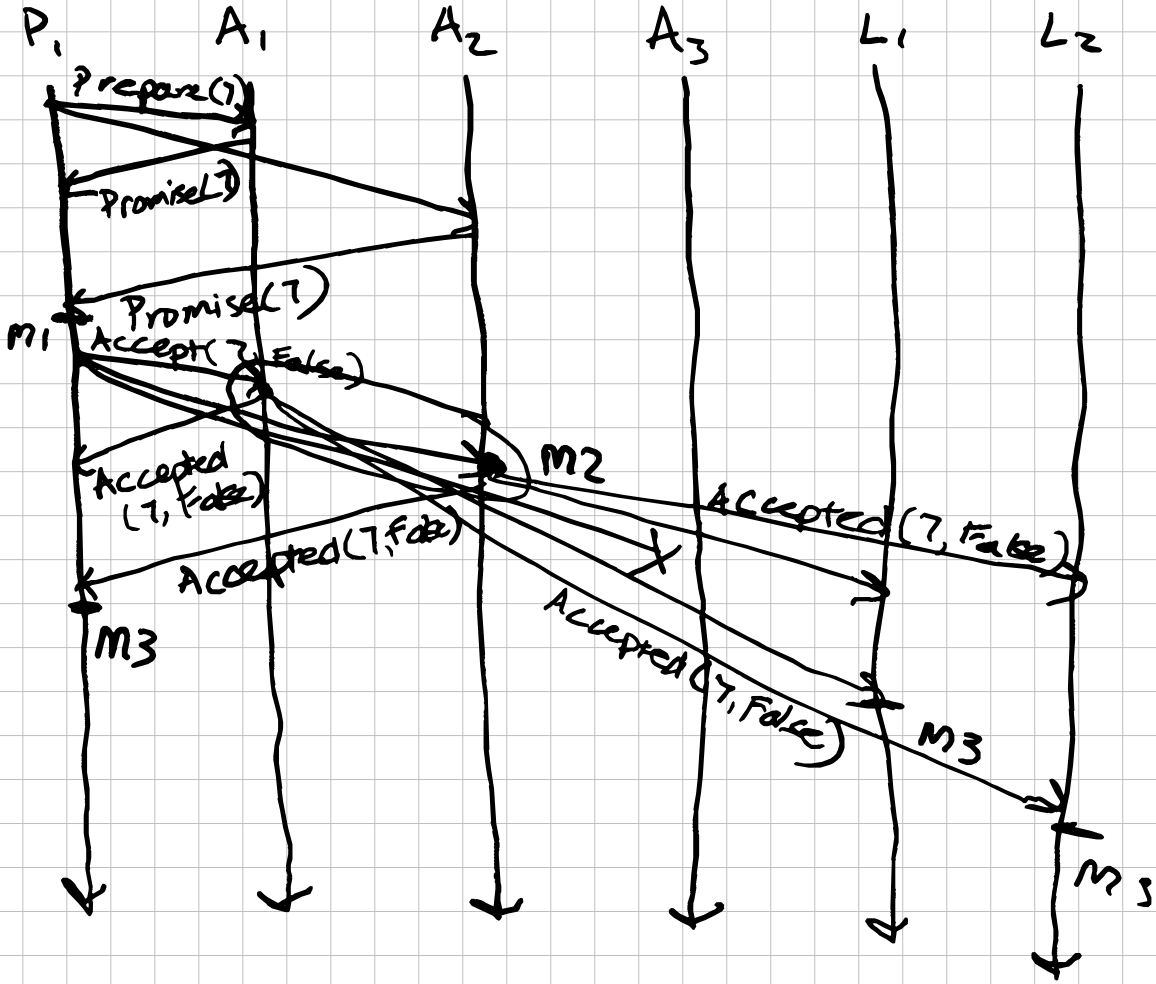
CSE138 Lecture 13

this time
- Paxos: the interesting parts
- Multi-Paxos



Phase 1 of Paxos:

Proposer

To propose a value, a proposer P sends Prepare(n) to a majority of acceptors.

n must be:
- unique
- higher than anything P has tried before.

Acceptor

When an acceptor receives Prepare(n),

"Did I previously promise to ignore messages with this proposal number?"

- If yes: 'ignores it.'
- If no: it asks,

"Have I previously accepted anything?"

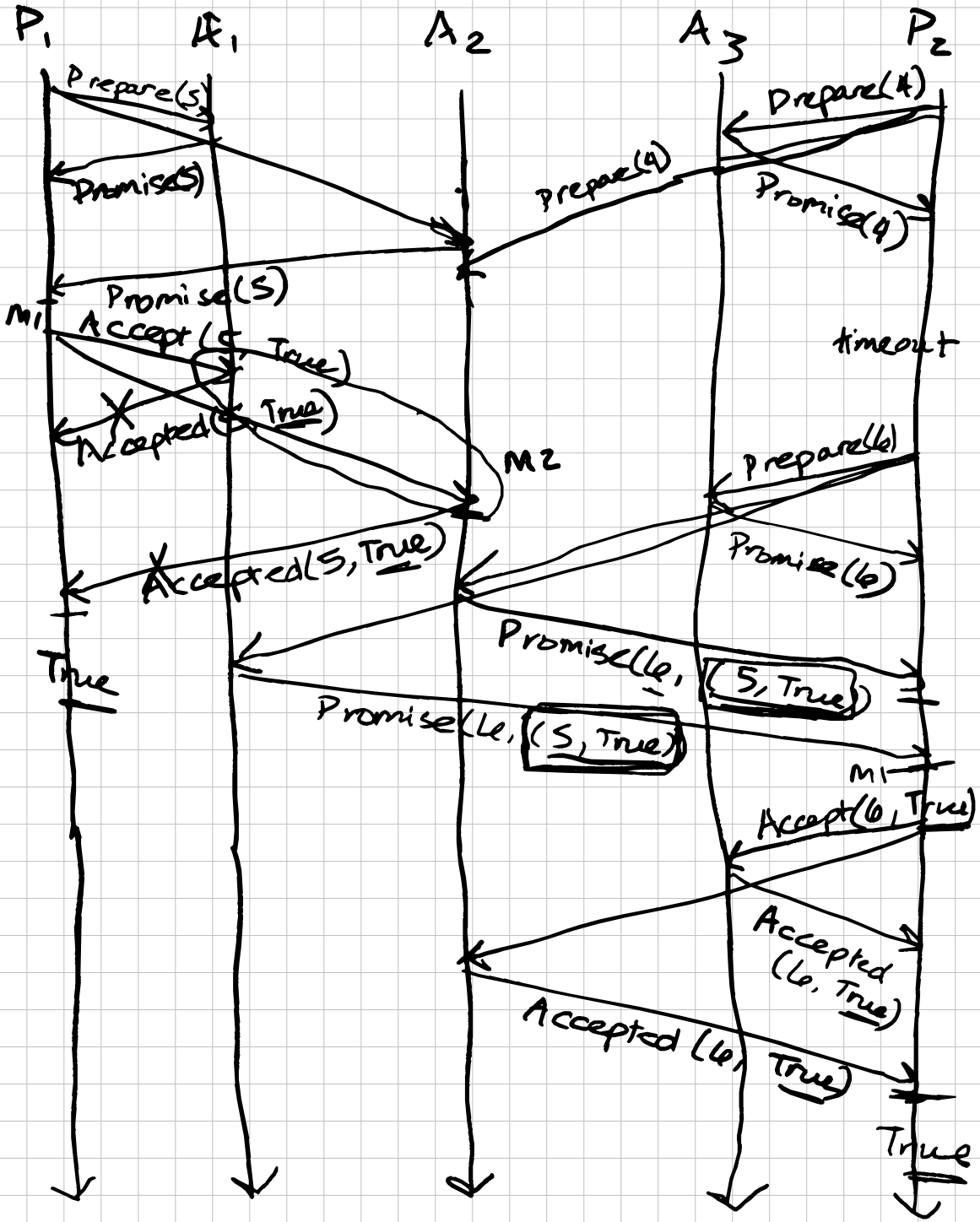
- if yes:

respond to the proposer with Promise(n, (n_{prev}, val_{prev}))

n_{prev}: highest previously accepted proposal number

val_{prev}: previously accepted value

- if no: respond with Promise(n).


P1
A1
A2
A3
P2
Prepare(5)
Promise(5)
Prepare(4)
prepare(4)
Promise(4)
Promise(5)
M1
Accept(5, True)
timeout
Accepted(5, True)
M2
prepare(6)
Promise(6)
Accepted(5, True)
Promise(6, (5, True))
True
Promise(6, (5, True))
M1
Accept(6, True)
Accepted (6, True)
Accepted (6, True)
True

Phase 2 of Paxos:

either kind ↓

Proposer:

when a proposer has received **Promise** messages from a majority of acceptors for a particular proposal number n, it sends Accept(n, val) message to a majority of acceptors, where val is chosen as follows:

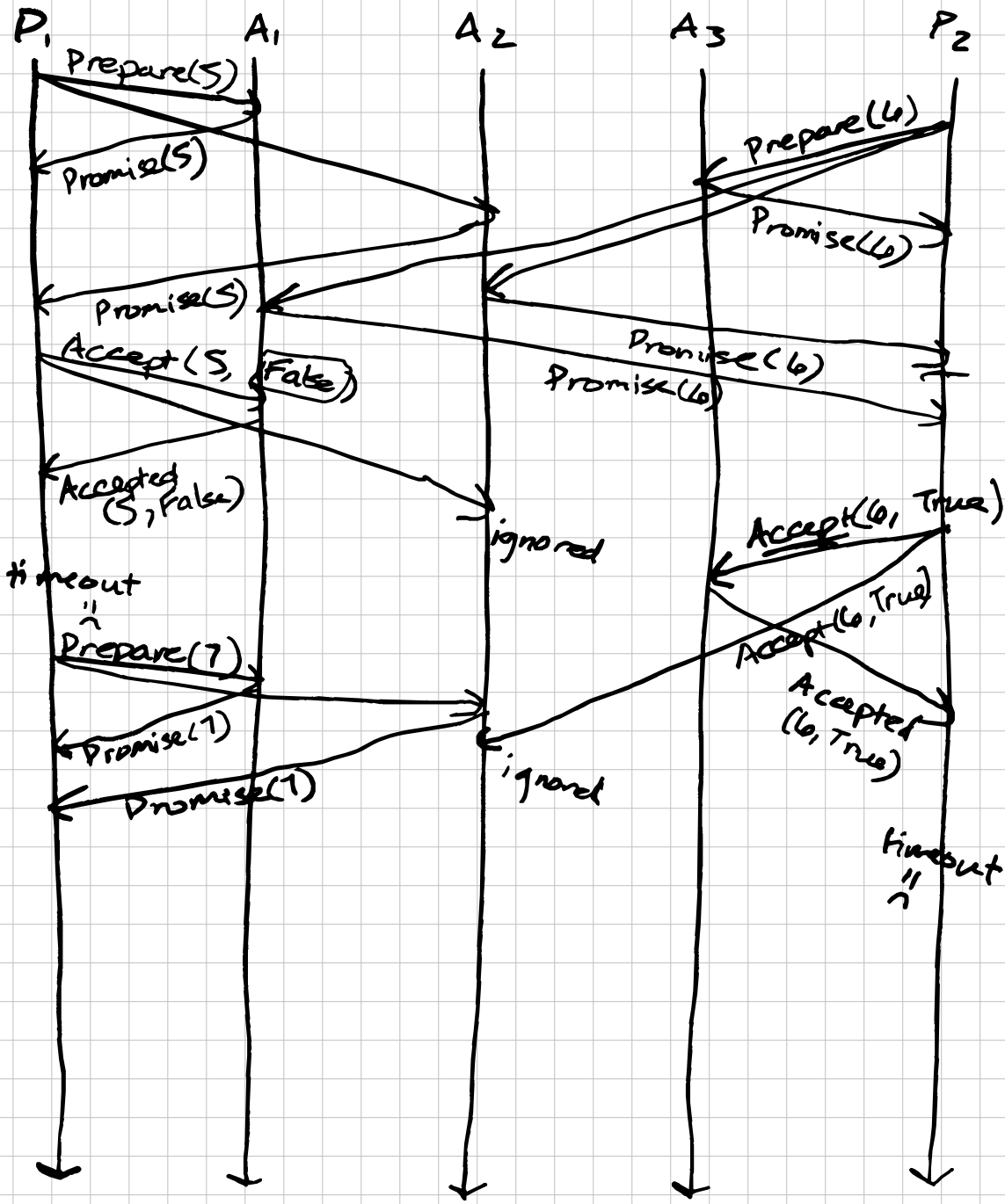
The proposer asks, "Have I gotten any (n_{prev}, val_{prev}) pairs from acceptors?"

- If yes: val must be the val_{prev} that went with the highest n_{prev}.
- If no: val can be anything the proposer wants.

Acceptor:

When an acceptor receives Accept(n, val):
It asks: "Did I previously promise to ignore messages with this proposal number?"
- If yes: ignores it!
- If no: responds with Accepted(n, val), and also send Accepted(n, val) to all learners.

"dueling proposers" scenario!



Consensus isn't reached because neither proposer can get a majority of acceptors to respond to Accept messages, so phase 2 can never complete.

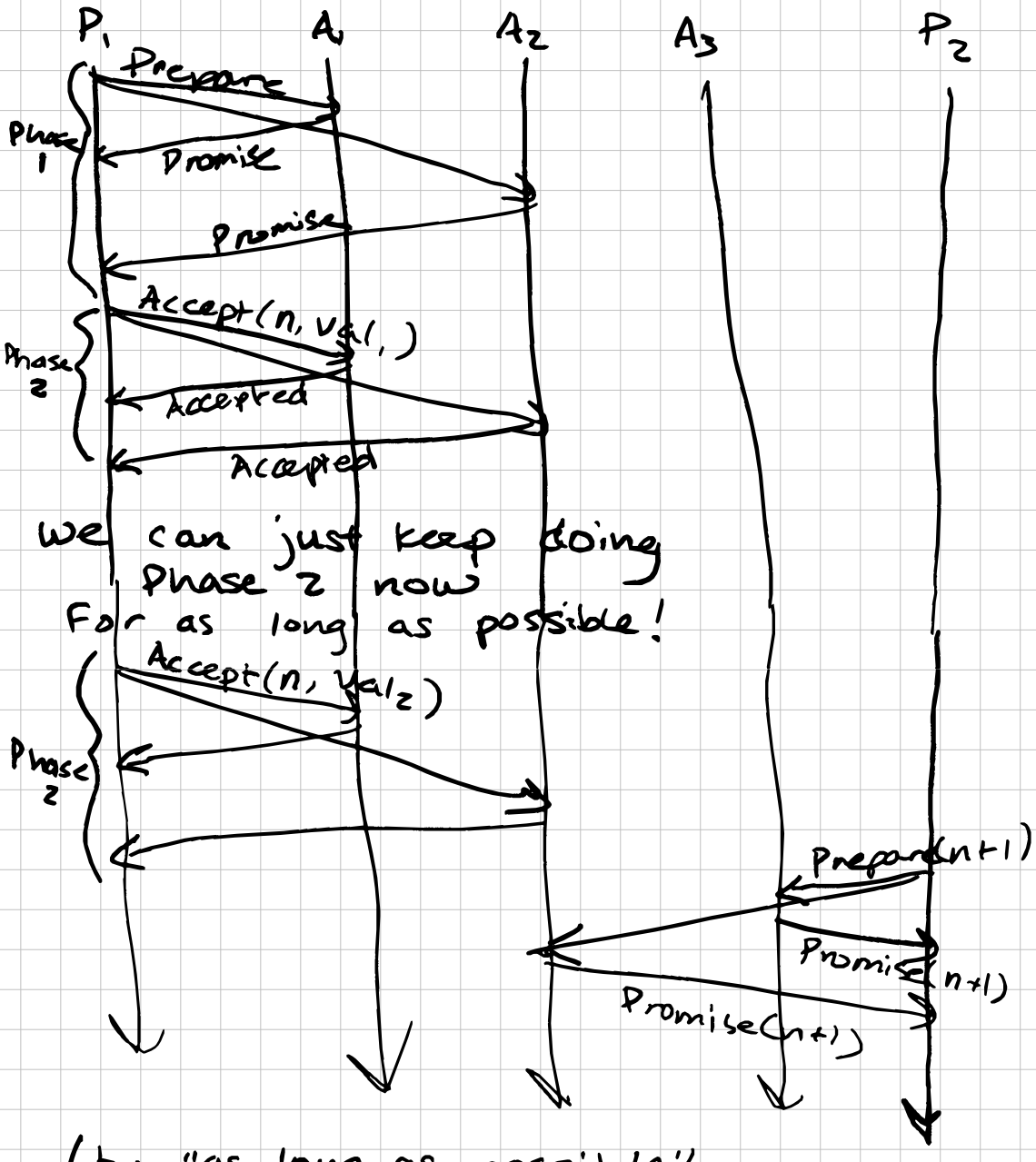
Multi-Paxos

What if you want to decide on not just one value, but a sequence of values?

(example: totally-ordered broadcast)

What message is... first?
second?
third?
⋮

What can be done to make things faster?



We can just keep doing Phase 2 now for as long as possible!

(by "as long as possible", we mean "until someone else interrupts us" (or until we crash)).

It turns out that this is safe to do!

How fault-tolerant is Paxos?

If we have 3 acceptors, how many can crash? One.

In general, a minority can crash and we still have a chance of terminating.

If f is the number of crashed acceptors that you want to tolerate, you need $2f+1$ acceptors.

What about proposers?

If f is the number of crashed proposers to tolerate, we need $f+1$ proposers.