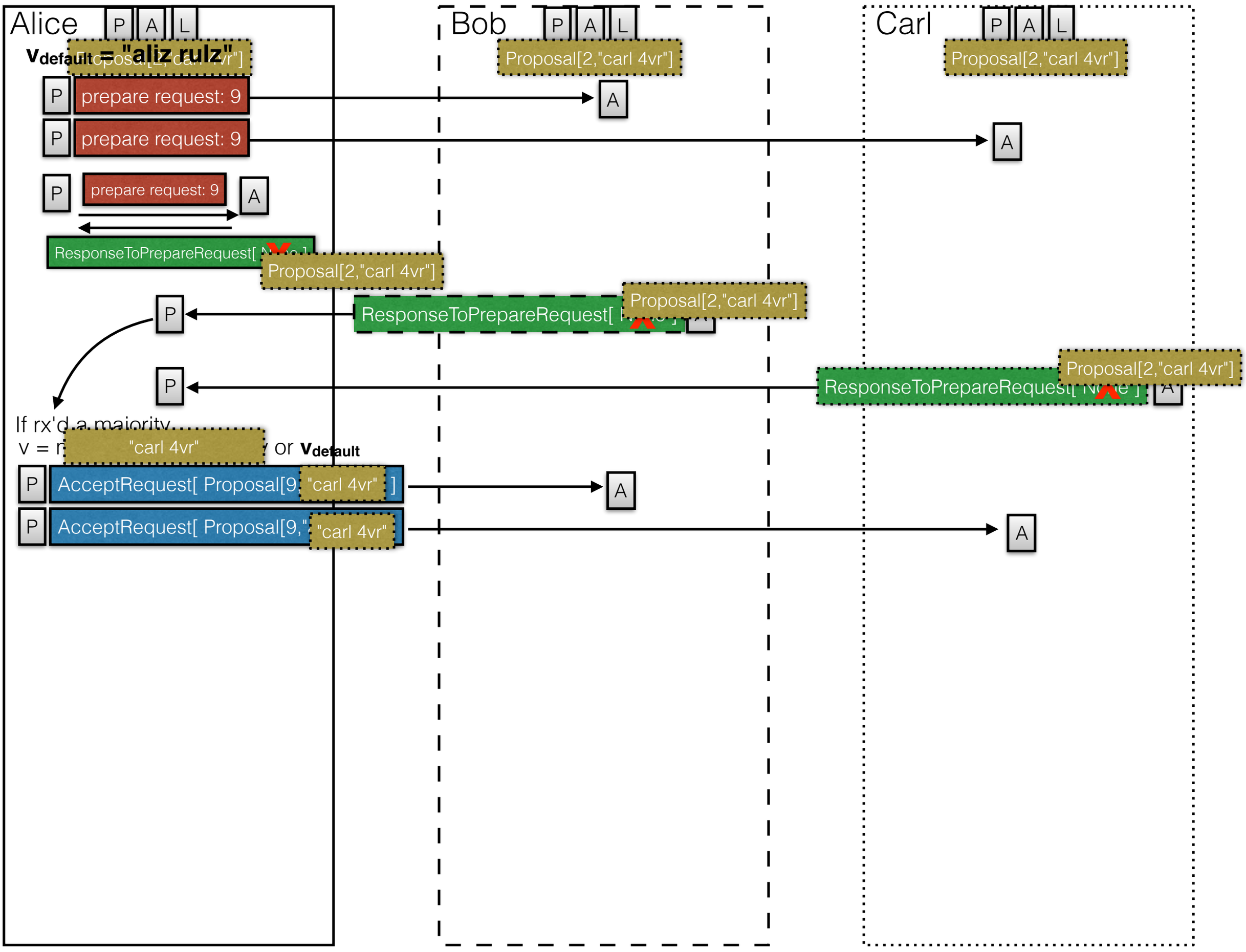
Proposal[2,"carl 4vr"]

If rx'd a majority

v = n… "carl 4vr" … or $v_{default}$

P AcceptRequest[Proposal[9,"aliz rulz"]] → A

P AcceptRequest[Proposal[9,"aliz rulz"]] → A


Alice
P
A
L
Bob
P
A
L
Carl
P
A
L
Vdefault = "aliz rulz"
Proposal[2,"carl 4vr"]
Proposal[2,"carl 4vr"]
Proposal[2,"carl 4vr"]
P
prepare request: 9
A
P
prepare request: 9
A
P
prepare request: 9
A
ResponseToPrepareRequest[
Proposal[2,"carl 4vr"]
P
ResponseToPrepareRequest[
Proposal[2,"carl 4vr"]
P
ResponseToPrepareRequest[
Proposal[2,"carl 4vr"]
A
If rx'd a majority
v = r
"carl 4vr"
or Vdefault
P
AcceptRequest[Proposal[9,
"carl 4vr"
]
A
P
AcceptRequest[Proposal[9,"
"carl 4vr"
A

Alice P A L

Bob P A L

Carl P A L

$v_{default}$ = "aliz rulz"

Proposal[2,"carl 4vr"]

P prepare request: 9 → A

P prepare request: 9 → A

P prepare request: 9 → A

ResponseToPrepareRequest[N...]

Proposal[2,"carl 4vr"]

P ← ResponseToPrepareRequest[...]

Proposal[2,"carl 4vr"]

P ← ResponseToPrepareRequest[None] A

Proposal[2,"carl 4vr"]

If rx'd a majority

v = ... "carl 4vr" ... or $v_{default}$

P AcceptRequest[Proposal[9, "carl 4vr"]] → A

P AcceptRequest[Proposal[9, "carl 4vr"]] → A

P AcceptRequest[Proposal[9,"aliz rulz"]] A

Alice P A L

Bob P A L

Carl P A L

$v_{default}$ = "aliz rulz"

Proposal[2,"carl 4vr"]

P prepare request: 9 → A

P prepare request: 9 → A

P prepare request: 9 → A

ResponseToPrepareRequest[N…]

Proposal[2,"carl 4vr"]

P ← ResponseToPrepareRequest[…]

Proposal[2,"carl 4vr"]

P ← ResponseToPrepareRequest[None] A

Proposal[2,"carl 4vr"]

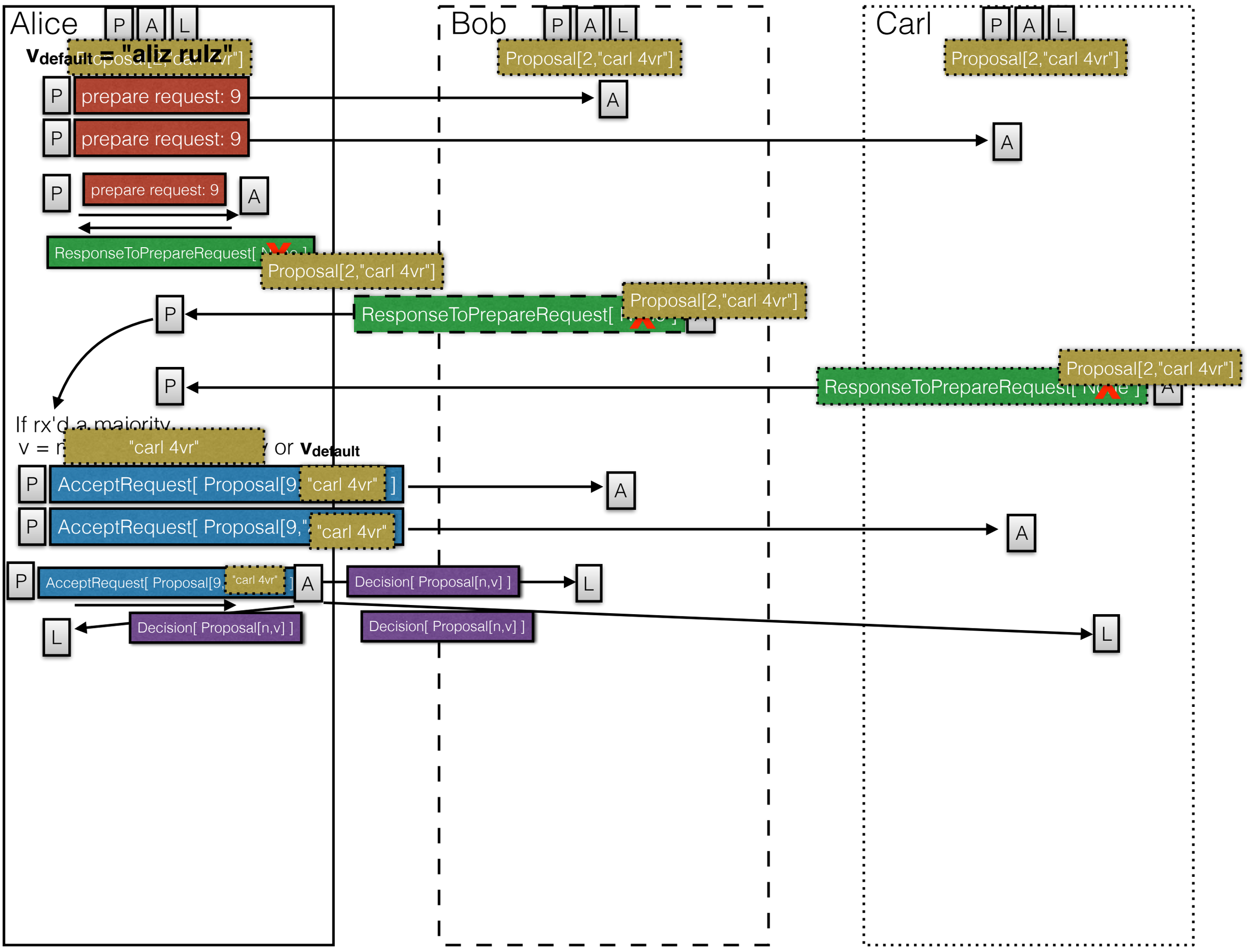
If rx'd a majority

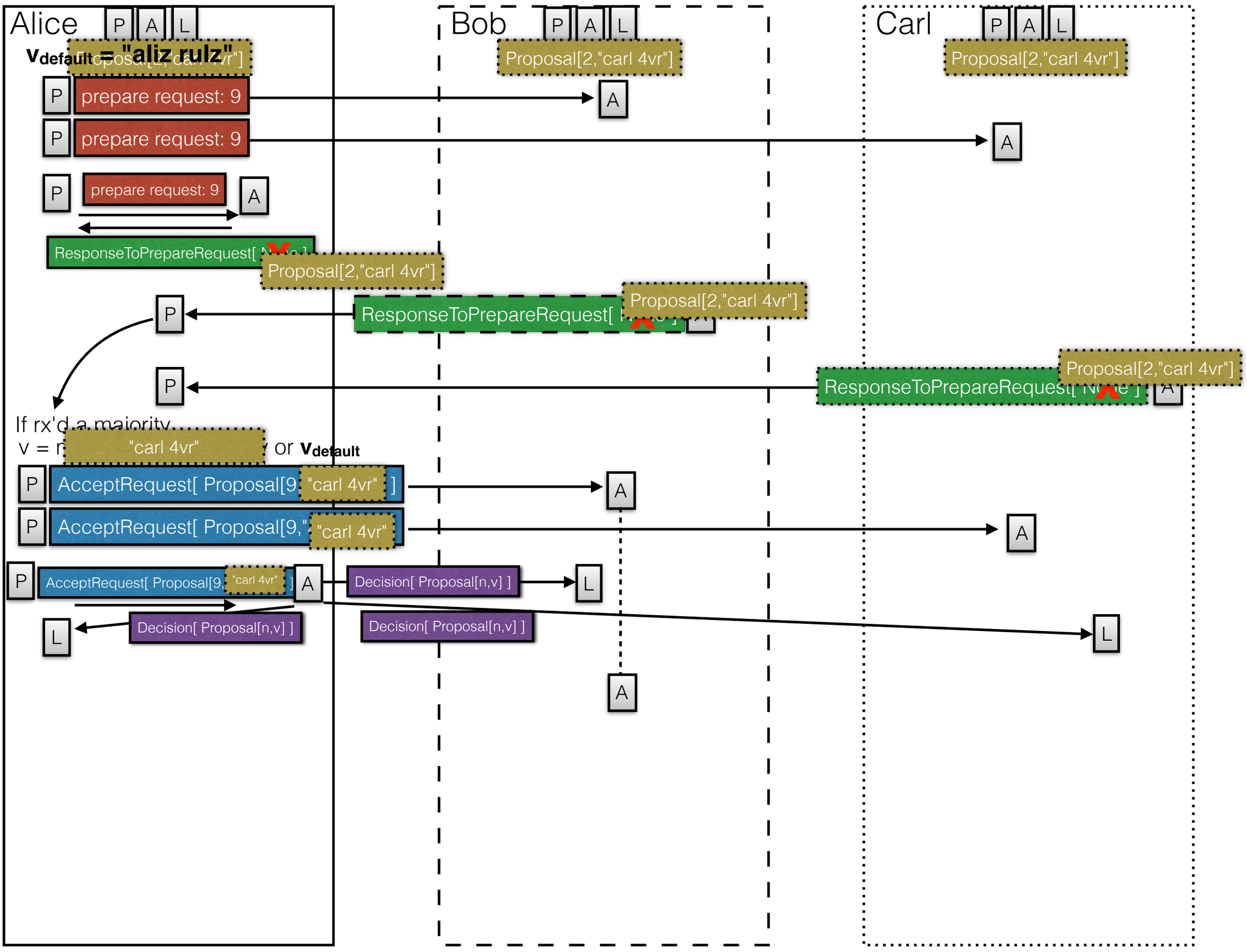
v = r… "carl 4vr" or $v_{default}$

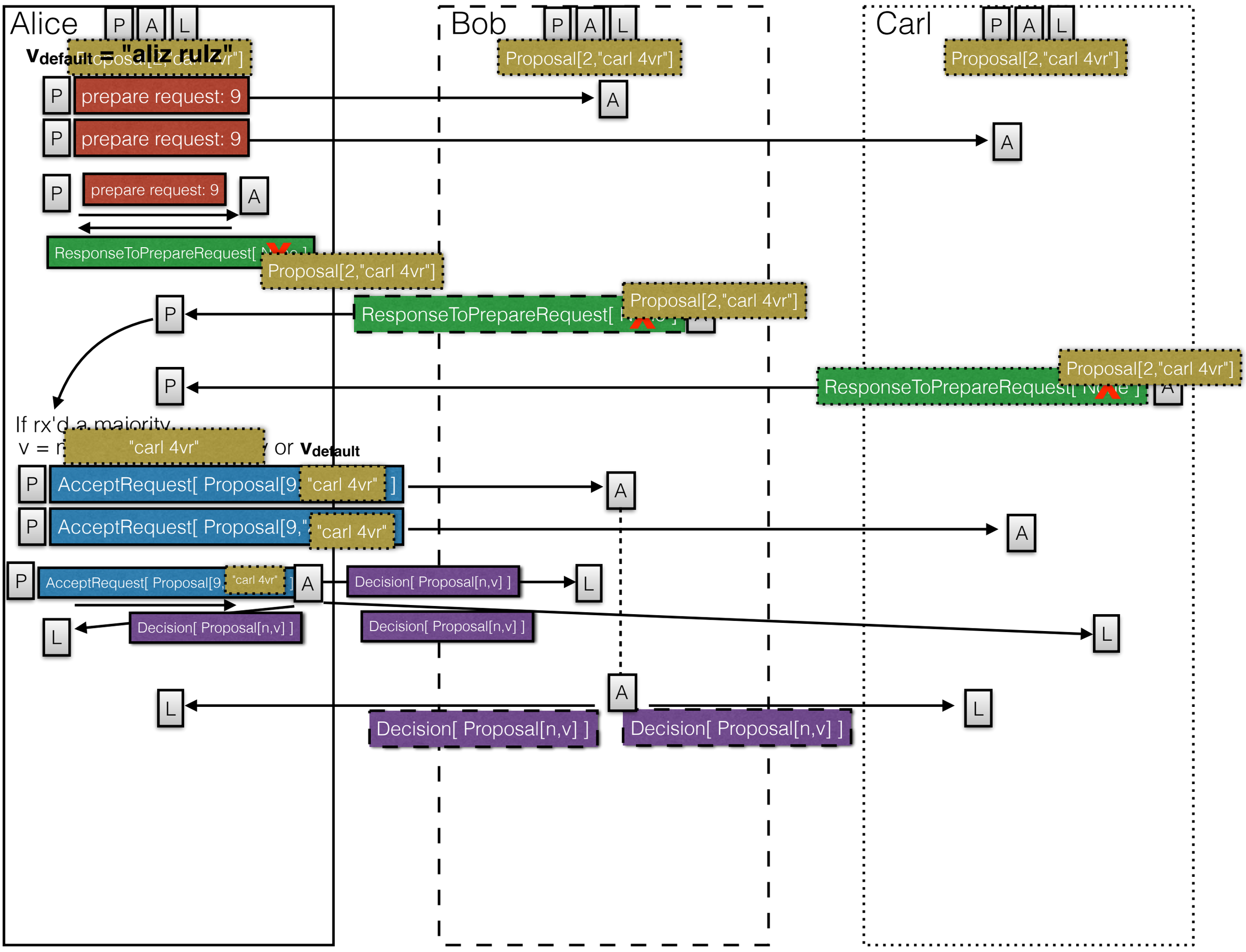
P AcceptRequest[Proposal[9, "carl 4vr"] → A

P AcceptRequest[Proposal[9, "carl 4vr"] → A

P AcceptRequest[Proposal[9, "carl 4vr"] A


Alice
P
A
L
Bob
Carl
Vdefault = "aliz rulz"
Proposal[2,"carl 4vr"]
prepare request: 9
ResponseToPrepareRequest[
Proposal[2,"carl 4vr"]
If rx'd a majority
v = n
or Vdefault
"carl 4vr"
AcceptRequest[Proposal[9,
]
Decision[Proposal[n,v]]


Alice
P
A
L
Bob
Carl
Vdefault = "aliz rulz"
Proposal[2,"carl 4vr"]
prepare request: 9
ResponseToPrepareRequest[
Proposal[2,"carl 4vr"]
If rx'd a majority
v = n
"carl 4vr"
or Vdefault
AcceptRequest[Proposal[9,
"carl 4vr"
Decision[Proposal[n,v]]


Alice
P
A
L
Bob
Carl
Vdefault = "aliz rulz"
Proposal[2,"carl 4vr"]
prepare request: 9
ResponseToPrepareRequest[
Proposal[2,"carl 4vr"]
If rx'd a majority
v = r
"carl 4vr"
or Vdefault
AcceptRequest[Proposal[9,
"carl 4vr"
Decision[Proposal[n,v]]

Alice P A L

Bob P A L

Carl P A L

$v_{default}$ = "aliz rulz"

Proposal[2,"carl 4vr"]

P prepare request: 9 → A

P prepare request: 9 → A

P prepare request: 9 → A

ResponseToPrepareRequest[N...]

Proposal[2,"carl 4vr"]

P ← ResponseToPrepareRequest[...]

Proposal[2,"carl 4vr"]

P ← ResponseToPrepareRequest[None] → A

Proposal[2,"carl 4vr"]

If rx'd a majority
v = n... "carl 4vr" or $v_{default}$

P AcceptRequest[Proposal[9, "carl 4vr"] → A

P AcceptRequest[Proposal[9, "carl 4vr"] → A

P AcceptRequest[Proposal[9, "carl 4vr"] → A → Decision[Proposal[n,v]] → L

L ← Decision[Proposal[n,v]]

Decision[Proposal[n,v]] → L

L ← A → Decision[Proposal[n,v]] → L

Decision[Proposal[n,v]]

Decision[Proposal[n,v]]

If rx'd a majority
value = v = "carl 4vr"

Alice P A L

Bob P A L

Carl P A L

$v_{default}$ = "aliz rulz"

Proposal[2,"carl 4vr"]

P prepare request: 9 → A

P prepare request: 9 → A

P prepare request: 9 → A

ResponseToPrepareRequest[N…]

Proposal[2,"carl 4vr"]

P ← ResponseToPrepareRequest[…]

Proposal[2,"carl 4vr"]

P ← ResponseToPrepareRequest[None] → A

Proposal[2,"carl 4vr"]

If rx'd a majority,
v = n… "carl 4vr" or $v_{default}$

P AcceptRequest[Proposal[9, "carl 4vr"] → A

P AcceptRequest[Proposal[9, "carl 4vr"] → A

P AcceptRequest[Proposal[9, "carl 4vr"] → A → Decision[Proposal[n,v]] → L

L ← Decision[Proposal[n,v]]

Decision[Proposal[n,v]]

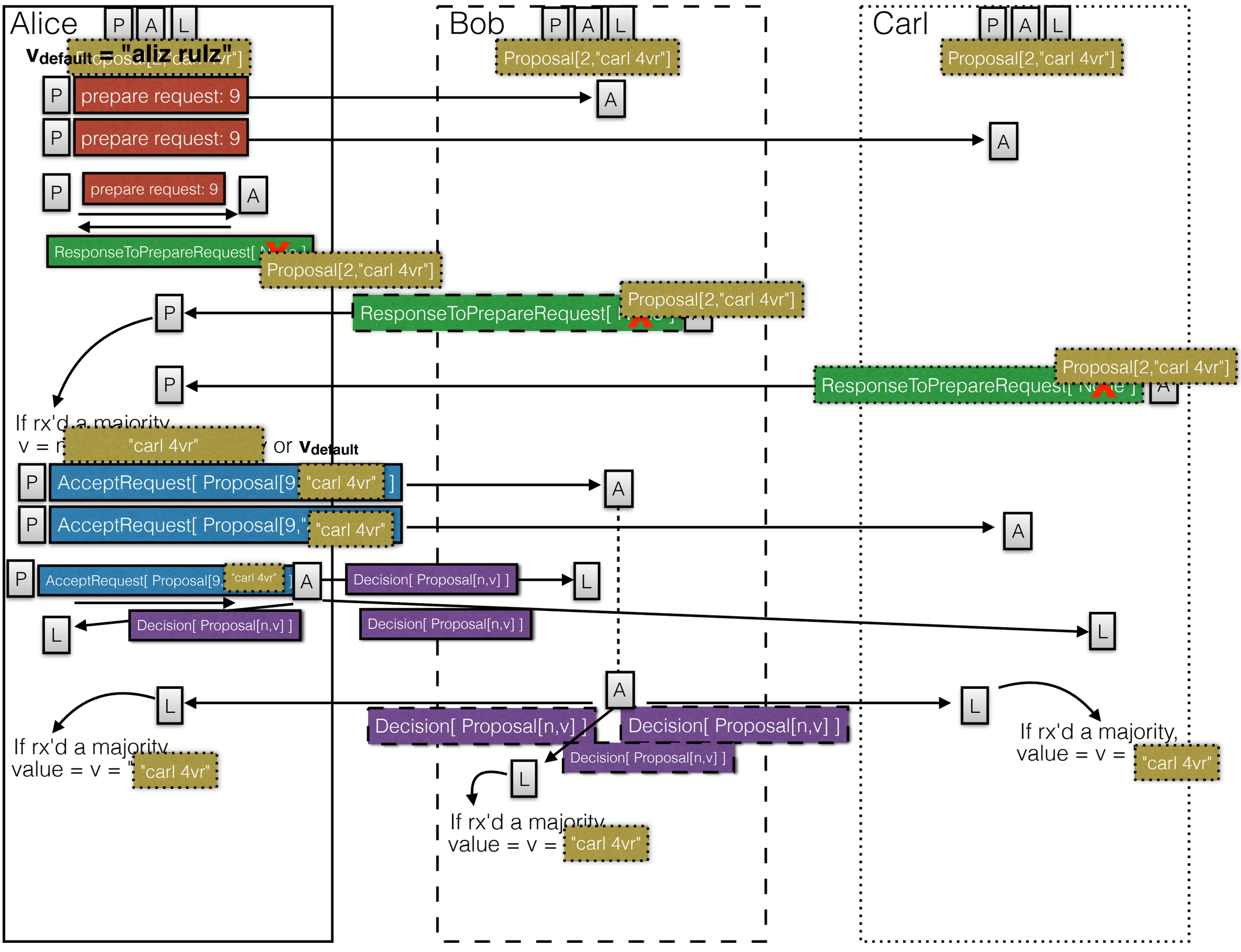
L

L ← A → Decision[Proposal[n,v]]

Decision[Proposal[n,v]]

L → If rx'd a majority, value = v = "carl 4vr"

If rx'd a majority,
value = v = "carl 4vr"


Alice
P
A
L
Bob
Carl
Vdefault = "aliz rulz"
Proposal[2,"carl 4vr"]
prepare request: 9
ResponseToPrepareRequest[
Proposal[2,"carl 4vr"]
If rx'd a majority
v = r
"carl 4vr"
or Vdefault
AcceptRequest[Proposal[9,
"carl 4vr"
Decision[Proposal[n,v]]
If rx'd a majority,
value = v =
"carl 4vr"

Alice P A L

Bob P A L

Carl P A L

$v_{default}$ = "aliz rulz"

Proposal[2,"carl 4vr"]

P prepare request: 9 → A

P prepare request: 9 → A

P prepare request: 9 → A

ResponseToPrepareRequest[N…]

Proposal[2,"carl 4vr"]

P ← ResponseToPrepareRequest[…] Proposal[2,"carl 4vr"]

P ← ResponseToPrepareRequest[None] Proposal[2,"carl 4vr"]

If rx'd a majority,
v = n… "carl 4vr" or $v_{default}$

P AcceptRequest[Proposal[9, "carl 4vr"] → A

P AcceptRequest[Proposal[9, "carl 4vr"] → A

P AcceptRequest[Proposal[9, "carl 4vr"] → A

A Decision[Proposal[n,v]] → L

L ← Decision[Proposal[n,v]]

Decision[Proposal[n,v]] → L

L

A

Decision[Proposal[n,v]] Decision[Proposal[n,v]]

Decision[Proposal[n,v]]

L

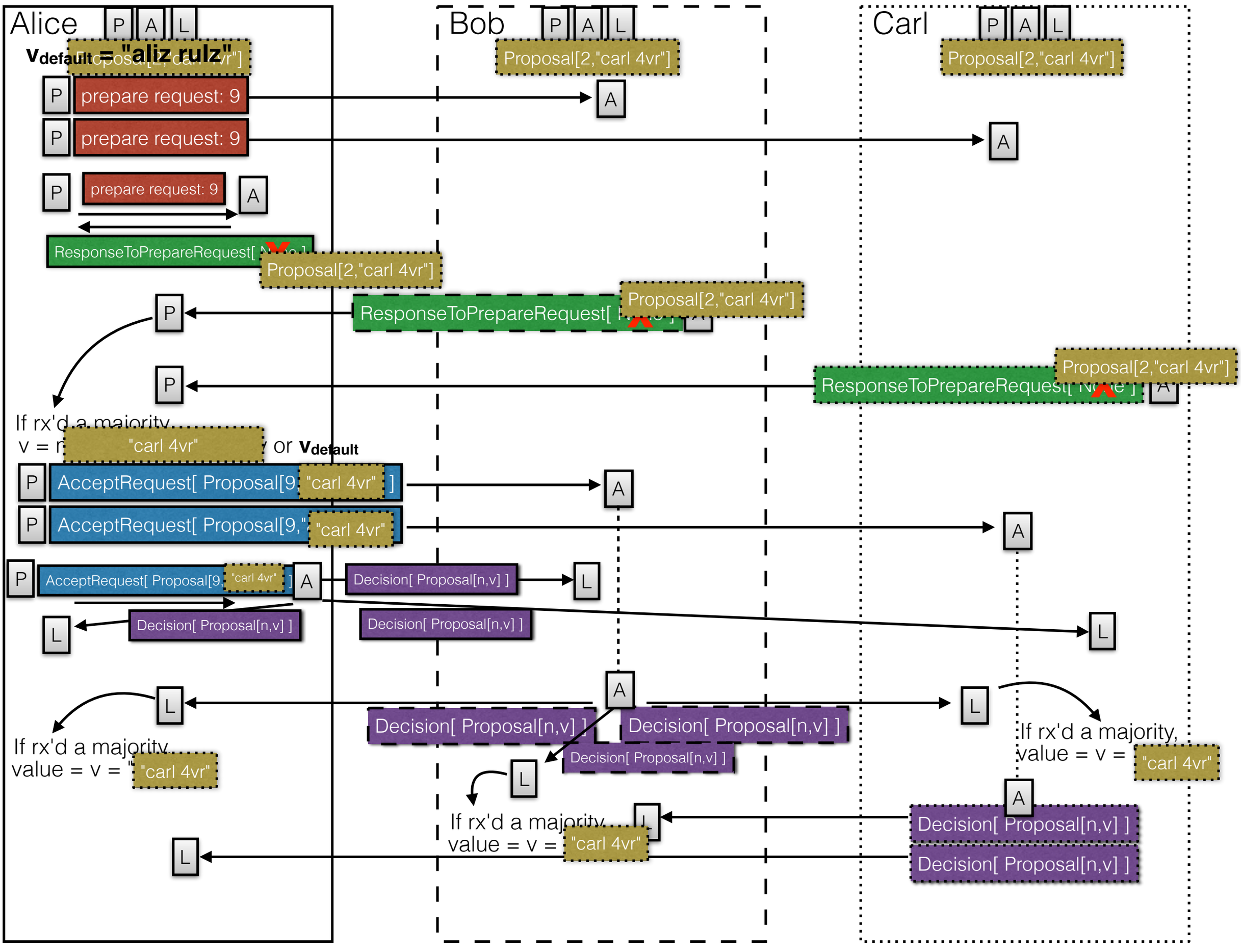
L

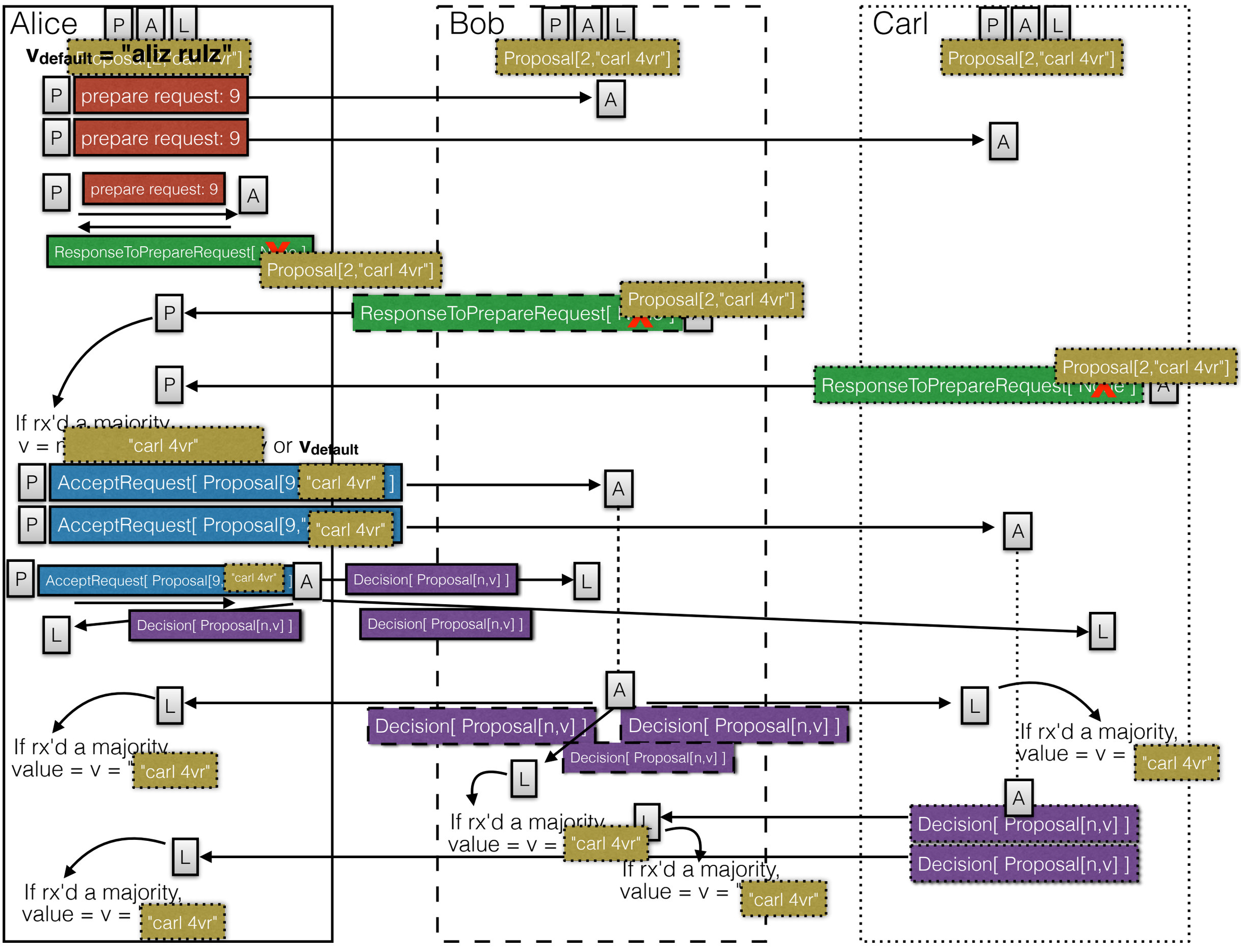
If rx'd a majority,
value = v = "carl 4vr"

If rx'd a majority,
value = v = "carl 4vr"

If rx'd a majority,
value = v = "carl 4vr"

A


Alice
P
A
L
Bob
Carl
Vdefault = "aliz rulz"
Proposal[2,"carl 4vr"]
prepare request: 9
ResponseToPrepareRequest[
Proposal[2,"carl 4vr"]
If rx'd a majority
v = r
"carl 4vr"
or Vdefault
AcceptRequest[Proposal[9,
"carl 4vr"
Decision[Proposal[n,v]]
If rx'd a majority,
value = v =
"carl 4vr"


Alice
P
A
L
Bob
Carl
Vdefault = "aliz rulz"
Proposal[2,"carl 4vr"]
prepare request: 9
ResponseToPrepareRequest[
Proposal[2,"carl 4vr"]
If rx'd a majority
v = r
"carl 4vr"
or Vdefault
AcceptRequest[Proposal[9,
"carl 4vr"
Decision[Proposal[n,v]]
If rx'd a majority,
value = v =
"carl 4vr"

Alice P A L

Bob P A L

Carl P A L

$v_{default}$ = "aliz rulz"

Proposal[2,"carl 4vr"]

P prepare request: 9 → A

P prepare request: 9 → A

P prepare request: 9 A

ResponseToPrepareRequest[N…]

Proposal[2,"carl 4vr"]

P ← ResponseToPrepareRequest[…] Proposal[2,"carl 4vr"]

P ← ResponseToPrepareRequest[None] Proposal[2,"carl 4vr"]

If rx'd a majority,
v = n… "carl 4vr" or $v_{default}$

P AcceptRequest[Proposal[9, "carl 4vr"] → A

P AcceptRequest[Proposal[9, "carl 4vr"] → A

P AcceptRequest[Proposal[9, "carl 4vr"] A

Decision[Proposal[n,v]] → L

L ← Decision[Proposal[n,v]]

Decision[Proposal[n,v]] → L

L

A

Decision[Proposal[n,v]]

Decision[Proposal[n,v]]

Decision[Proposal[n,v]]

L

L

If rx'd a majority,
value = v = "carl 4vr"

If rx'd a majority,
value = v = "carl 4vr"

L

If rx'd a majority,
value = v = "carl 4vr"

A

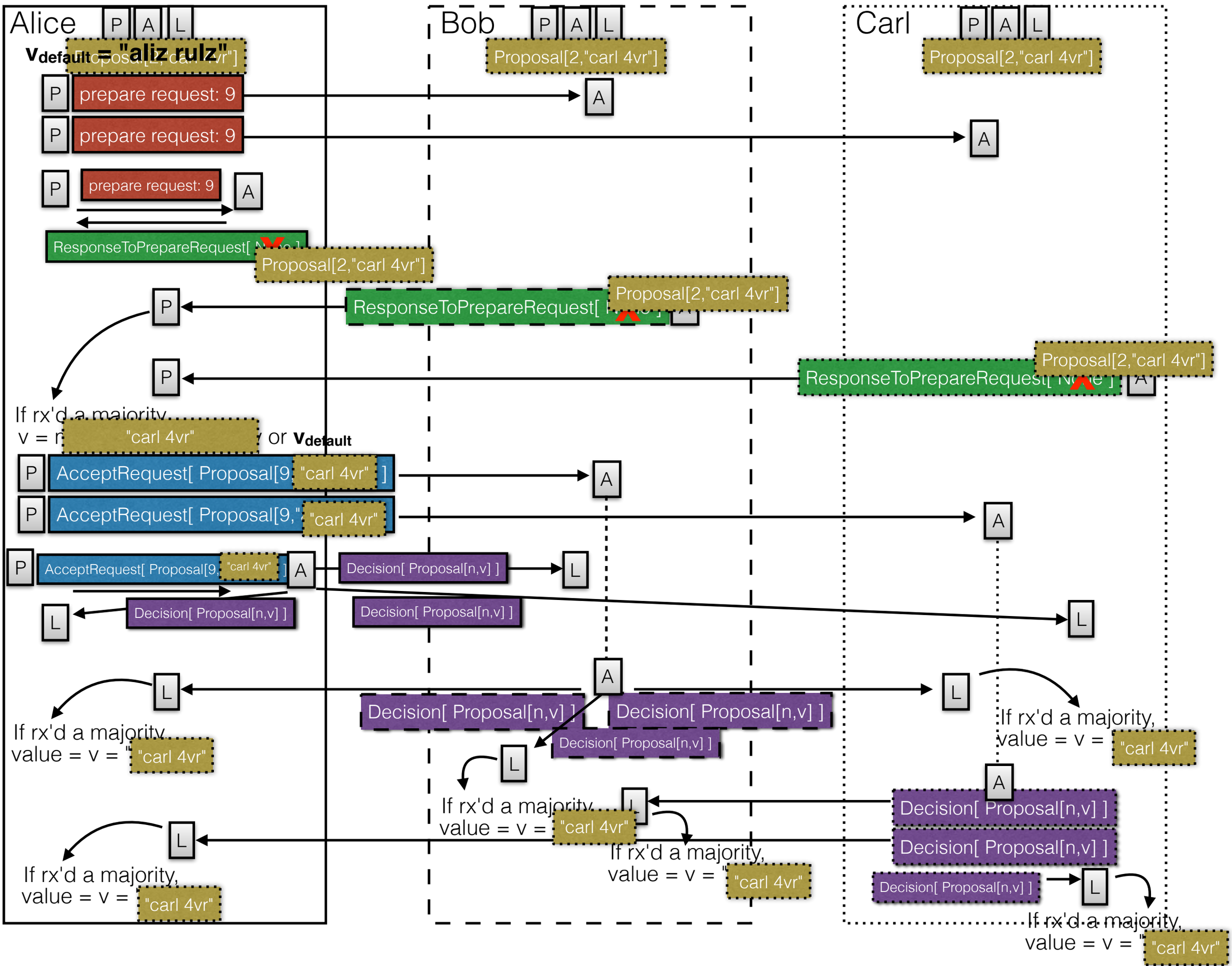
Decision[Proposal[n,v]]

Decision[Proposal[n,v]]

Decision[Proposal[n,v]] → L

L

If rx'd a majority,
value = v = "carl 4vr"


Alice
P
A
L
Bob
Carl
Vdefault = "aliz rulz"
Proposal[2,"carl 4vr"]
prepare request: 9
ResponseToPrepareRequest[
Proposal[2,"carl 4vr"]
If rx'd a majority,
v = r
or Vdefault
"carl 4vr"
AcceptRequest[Proposal[9,
"carl 4vr"
Decision[Proposal[n,v]]
If rx'd a majority,
value = v =
"carl 4vr"

Outline

- The Algorithm
- Example: how it works **initially**
- Example: how it handles **conflicts**
- Example: how it works **after consensus**

THE END