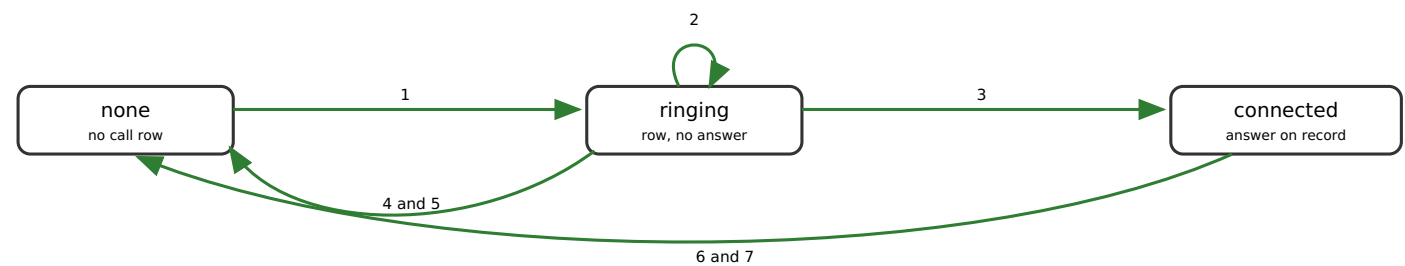


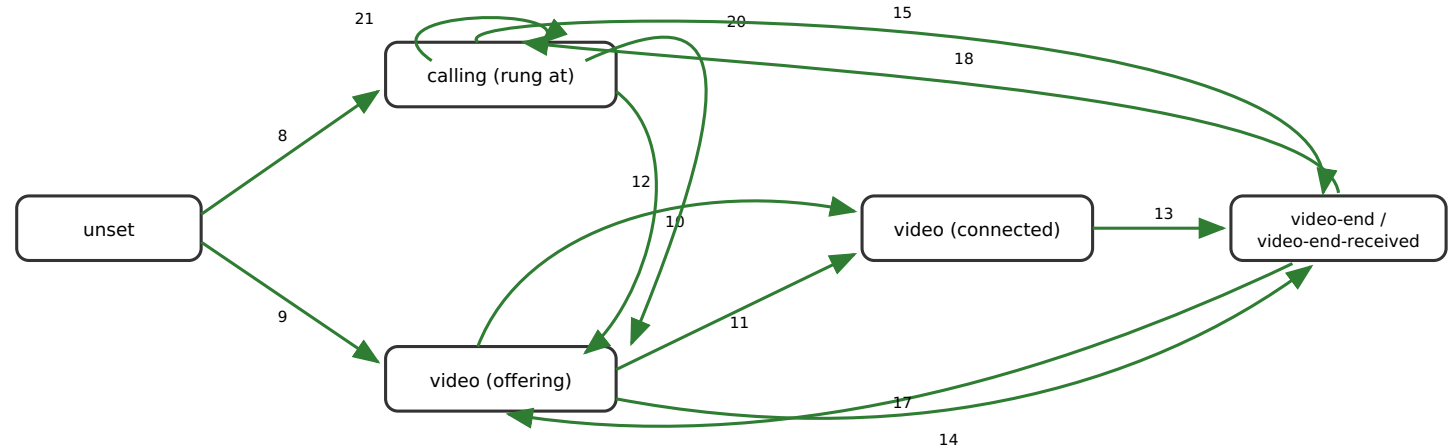
Yioop calling: states, transitions, timings, and the code for each

Every arrow ends on a state and every state has a way out, so no state is a dead end, and a call placed again after one that failed is a transition of its own rather than a repeat of the first. The last column says for each transition whether two things happening at once could give a wrong result, and what rules that out. Each arrow is numbered, and the table gives the time it takes and the file and line that carries it. Line numbers are against gold 21feeee1e plus this patch.

Server: the phase of one call (GroupModel::callPhase, GroupModel.php:2625)



Browser: the state of one screen (call_state, messages.js)



Each transition: what causes it, how long it takes, and the code

#	From → To	Cause	Time	Code	Race, and what rules it out
1	none → ringing	caller presses call; client-call reaches the server	at once	MessageComponent.php:867 createGroupCall	None. One writer, the caller, and the row is made in the request that carries client-call.
2	ringing → ringing	caller's screen polls; the row's time moves up	every 3s	MessageComponent.php:811 refreshGroupCall	None. The refresh only moves a time forward, so two refreshes in either order leave the same state.
3	ringing → connected	a client-answer is on record	at once	GroupModel.php:2631 numCallEventsOfType	None. The phase is read from events on record, and an event once written is never unwritten.
4	ringing → none	callee refuses	at once	MessageComponent.php:443 deleteGroupCall	None. Deleting a row twice leaves it deleted. A refusal racing a hangup gives the same end.
5	ringing → none	nobody answers	60s (MAX_CALL_RING_TIME)	GroupModel.php:2635	None. The reading is a comparison of times, not a write, so two readers agree.
6	connected → none	either side hangs up	at once	MessageComponent.php:837 call-end deletes the row	None. Same as 4: the delete settles it whichever side gets there first.
7	connected → none	no event from either side	20s (MAX_CALL_EVENT_AGE)	GroupModel.php:2638	None. A comparison of times against events on record.
8	unset → calling	reads client-call on socket or stream	at once on the channel	messages.js:425 handleClientCall	None. The channel hands events in the order they were written.

9	unset → video (offering)	presses call, or the answer button in a contact row	at once; the answer button waits for the channel to open	messages.js:152 and :199 answerWaitingCall; the wait at messages.js:180	Was one: the screen could offer before its channel was open, losing the reply. Ruled out by answering on the channel opening, and, where no channel opens at all, by taking the call after a four second wait so the screen is never left holding a call it never answered.
10	calling → video (offering)	picks up in the conversation	at once	messages.js:270 makeOffer	None. One screen, one press.
11	video (offering) → video (connected)	reads client-answer	at once on the channel	messages.js:445 handleClientAnswer	None. An answer arriving sets the mark before it is acted on.
12	video (offering) → video (offering), offering again	the phase reads connected but no answer arrived	after 9s (CALL_ANSWER_WAIT), checked every 3s	basic.js:4178	Was one: an ask sent before the call began came back saying no call, and the offer was repeated or the call ended. Ruled out by each answer carrying the moment its ask went out, and an answer older than the call being passed by.
13	in a call → ended	presses hang up, or reads call-end	at once	messages.js:414 handleCallEnd	None. Either the press or the read reaches the end state, and both lead to the same one.
14	video (offering) → ended	the phase reads none: refused, hung up, or aged out	within 3s of the phase changing	basic.js:4176	Same race as 12, ruled out the same way. This was the call that hung up before it reached the other side.
15	calling → ended	nobody picks up at this screen	60s	messages.js:254 cancelCallSound	None. One timer on one screen, cleared when the call is picked up.
16	in a call → none	the page is closed or reloaded	at once, on the way out	messages.js:183 pagehide sends call-end	None. The ending is sent on the way out and the row is deleted by whichever request lands first.
17	ended → video (offering)	the person presses call again	at once	messages.js setCallState, which runs from any state	None. One screen, one press, and setCallState reads and writes only that screen.
18	ended → calling	the other side calls again	at once on the channel	messages.js:425 handleClientCall, which sets calling from any state	None. Reading client-call sets calling whatever the screen was doing.
19	ended → video (offering), a second time	a call that failed is placed again	at once; the screen is proven live again within 3s	messages.js setCallState clears the live mark on every attempt	Was one: the mark saying the server had seen the last call stayed set, so an answer about the moment before this attempt began ended it at once. Ruled out by clearing that mark whenever a screen enters a call, whatever it was doing before.
20	video (offering) → video (offering), as the callee	both people press call in the same moment	within 3s of the ask	basic.js the ask, and messages.js answerWaitingCall	Was one, and the one Chris found: two calls for the same pair race for one row. The database keeps the first and drops the second, so one side holds a call the server never recorded and both sides wait as callers. Ruled out by the side the server reports as being called taking the callee part, and only where that side placed the call itself: a screen picking up is also a screen the other side is calling, so without that the pick up restarted the call instead of answering it.
21	calling → calling, rung a second time	the channel and the ask both say the same call is ringing	at once, then within 3s	messages.js startCallSound	Was one: each start laid a new flashing timer over the last and the ringing ran at twice the rate, while stopping cleared only the newest, so hanging up left it ringing. Ruled out by a start clearing any timer already running.

Worst case for a call to show on the other side: 3s, one ask. Worst case for a lost answer to be noticed and the offer repeated: 9s plus up to 3s. Worst case for a call nobody answers to clear: 60s.