Summary of P802.1CQ/D0.5 in Task Group Ballot, July 2020

Roger Marks (EthAirNet Associates)
Antonio de la Oliva (InterDigital)

2020-07-15

P802.1CQ Administrative Status

- P802.1CQ Multicast and Local Address Assignment
  - https://1.ieee802.org/tsn/802-1cq/
- PAR approved 2016-02-05
  - renewed 2020-06-03; expires 2022-12-31
- assigned to TSN, 2019-07 (previously DCB, then OmniRAN)
- 802.1 WG Motion (approved 2019-07-18):
  - *802.1 authorizes Roger Marks, the Editor of P802.1CQ Multicast and Local Address Assignment to prepare drafts for and conduct Task Group balloting*
- Draft 0.3 released 2019-10-21
- TSN last updated in presentation of 2020-03-19
- Draft 0.5 released 2020-06-30
  - 49 pages, plus frontmatter
  - Task Group Ballot, 2020-06-30 through 2020-07-31

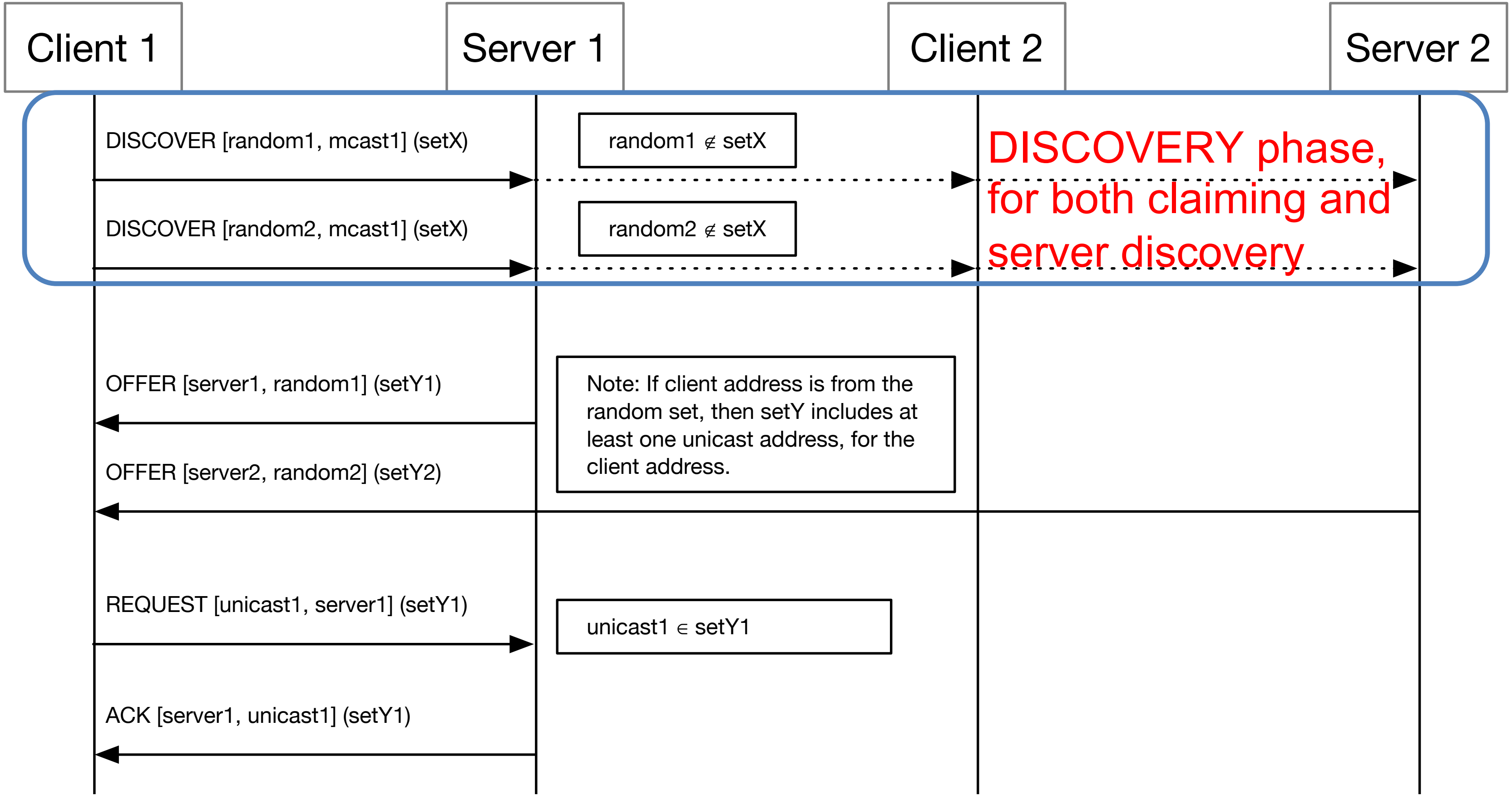
Key PAR Content

- Scope
  - *This standard specifies protocols, procedures, and management objects for locally-unique assignment of 48-bit and 64-bit addresses in IEEE 802 networks. <u>Peer-to-peer address claiming and address server capabilities are specified</u>.*
- Need for the Project:
  - *Currently, global addresses are assigned to most IEEE 802 end station and bridge ports. Increasing use of virtual machines and Internet of Things (IoT) devices could exhaust the global address space. To provide a usable alternative to global addresses for such devices, this project will define a <u>set of protocols that will allow ports to automatically obtain a locally-unique address in a range from a portion of the local address space</u>. Multicast flows also need addresses to identify the flows. They will benefit from a <u>set of protocols to distribute multicast addresses</u>. Peer-to-peer address claiming and address server capabilities will be included to serve the needs of smaller (e.g. home) and larger (e.g. industrial plants and building control) networks.*

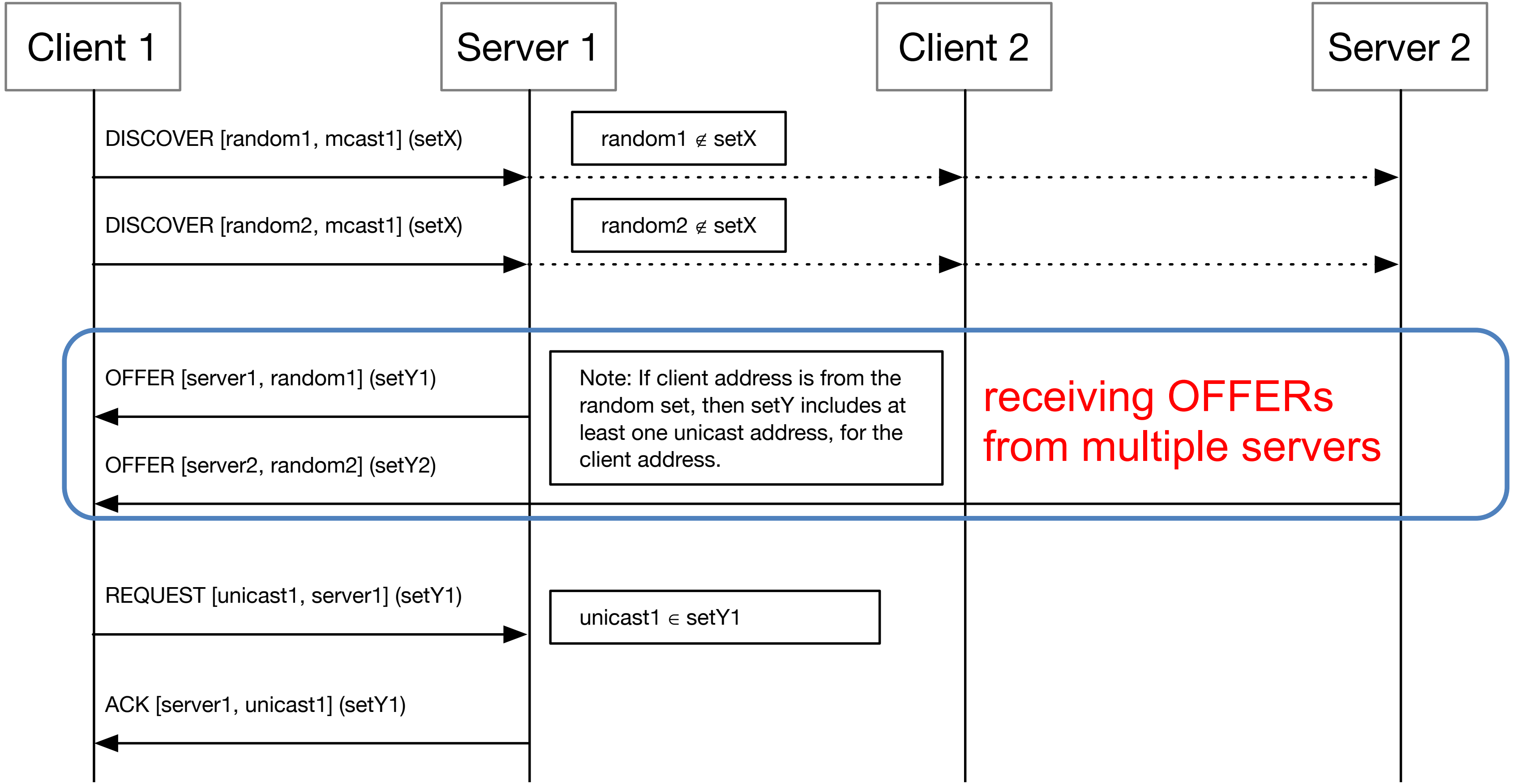
Key Points of P802.1CQ/D0.5

- Includes MAC Address Acquisition Protocol (MAAP) from IEEE Std 1722-2016, with minor error corrections, as one method
  - Claiming only (no client-server assignment)
- new **Protocol for Assignment of Local and Multicast Addresses** (PALMA)
  - does *not* use MAAP Ethertype or MAAP address ranges
  - both claiming and client-server assignment
- Messaging begins with DISCOVER (multicast from client; seeks server or peer)
  - client may adopt a source address chosen randomly from a specified range
    - note that we call it “client” even when there is no server
    - client may specify the requested range of addresses
- If server hears DISCOVER:
  - Server sends OFFER to client
  - Client sends REQUEST to server
  - Server sends assignment in ACK to client
- If peer hears DISCOVER claim for address already in use:
  - Peer sends DEFEND to client
  - Client sends multicast ANNOUNCE (stating self-assigned use of undefended claim)

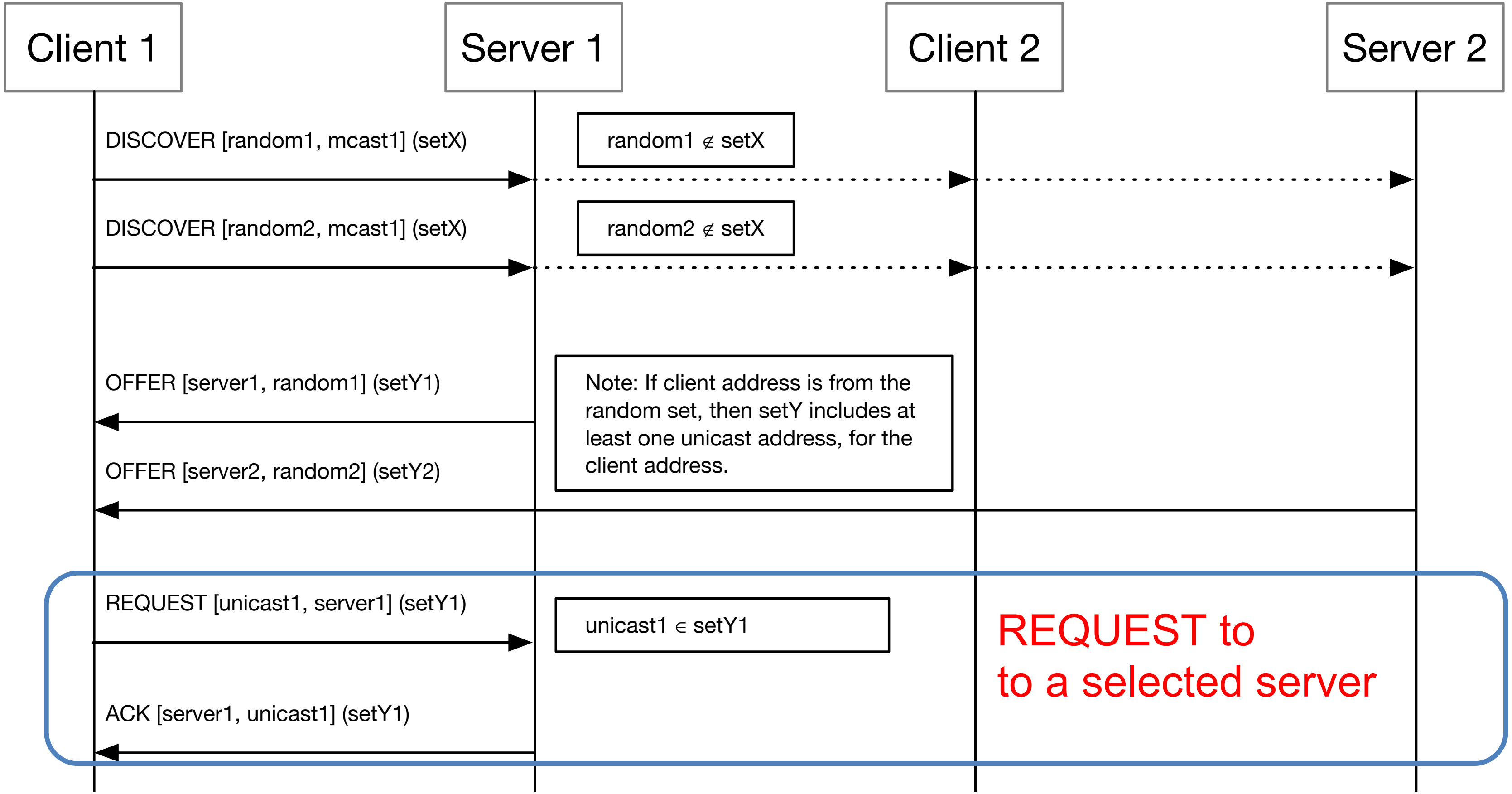
PALMA Client/Server Message Exchange



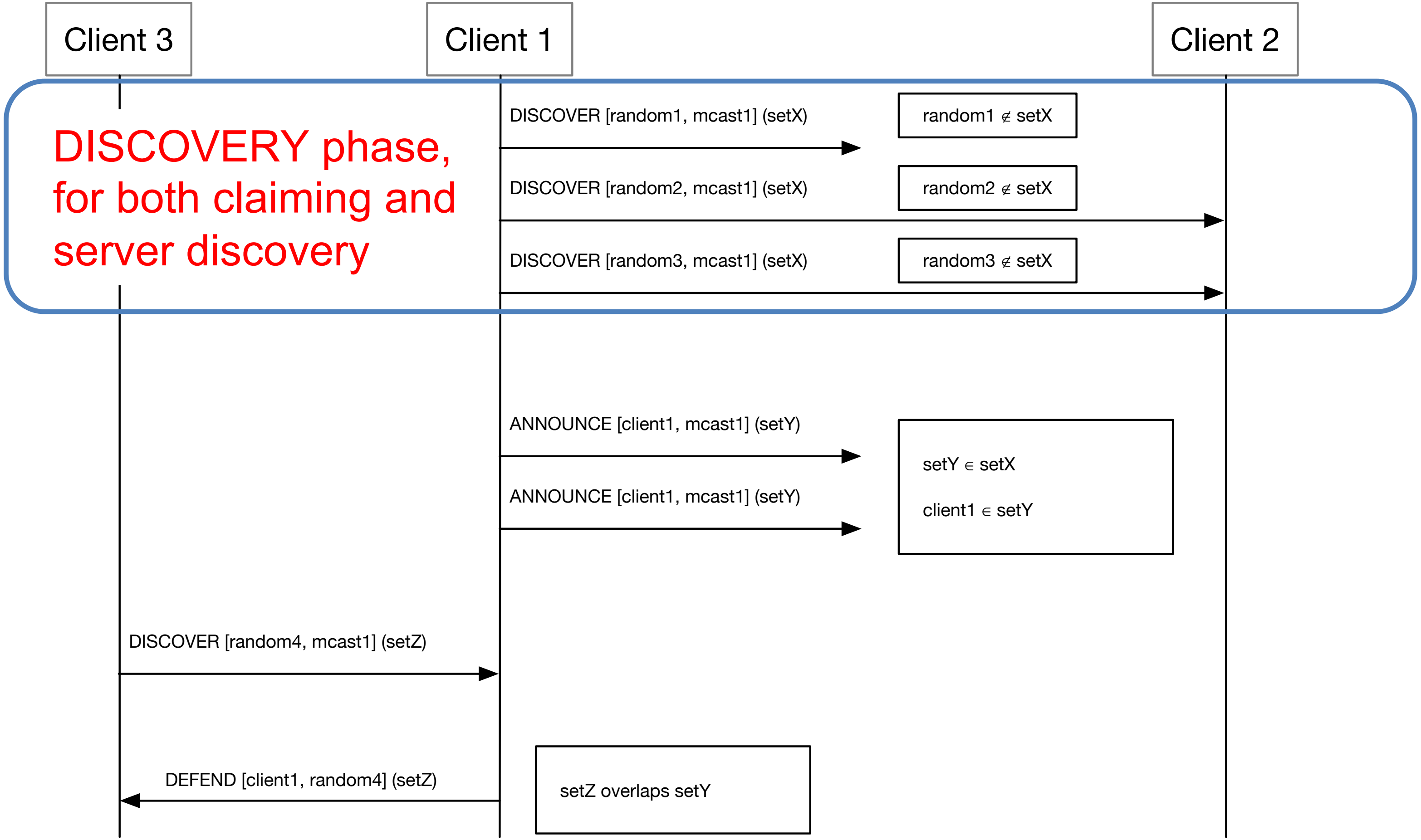
PALMA Client/Server Message Exchange



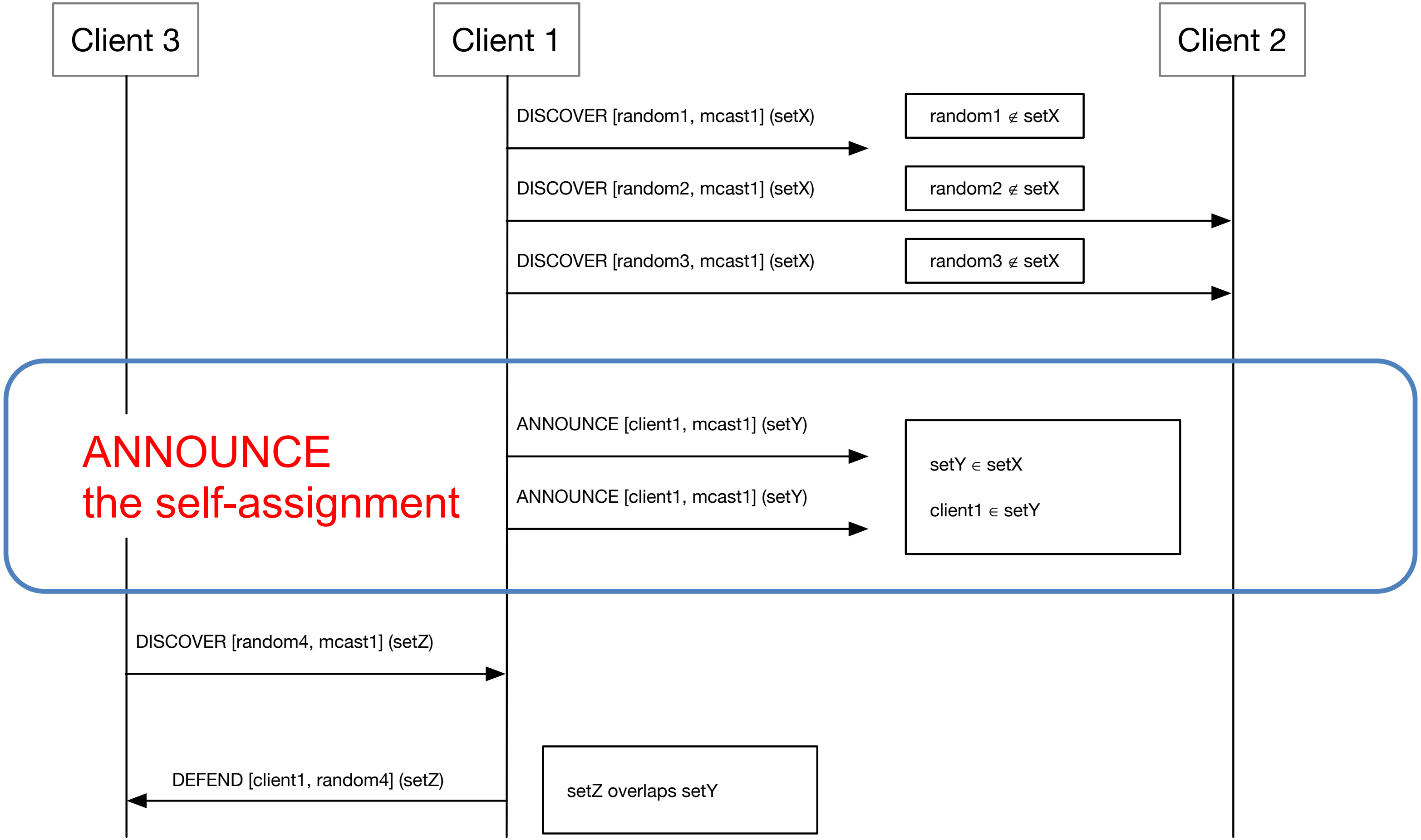
PALMA Client/Server Message Exchange



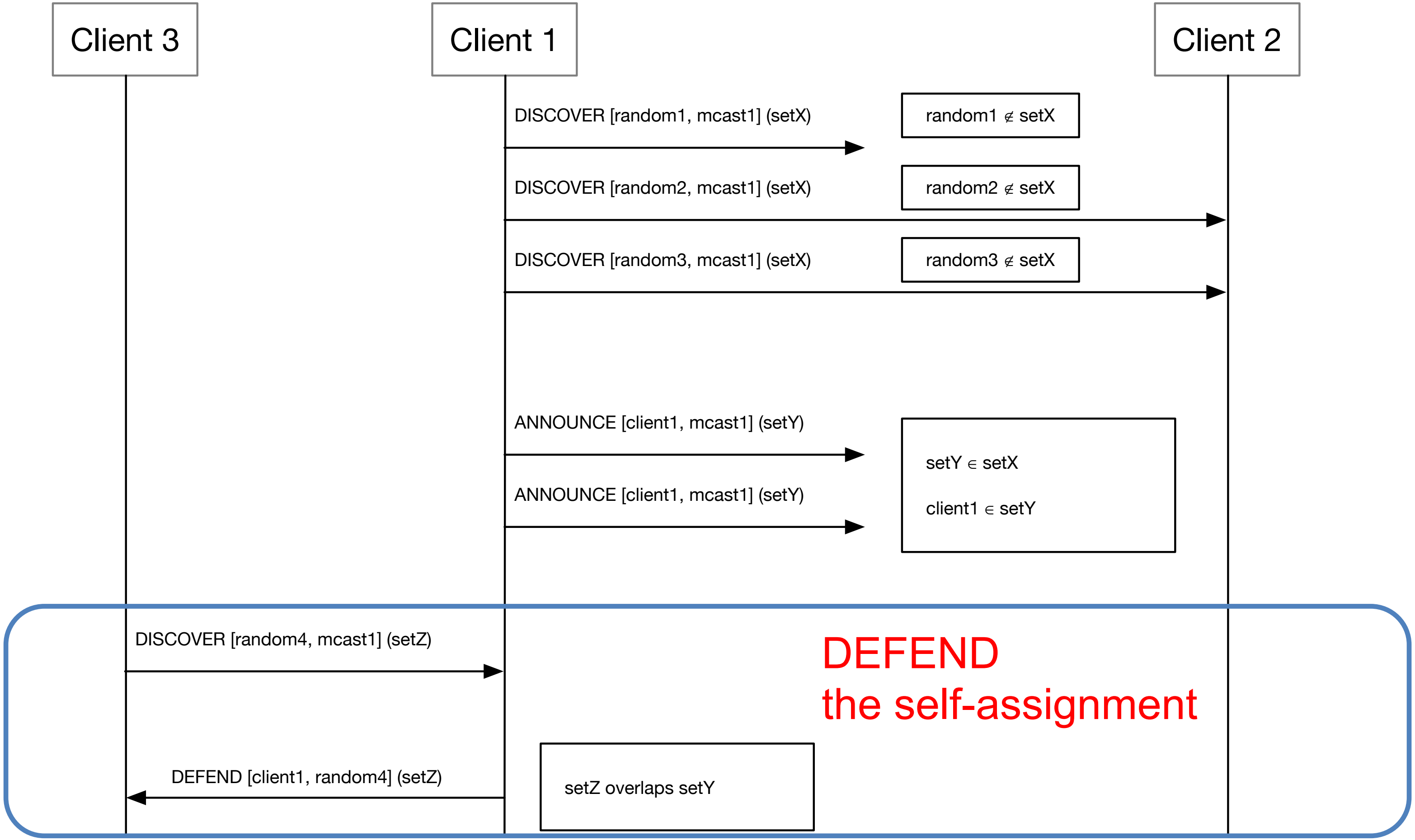
PALMA Claiming Message Exchange



PALMA Claiming Message Exchange



PALMA Claiming Message Exchange

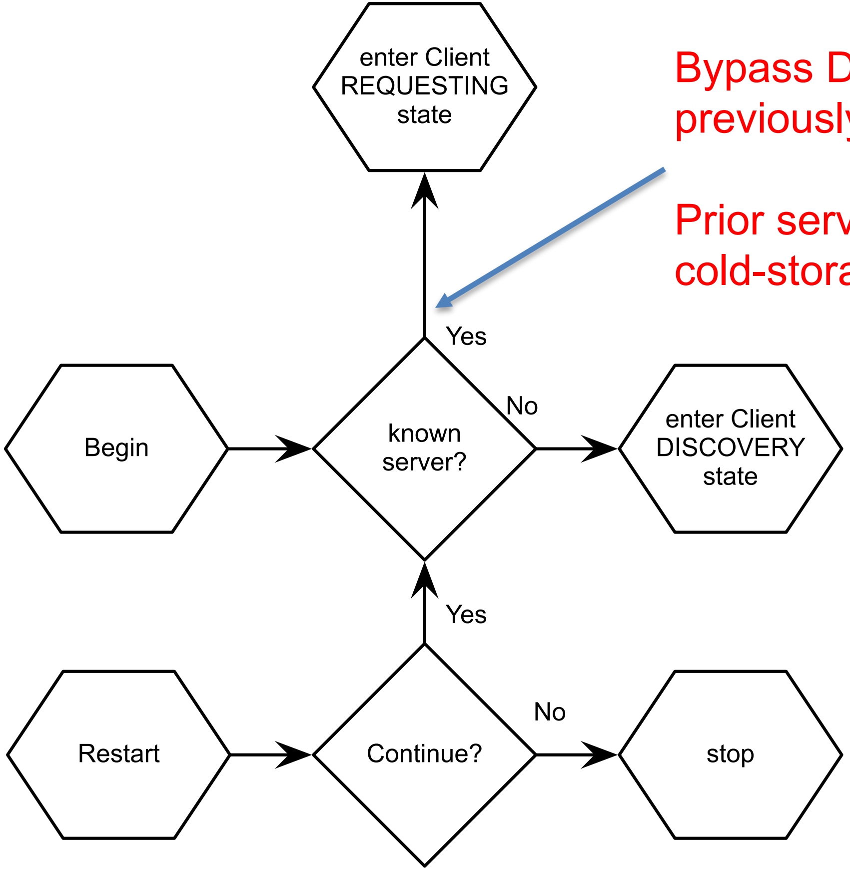


Address Pools

- Each server needs a disjoint address pool
- Address pool for claiming is disjoint from all server pools
- Device without an address chooses from range beginning with 2A-00.
- A device may issue a DISCOVER proposing a prior PALMA-assigned address.
- If a server offers an address, device is prohibited from self-assigning.

Usage	Initial Address Octets	SLAP Quadrant	PALMA assignment method	Address count limit
unicast	0A	11 (SAI)	self-assignment	16
unicast	1A	11 (SAI)	server assignment	-
multicast	0B	11 (SAI)	self-assignment	-
multicast	1B	11 (SAI)	server assignment	-
unicast or multicast	per assigned CID	01 (ELI)	server assignment, in accordance with authorization per assigned CID	-
unicast or multicast	any	00 (AAI)	server assignment	-

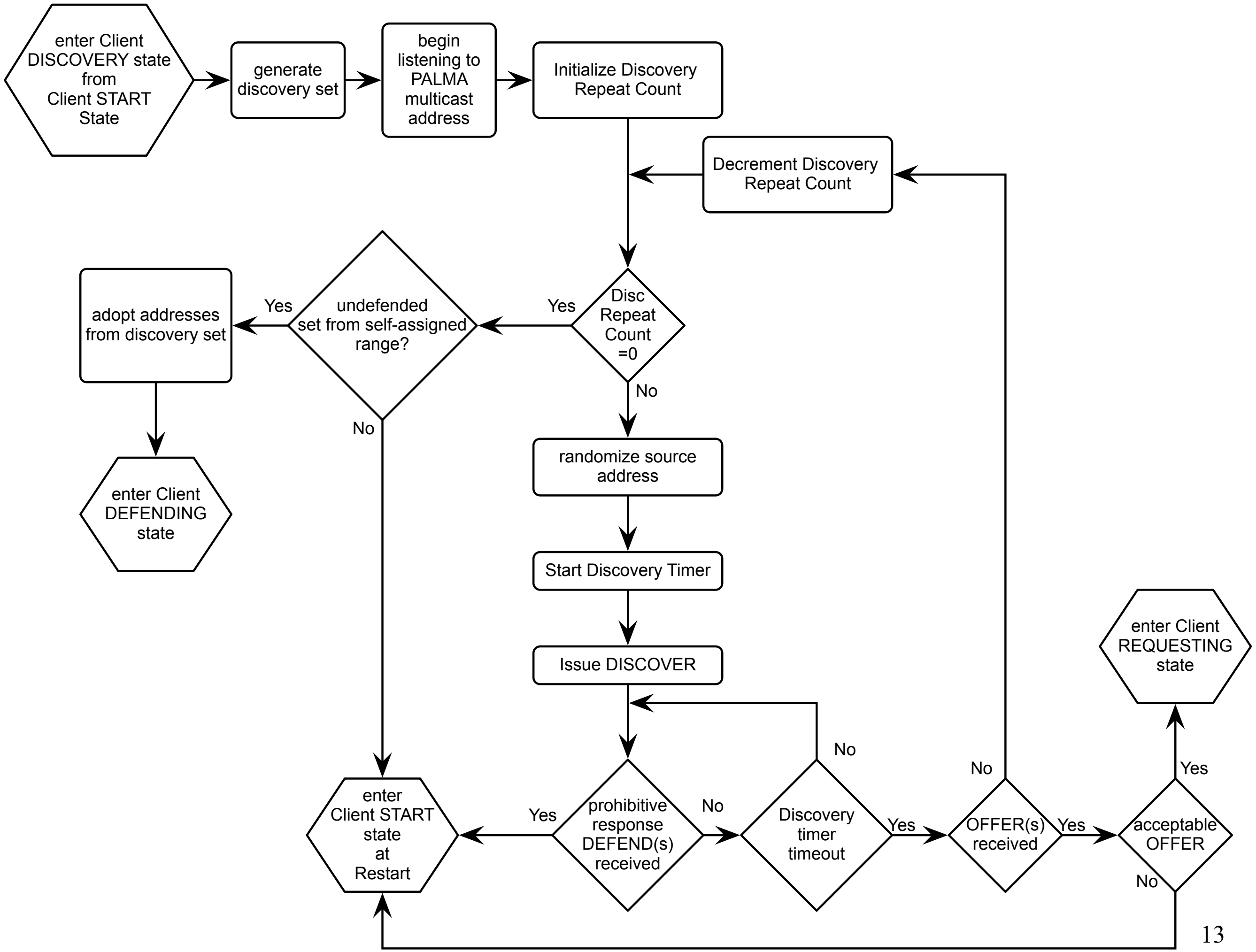
PALMA Client Flow Diagram: START State



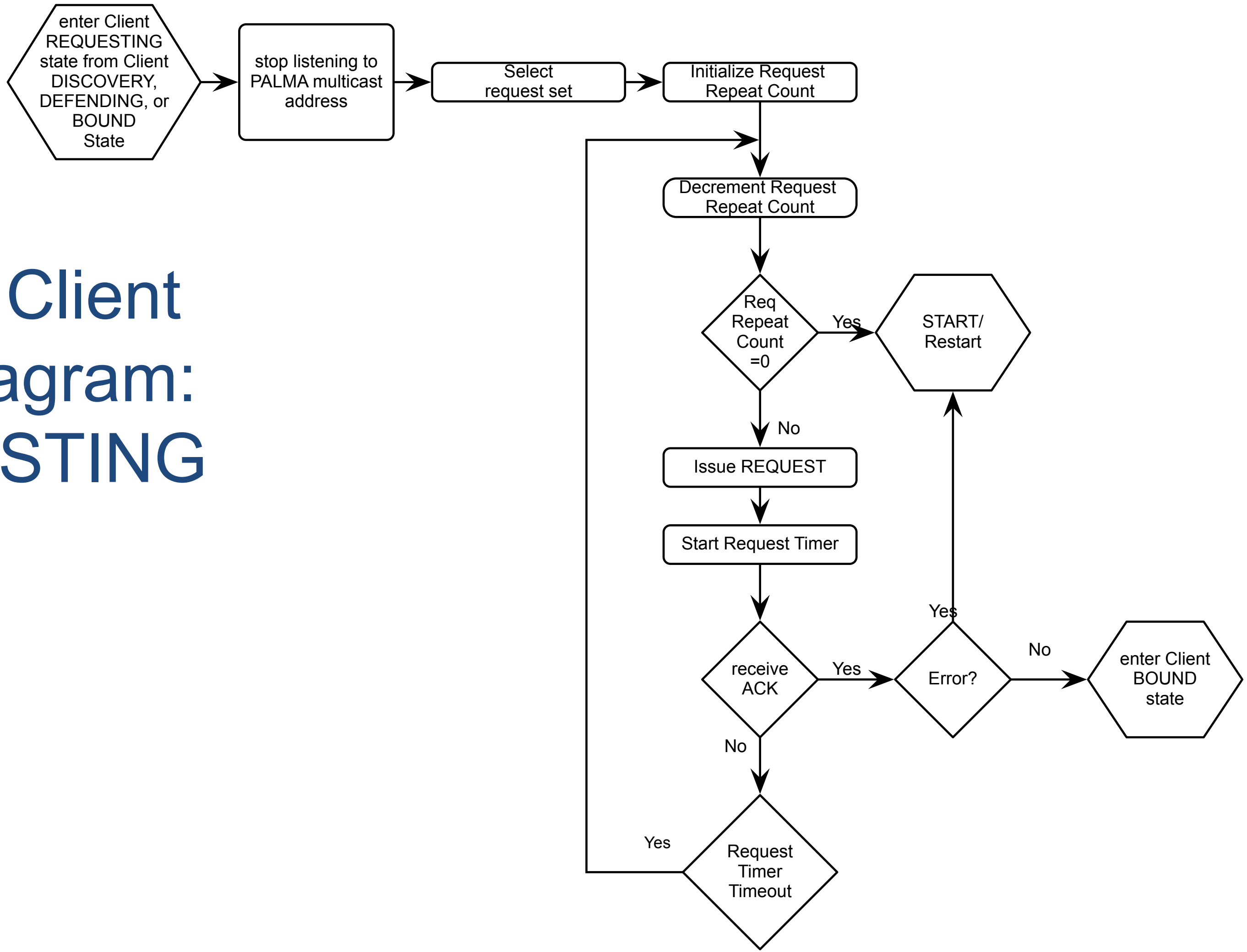
Bypass DISCOVERY when a server has been previously identified and stored.

Prior server assignment can be also be recalled from cold-storage for quick renewal or reactivation.

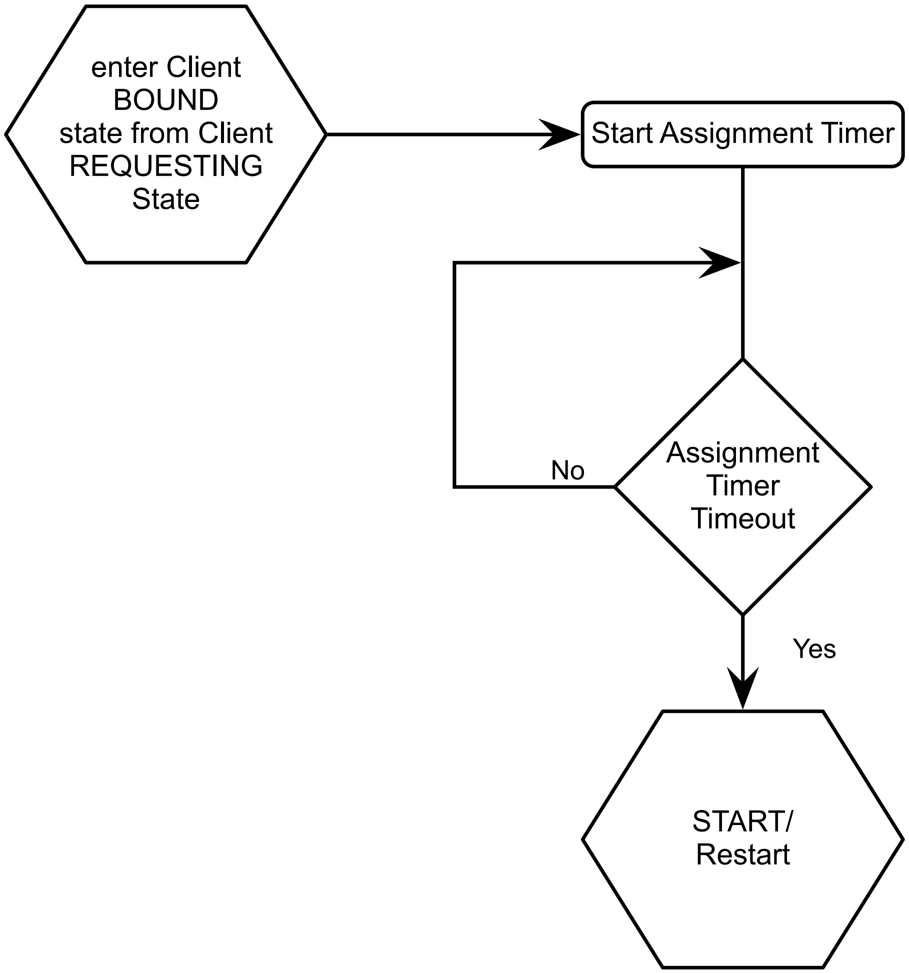
PALMA Client Flow Diagram: DISCOVERY State



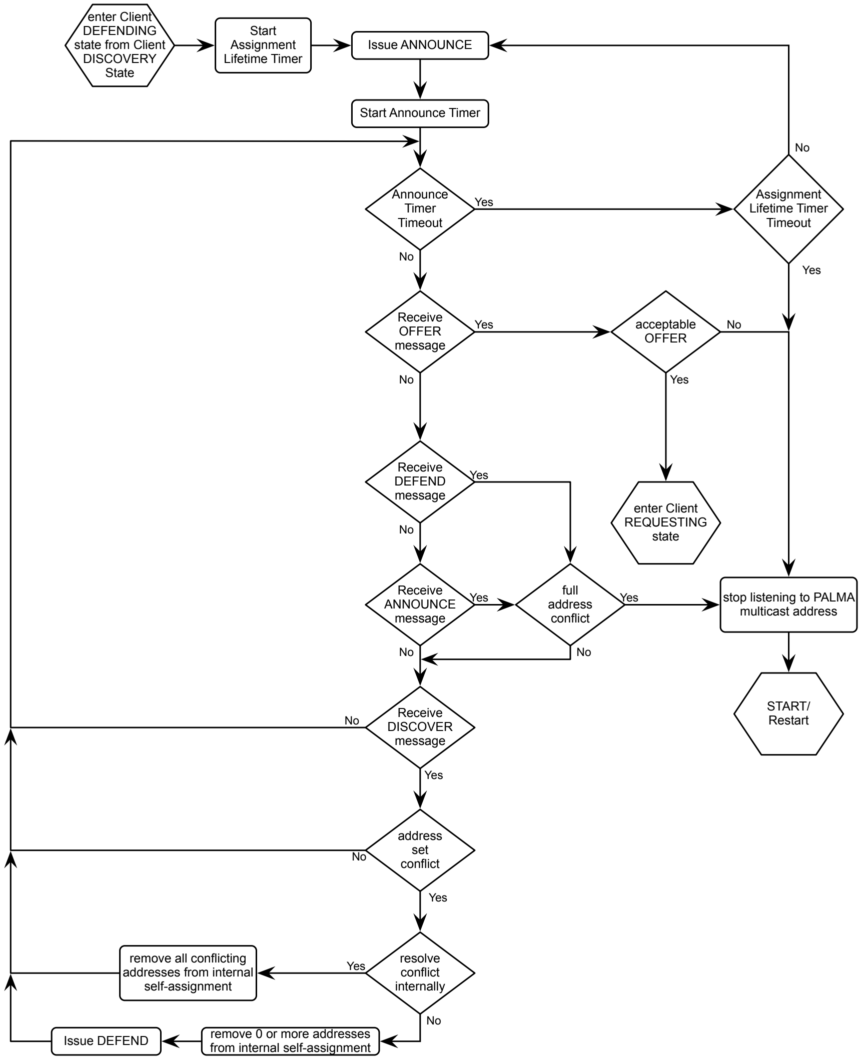
PALMA Client Flow Diagram: REQUESTING State



PALMA Client Flow Diagram: BOUND State



PALMA Client Flow Diagram: DEFENDING State



PALMA Client State Machine

	STATE				
event	**START**	**DISCOVERY**	**REQUESTING**	**BOUND**	**DEFENDING**
Restart!	continue[Begin!; stop_listen_mcast]		-x-	-x-	-x-
Begin!	known_server[State(REQUESTING/ Begin!); DISCOVERY/ DiscAddr!]	-x-	stop_listen_mcast init_req_count Start ack_timer sReq	update_set Start life_timer	Start life_timer Start ann_timer sAnn
DiscAddr!		generate_set increment_token init_dsc_count Start dsc_timer start_listen_mcast sDiscover	-x-	-x-	-x-
rOffer!	-x-	store_offer	-x-	-x-	accept_offer [State(REQUESTING/Begin!); State(START/Begin!)]
dsc_Timer!	-x-	accept_offer [State(REQUESTING/Begin!); dec_dsc_count, Start dsc_timer, sDiscover]	-x-	-x-	-x-
dsc_Count!	-x-	Stop dsc_timer delete_offer claimcheck[update_set, State(DEFENDING/Begin!); State(START/Restart!)]	-x-	-x-	-x-
rDef!	-x-	update_remote Stop dsc_timer State(START/Restart!)	-x-	-x-	update_remote check_set[compare_MAC[sDef; Stop ann_timer, State(START/Restart!)]]
rAnn!	-x-	update_remote check_set[Stop dsc_timer, State(START/Restart!)]	-x-	-x-	update_remote check_set[compare_MAC[sDef; Stop ann_timer, State(START/Restart!)]]
rDiscover!	-x-	check_set[Stop dsc_timer, State(START/Restart!)]	-x-	-x-	check_set[resolve[update_set; sDef]]
rAck!	-x-	-x-	accept_ack [State(BOUND/Begin!); State(START/Restart!)]	-x-	-x-
req_Count!	-x-	-x-	Stop ack_timer State(START/Restart!)	-x-	-x-
ann_Timer!	-x-	-x-	-x-	-x-	Start ann_timer sAnn
ack_Timer!	-x-	-x-	dec_req_count Start ack_timer sReq	-x-	-x-
life_Timer!	-x-	-x-	-x-	State(REQUESTING/ Begin!)	Stop ann_timer State(START/Restart!)
PortOp!	Begin!	Stop dsc_timer State(START/Restart!)	Stop ack_timer State(START/Restart!)	Stop life_timer State(START/ Restart!)	Stop ann_timer State(START/Restart!)
Release!	-x-	Stop dsc_timer State(START/Restart!)	Stop ack_timer State(START/Restart!)	sRel drop_set Stop life_timer State(START/ Restart!)	drop_set Stop ann_timer State(START/Restart!)

Next Steps

- Collect and address Task Group Ballot feedback
  - Please review and comment. All comments welcome.
- Recheck and refine logic
  - Carefully specify which fields are stored and recalled at each action
  - Working on a reference implementation of D0.5 to identify bugs
- Document specific processes for wireless LAN case
- Create a table or figure to show the relationship among address sets
- Complete some missing background sections
- Explain the IETF DHCPv6 draft on assigning MAC addresses