P802.1CQ/D0.6
Preview

Roger Marks
(EthAirNet Associates)
P802.1CQ Editor
2021-05-03

Summary

- P802.1CQ/D0.5 was reviewed in TG Ballot.
- Comment resolution was completed in November.
  - Address Blocks were introduced for address claiming
- In March, Editor presented “Block Address Registration and Claiming (BARC)”
  - cq-marks-BARC-0321-v00.pdf
  - address blocks used for registrar-managed addresses as well
  - Address Registration and Claiming (ARC)
    - address blocks, and also claiming address ranges using MAAP
      - presented also to IEEE 1722 Working Group
- Main issues raised in March concerned VLAN operation
- This contribution previews P802.1CQ/D0.6
  - refinements and details since March presentation
  - discussion on improved VLAN support

BARC assigns MAC Addresses in Address Blocks

1) Address Blocks (ABs) are sets of local addresses.
2) An AB includes equal-sized unicast and multicast address subblocks.
3) No BARC address falls within more than one AB.
4) An Address Block Designation (ABD) is a CABA or a RABI.
5) Claimable AB Address (CABA) is claimable by a Claimant without using a Registrar.
    - identifies Claimable Address Blocks (CABs) holding Claimable Addresses (CAs)
    - CABA is a multicast MAC address, not in any AB, and used as a DA.
6) RABI
    - identifies a Registrable Address Block (RAB) holding Registrable Addresses (RAs)
    - RABIs are held in the inventory of a Registrar for assignment
        - may be assigned to Claimants
        - may be claimed by Registrants
        - more detail in a future presentation
7) A large set of Temporary Unicast Addresses (TUAs) is specified
    - useful for initial discovery by Claimant lacking a unicast address

MAC Address Categorization

determinable via inspection:	**Expanded name**	**I/G**	**indicates, by inspection**
CA	claimable address, in claimable address block (CAB)	U,M	CABA, CABA type, CAB (including all other CAs in CAB)
CABA	CAB address	M	CABA type, CAB
RA	registrable address, in registrable address block (RAB)	U,M	ABI, ABI type, all other RAs in RAB
TUA	temporary unicast address	U	note: ~6.9E10 to choose among

BARC MAC Address Structure

N11	r	i	j	k	for registrable addresses, r=1; for claimable addresses, r=0
N10	1	1	1	m	m is the usual multicast (I/G) bit; 111 is local “SAI” range per IEEE Std 802c
N9					0000 for claimable addresses
N8					
N7					
N6					
N5					
N6					
N5					
N2					
N1					
N0					

12 nibbles per 48-bit address

- address block includes subblocks of
  - 16^{jk} claimable addresses, or
  - 16^{jk} registrable addresses (or aggregated into larger blocks)
- for claimable addresses, *i* distinguishes
  - Claimable Addresses (CAs) from
  - CABAs
    - identifiers that are also used as addresses

	r	i	jk	m
RA	1	ABI Type		I/G
CA	0	1	CABA Type	I/G
CABA	0	0	CABA Type	1
TUA	0	0	0	0

CABA and CA, CABA Type 0-3

CABA Type 0

CABA	CAB
0 0 0 0	0 1 0 0
1 1 1 1	1 1 1 *
0 0 0 0	0 0 0 0
X8	X8
X7	X7
X6	X6
X5	X5
X4	X4
X3	X3
X2	X2
X1	X1
X0	X0

CABA Type 1

CABA	CAB
0 0 0 1	0 1 0 1
1 1 1 1	1 1 1 *
0 0 0 0	0 0 0 0
X8	X8
X7	X7
X6	X6
X5	X5
X4	X4
X3	X3
X2	X2
X1	X1
0	*

CABA Type 2

CABA	CAB
0 0 1 0	0 1 1 0
1 1 1 1	1 1 1 *
0 0 0 0	0 0 0 0
X8	X8
X7	X7
X6	X6
X5	X5
X4	X4
X3	X3
X2	X2
0	*
0	*

CABA Type 3

CABA	CAB
0 0 1 1	0 1 1 1
1 1 1 1	1 1 1 *
0 0 0 0	0 0 0 0
X8	X8
X7	X7
X6	X6
X5	X5
X4	X4
X3	X3
0	*
0	*
0	*

2 contiguous subblocks per CABA (one unicast, one multicast)

- 6.9E10 Type 0 CABAs
- 1 CA/subblock

- 4.3E9 Type 1 CABAs
- 16 CAs/subblock

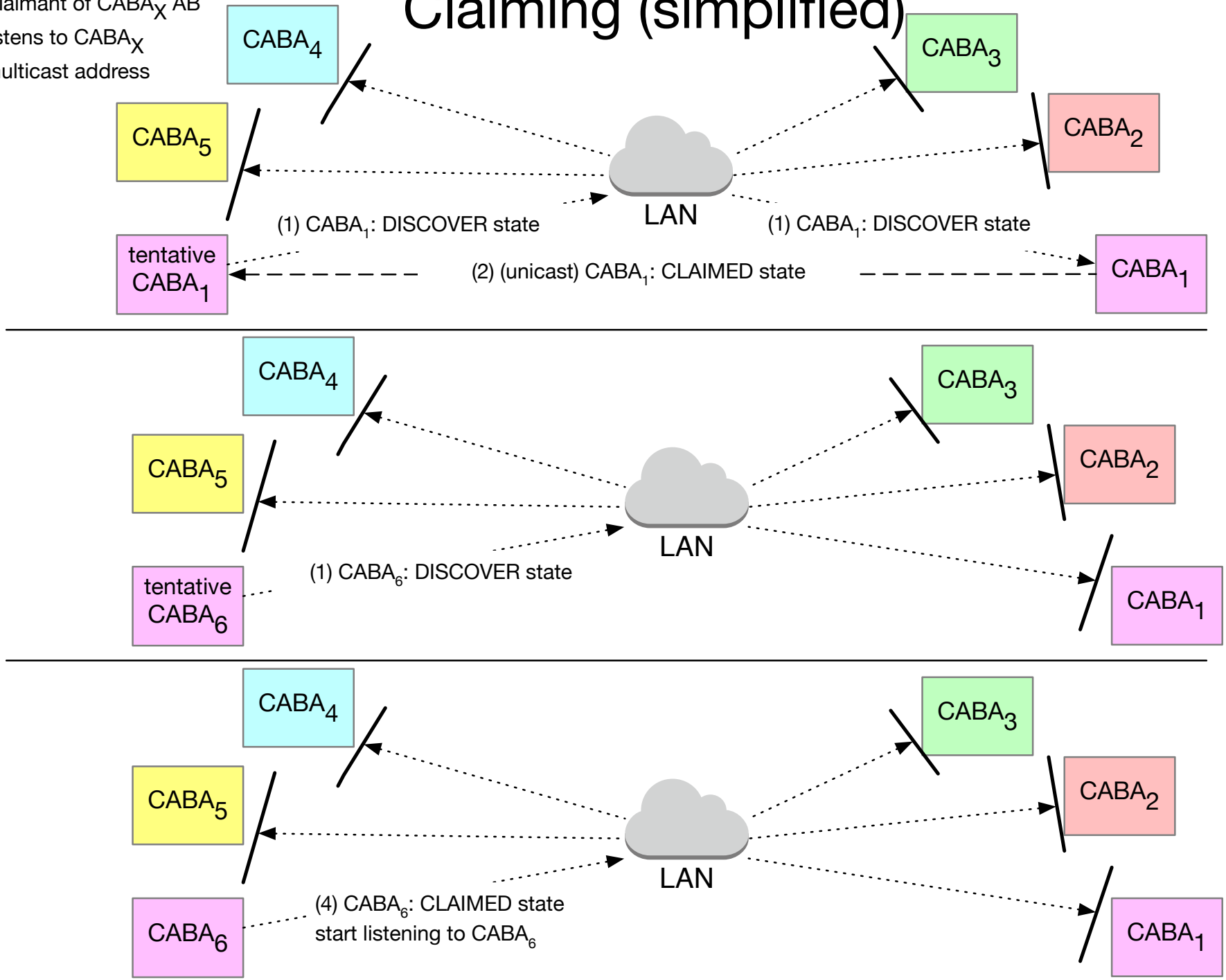
- 2.7E8 Type 2 CABAs
- 256 CA/subblock

- 1.7E7 Type 3 CABAs
- 4096 CAs/subblock

* indicates wildcard (any value)

Claiming (simplified)

Claimant of $CABA_X$ AB listens to $CABA_X$ multicast address



Registrar

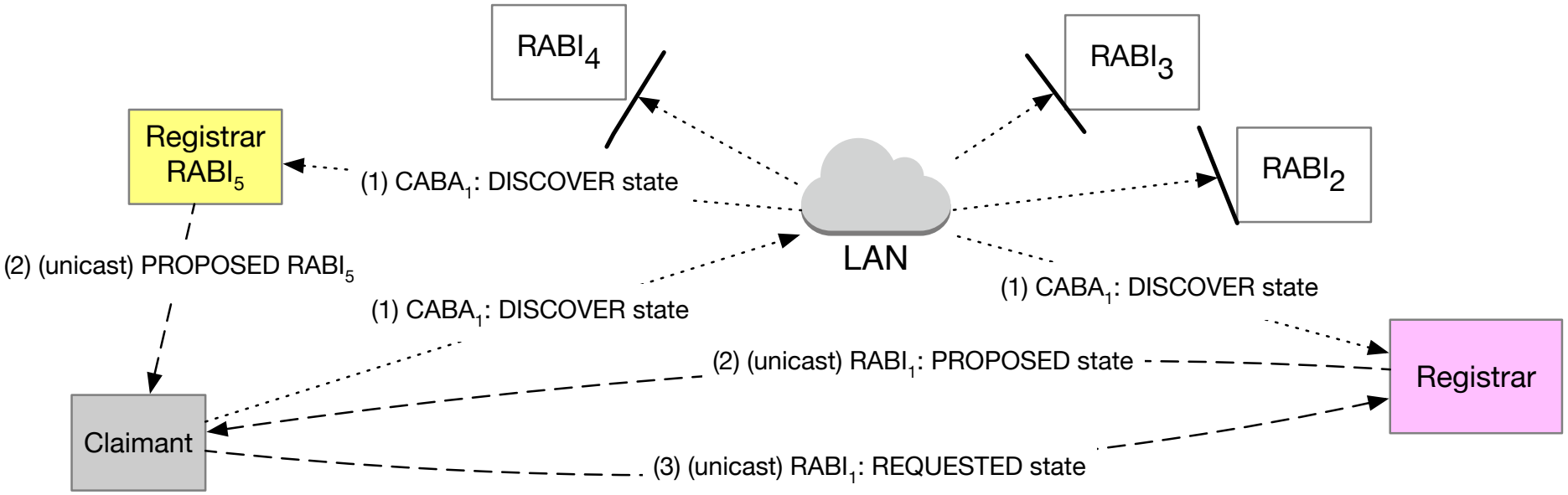
- Claimant need not be aware of Registrar when initiating a claim.

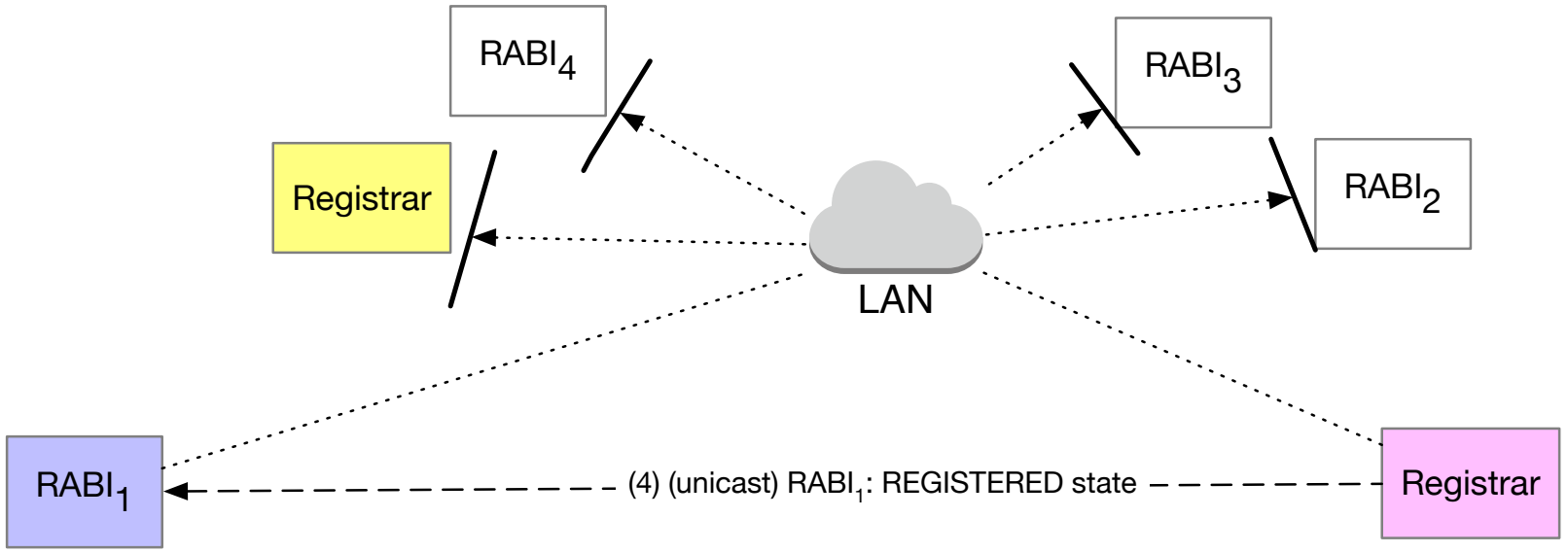
- Registrar maintains an inventory of RABIs.
  - a protocol specifies how Registrars acquire RABIs.
  - set of RABs is disjoint from the set of CABs
    - AB is either claimable (CAB) or registrable (RAB); not both

- Registrar listens for all messages to a CABA.
  - r=0, i=0, m=1, i.e. DA begins 00**-1111
    - [MMRP NumberOfValues field is 13 bits]

- Registrar can respond to a DISCOVER with a proposal of a RABI in its inventory.
  - The proposal can also defend the DISCOVER's CABA.
  - Registrar confirms registration of request.

Operation with Registrars





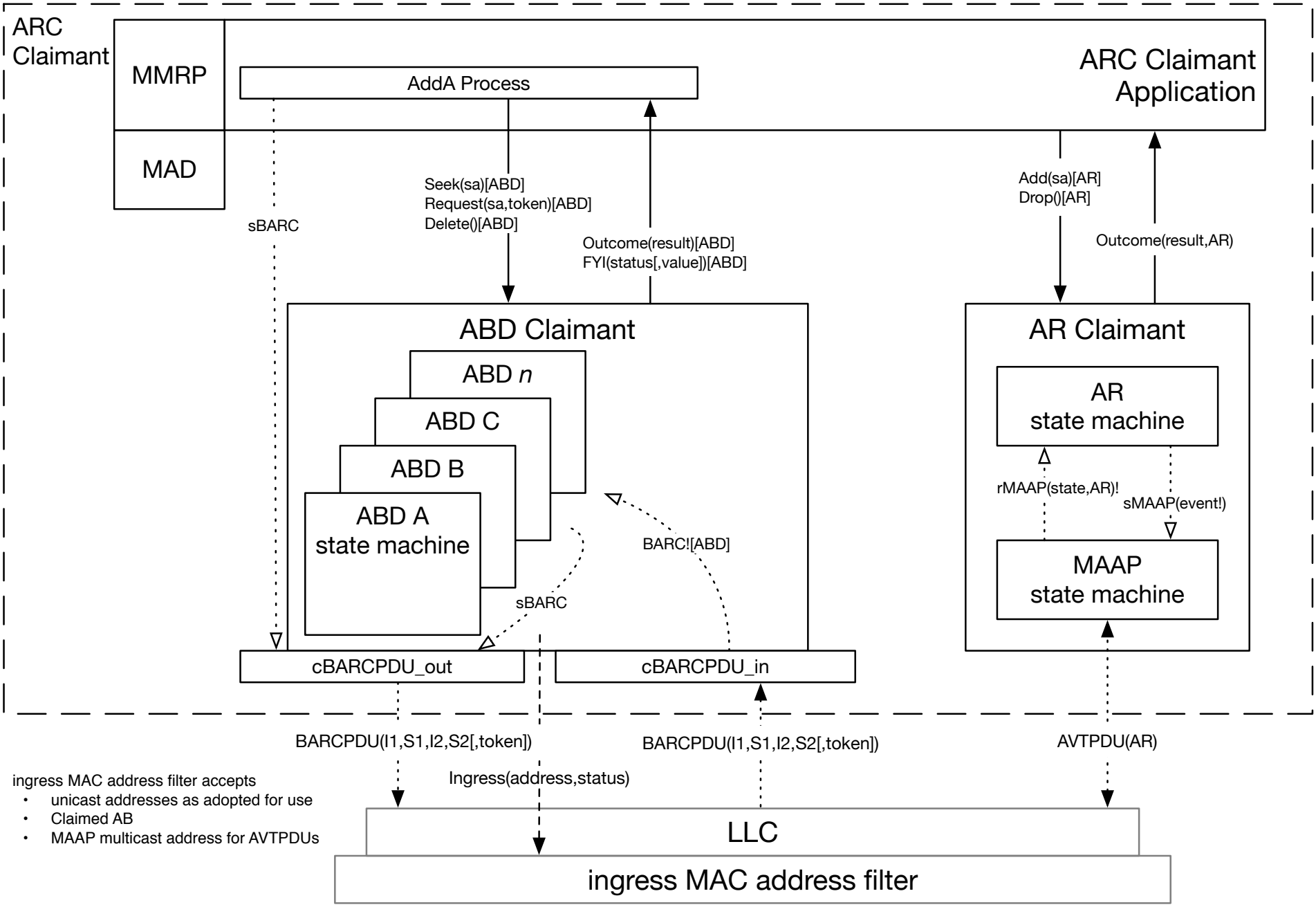
BARC Design

- A BARC architecture follows, with details including state machines.
  - additional details in Appendix

- BARC (Block Address Registration and Claiming) is put into the broader context of Address Registration and Claiming (ARC), which supports both:
  - address blocks (ABs), identified by Address Block Identifiers (ABIs)
  - address ranges (ARs), excluding addresses specified by BARC

- ARC is the general protocol
  - BARC handles ABI Registration and CABA Claiming
  - existing MAAP handles AR Claiming

ARC Architecture – ARC Claimant



ingress MAC address filter accepts

- unicast addresses as adopted for use
- Claimed AB
- MAAP multicast address for AVTPDUs

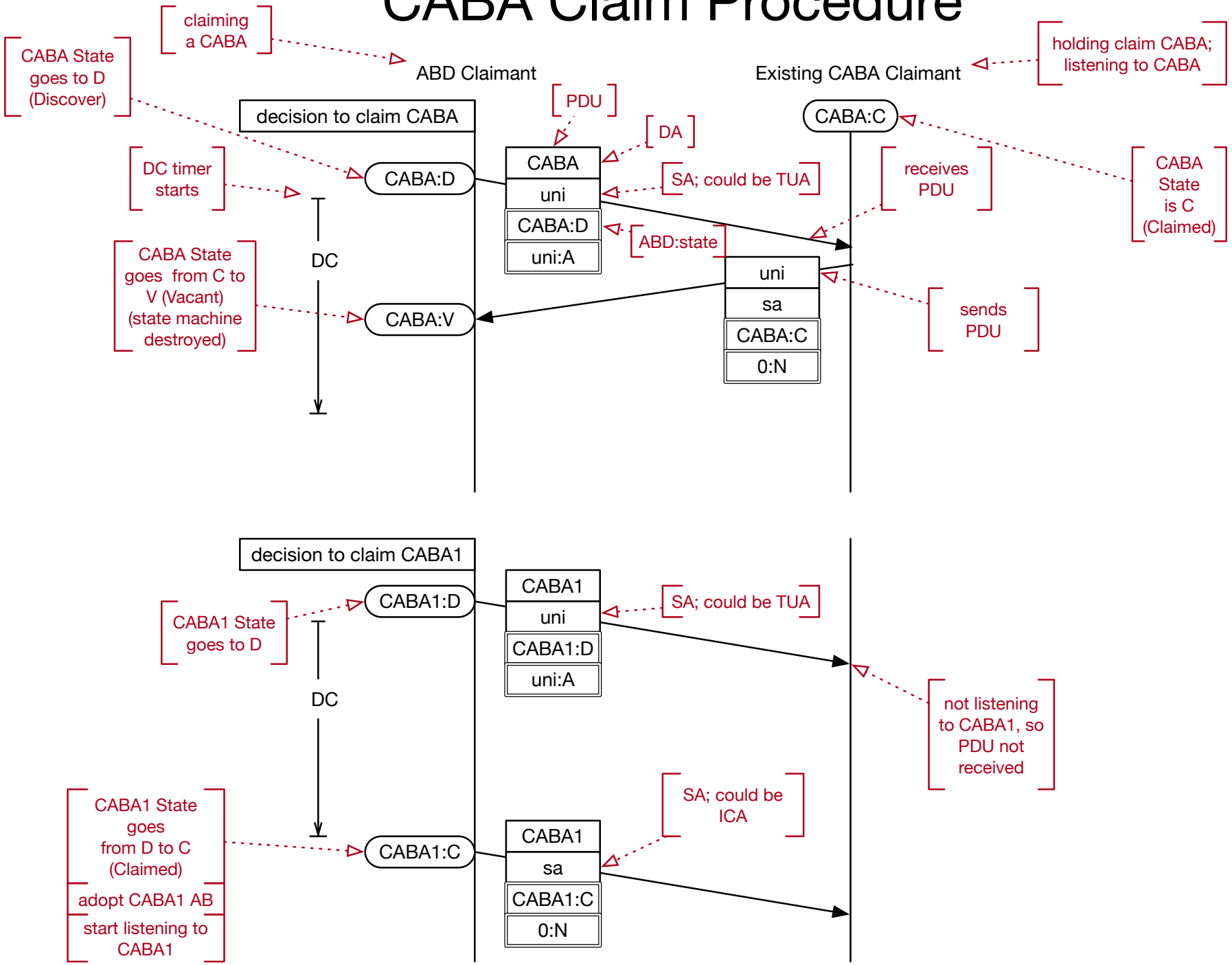
BARCPDU Summary

field name	purpose	content
DA	dest addresss	DA
SA	source address	
E	Ethertype	[tbd; could be 22F0 (MAAP Ethertype)]
t	subtype	[tbd, per 1722 WG; see IEEE 1722 Table 6]
S1	State	D (Discover), C (Claimed), V (Vacant), R (Registered), pr (preregistration), A (address), RD, RC, RV, RX, N(null)
I1	identifier	48-bit address or ABI
S2	State	P (proposed), A (address), N (null)
I2	identifier	48-bit address or ABI
T	token	length tbd

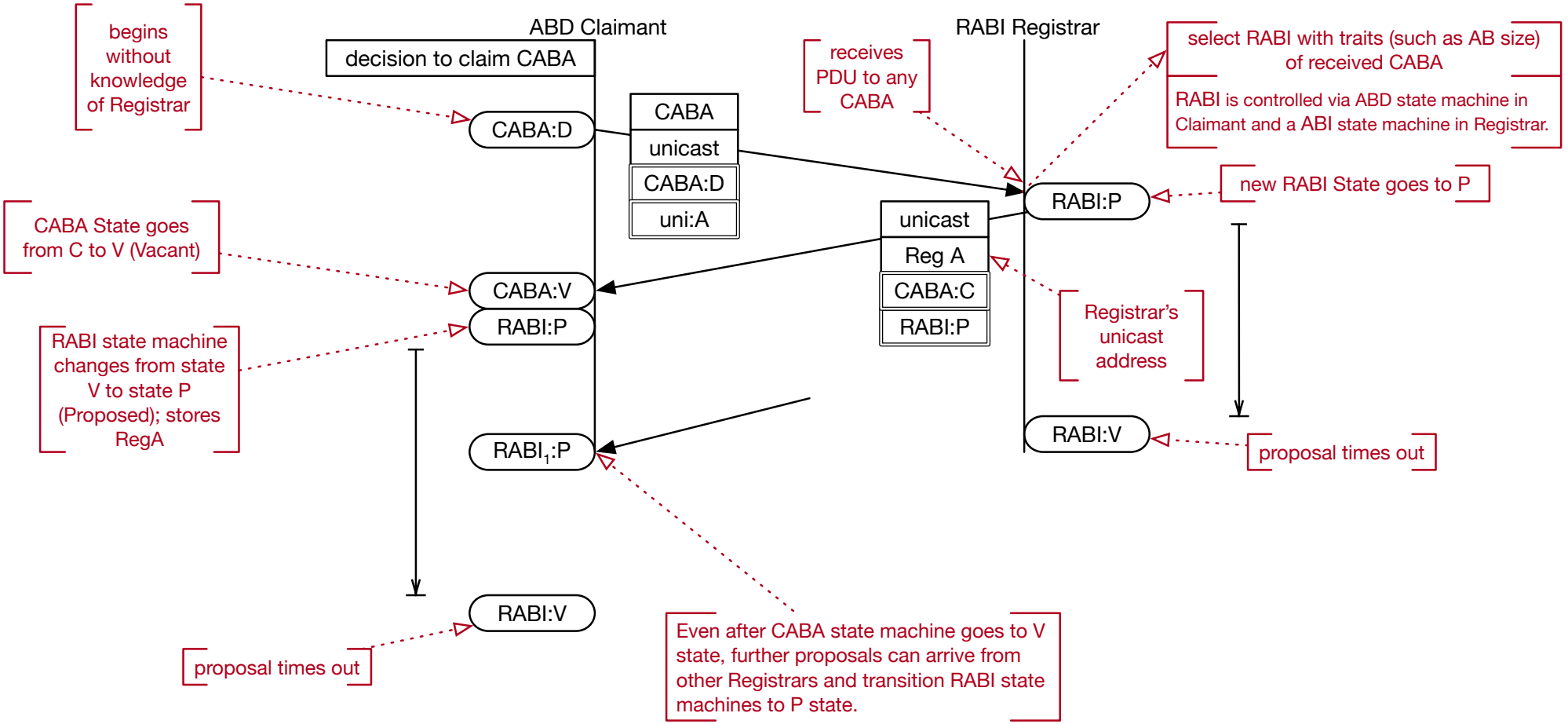
AVTPDU Summary

field name	purpose	content
DA	dest addresss	91:E0:F0:00:FF:00 for MAAP multicast
E	Ethertype	22F0 (MAAP Ethertype)
t	subtype	FE per IEEE 1722 Table 6

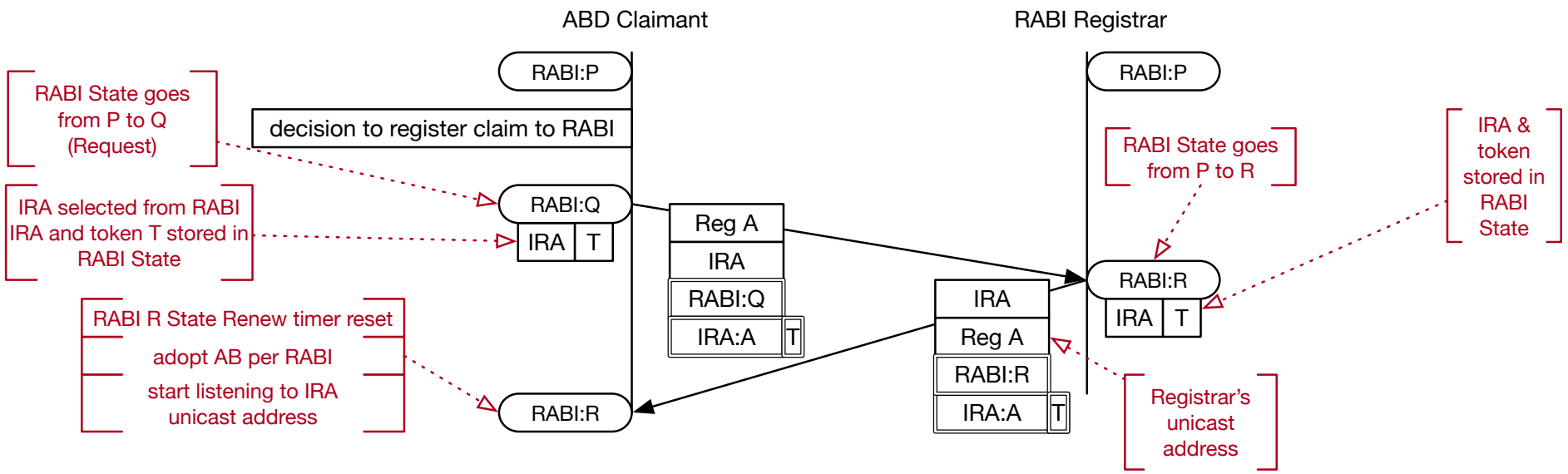
CABA Claim Procedure



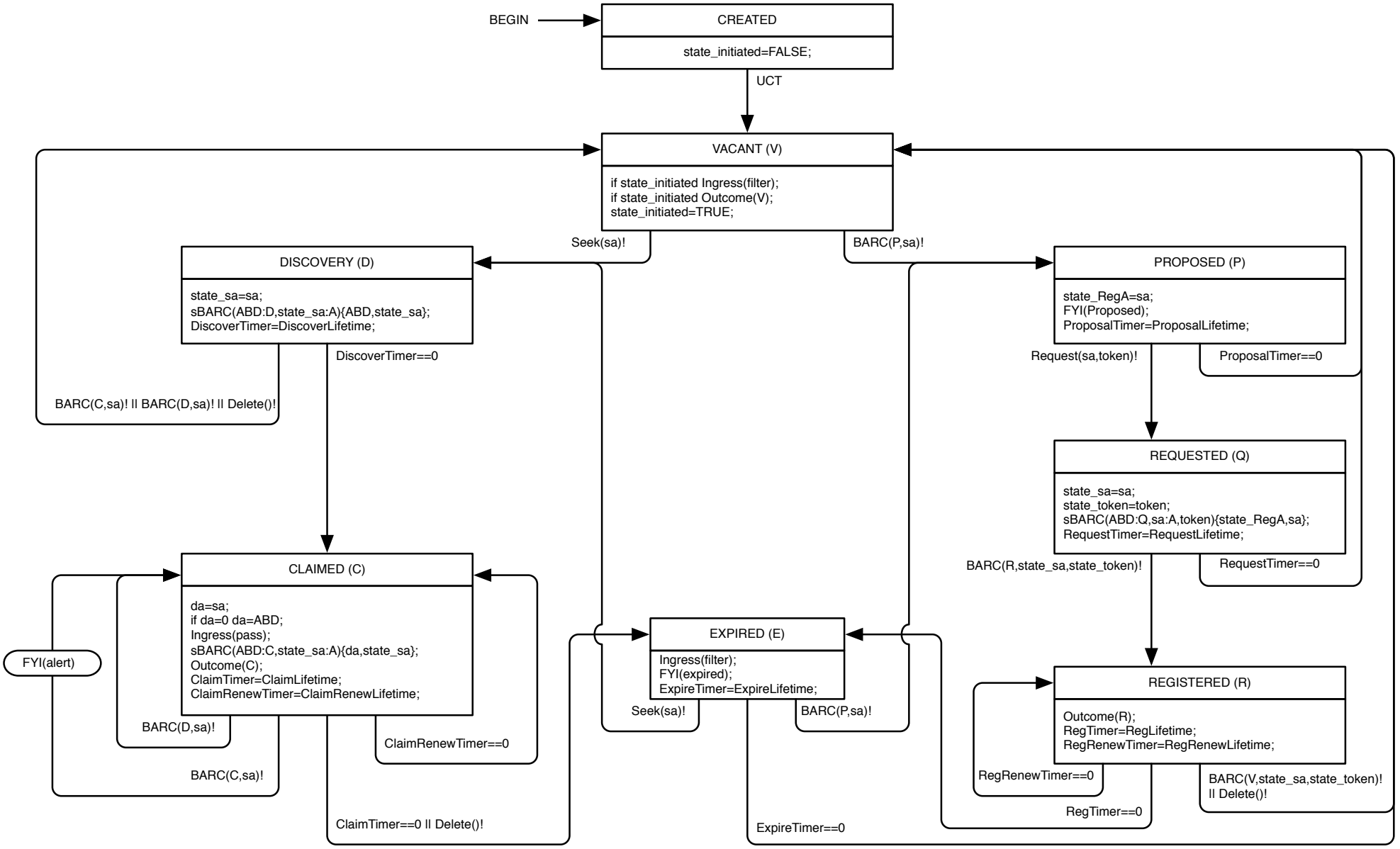
ABD Claimant/Registrar Procedure



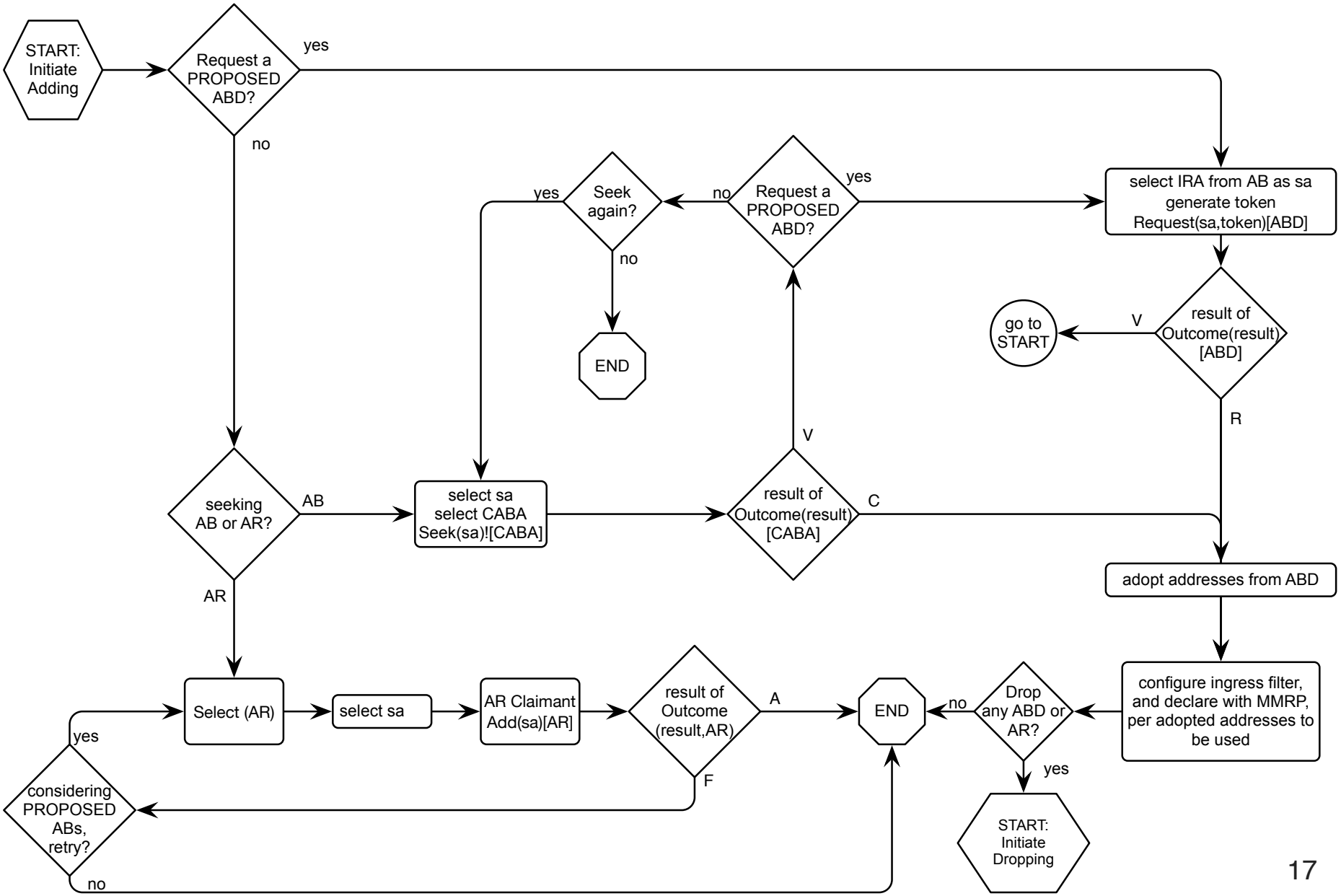
RABI Registration



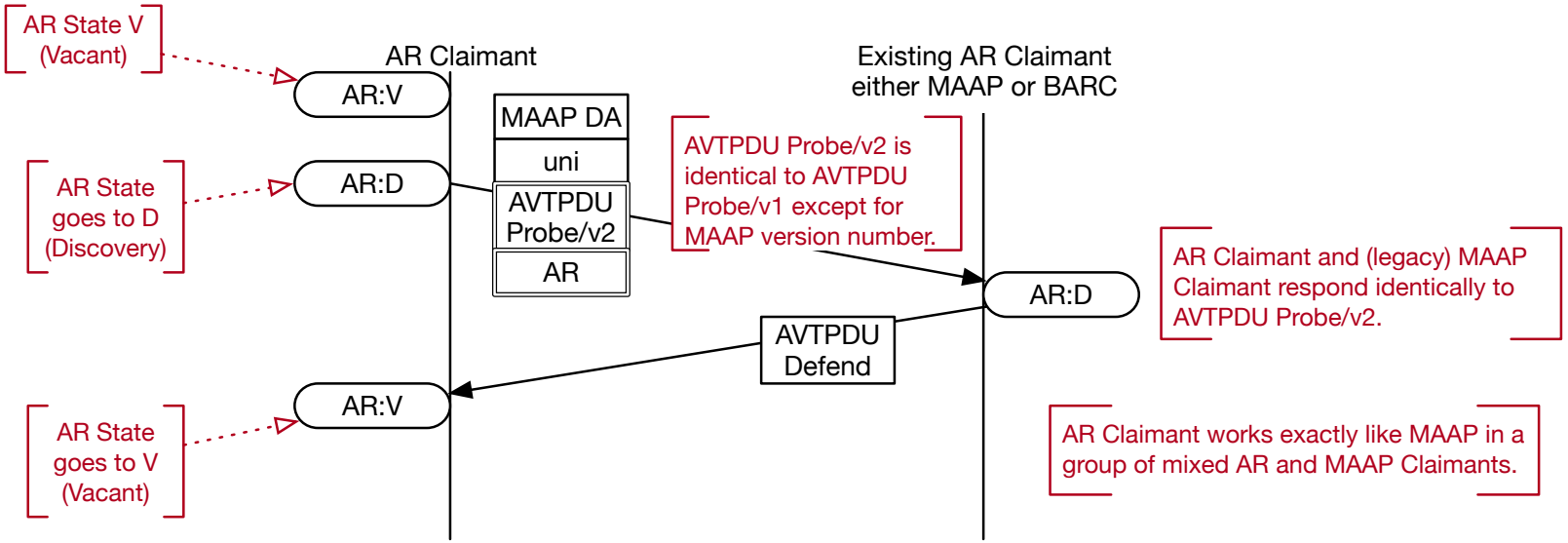
ABD Claimant: ABD State Machine

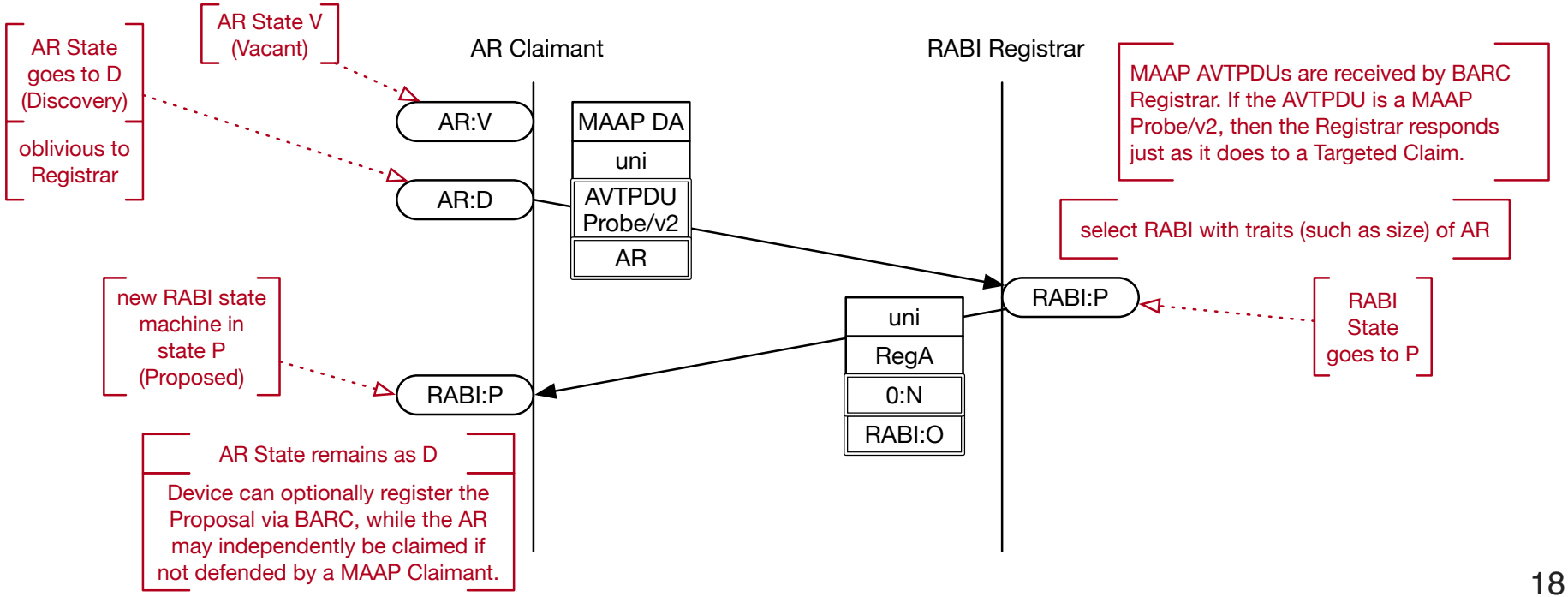


ARC Claimant Application Process: AddA

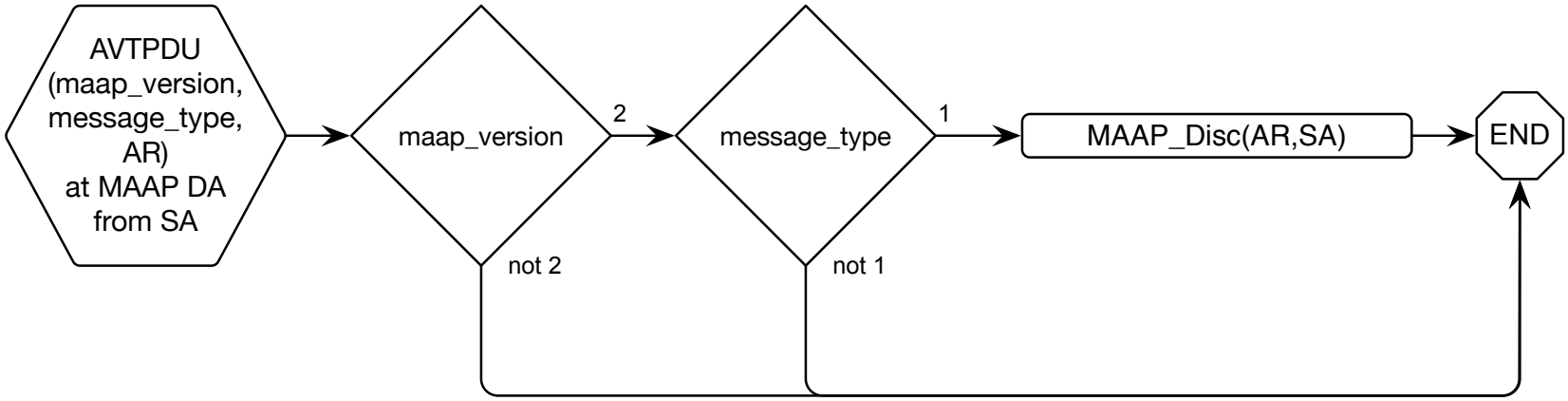


AR Claimant Procedure





BARC Registrar: AVTPDU Processor



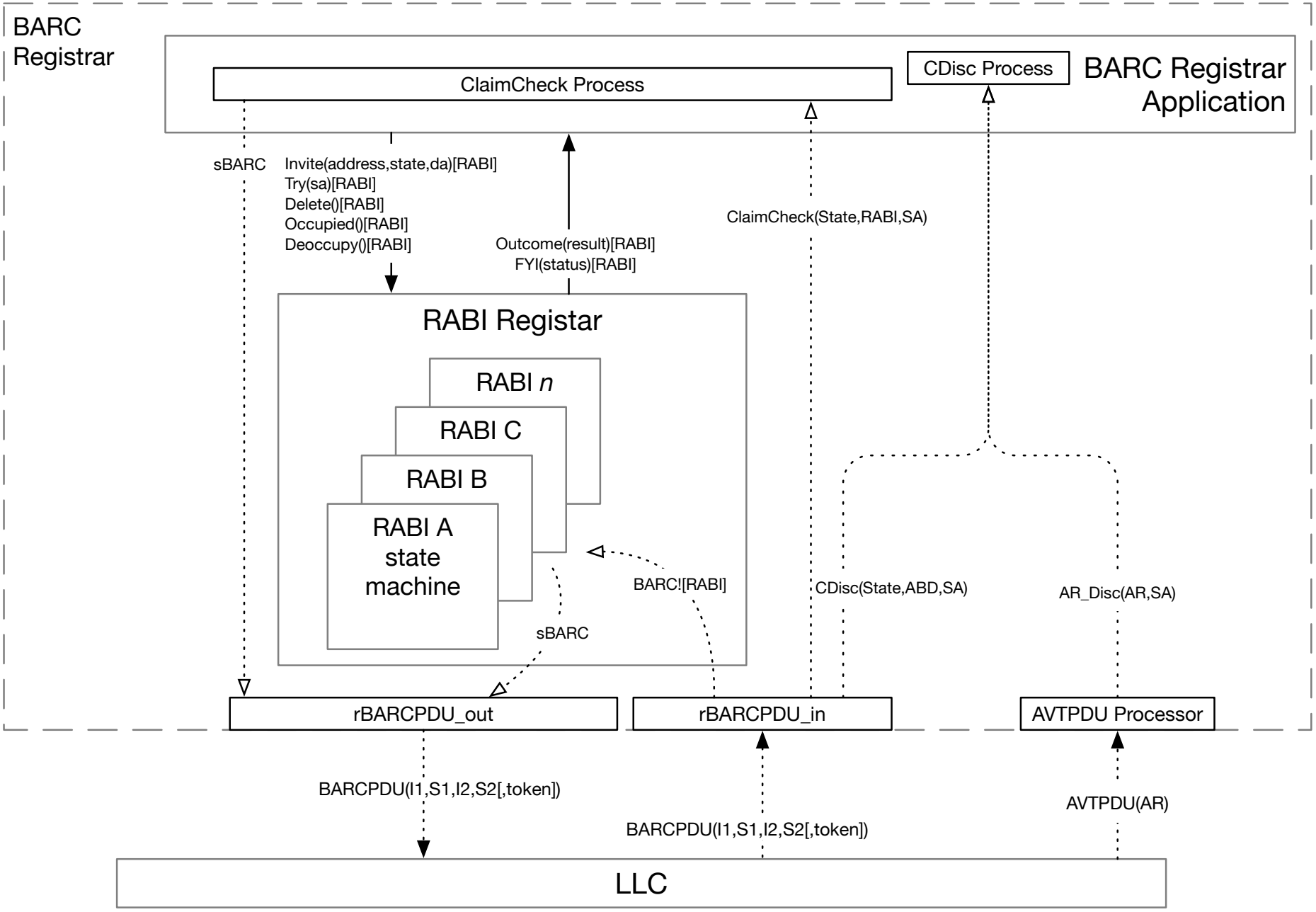
AR State Transition Table

	State		
Event	**VACANT (V)**	**DISCOVERY (D)**	**ACQUIRED (A)**
Add(sa)!	sMAAP(Begin(AR,sa)!) **DISCOVERY**		
rMAAP(AR:Defend)!		Outcome(A,AR) **ACQUIRED**	
rMAAP(AR:Initial)!		Outcome(F,AR) **VACANT**	Outcome(X)[AR] **VACANT**

rMAAP(AR:State!) invokes an event at the state machine when the MAAP state changes to State

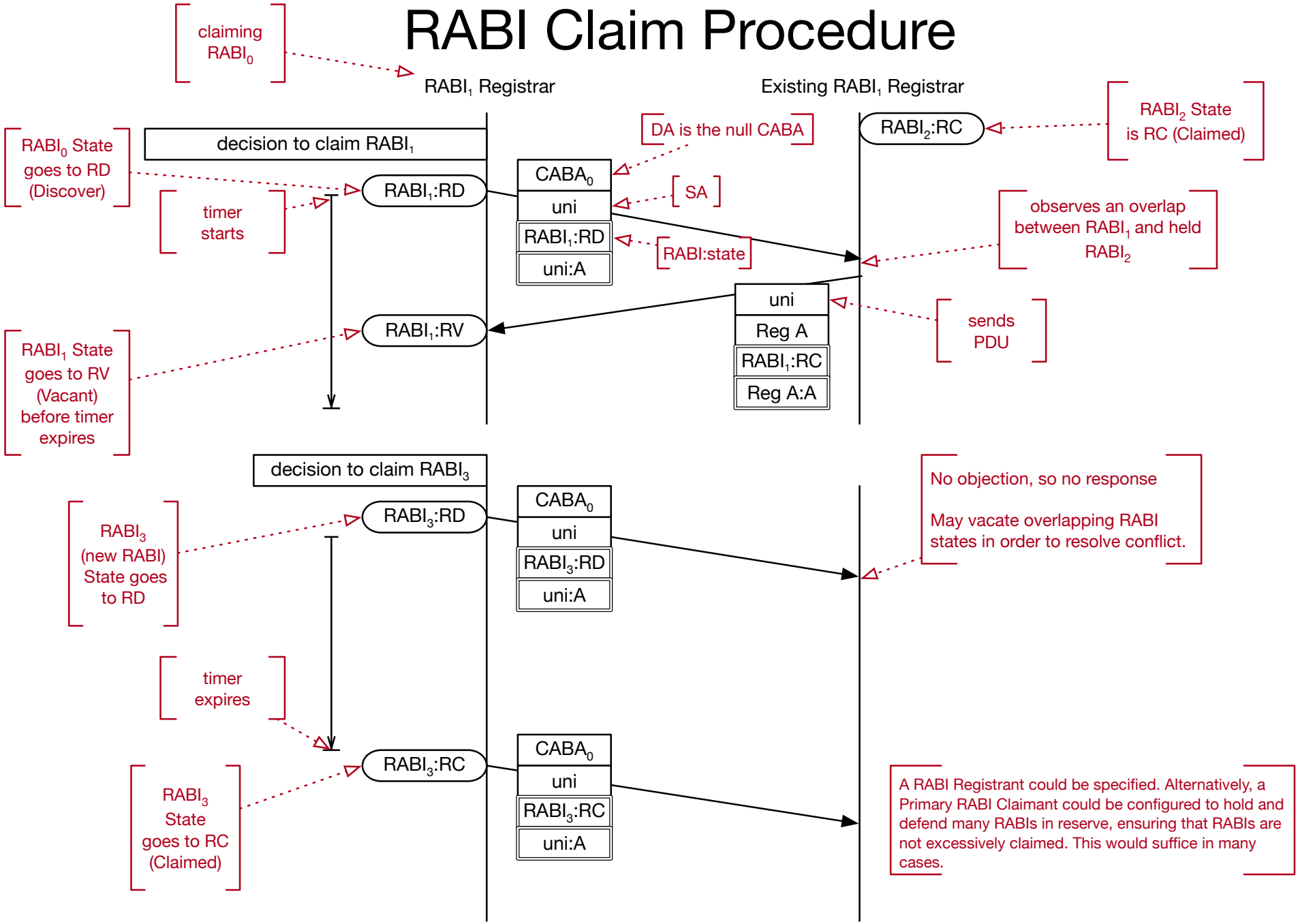
sMAAP(Action!) invokes Action! event at MAAP state machine

BARC Architecture – Registrar

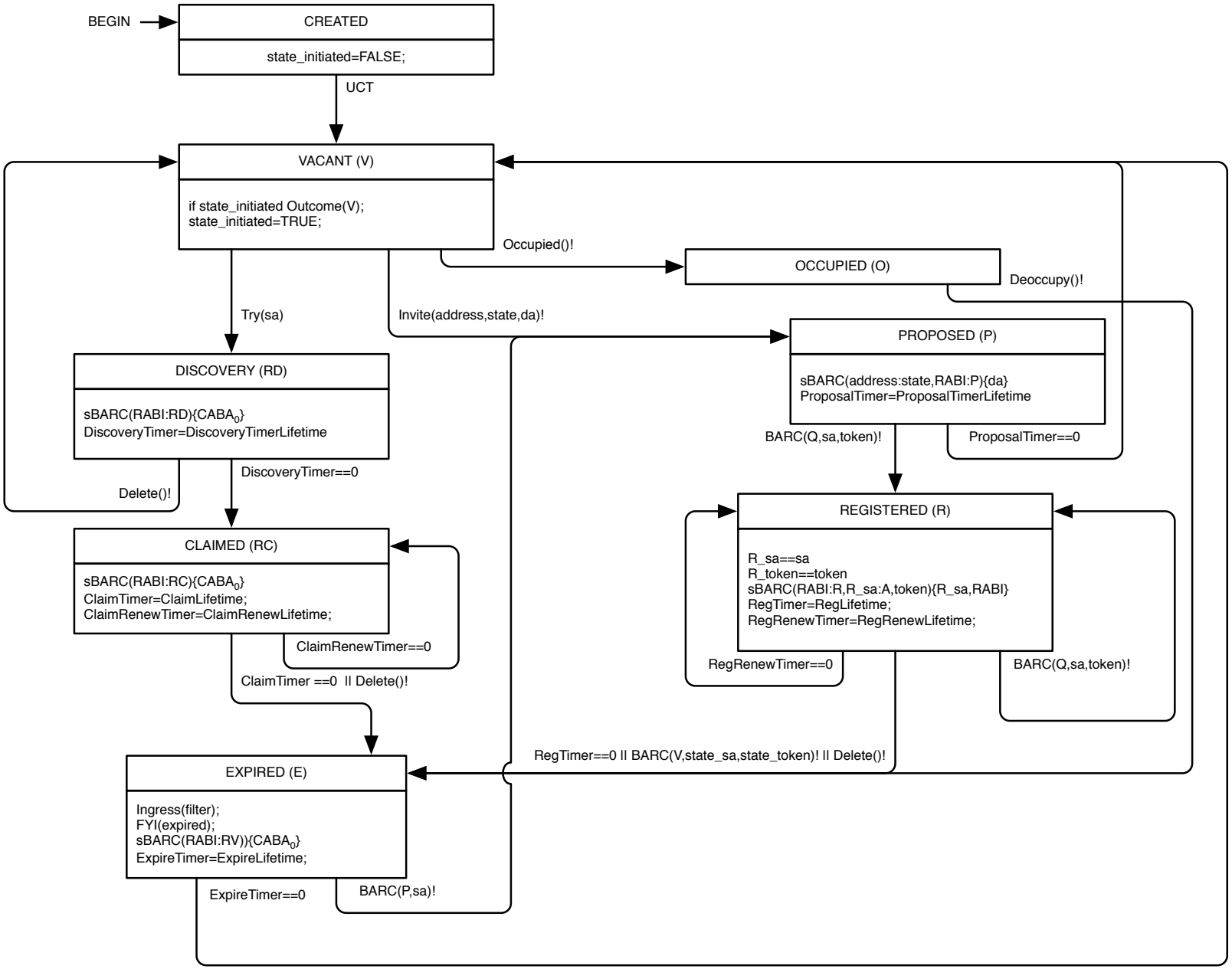


ingress MAC address filter accepts BARCPDUs addressed to any CABA
and MAAP multicast address for AVTPDUs

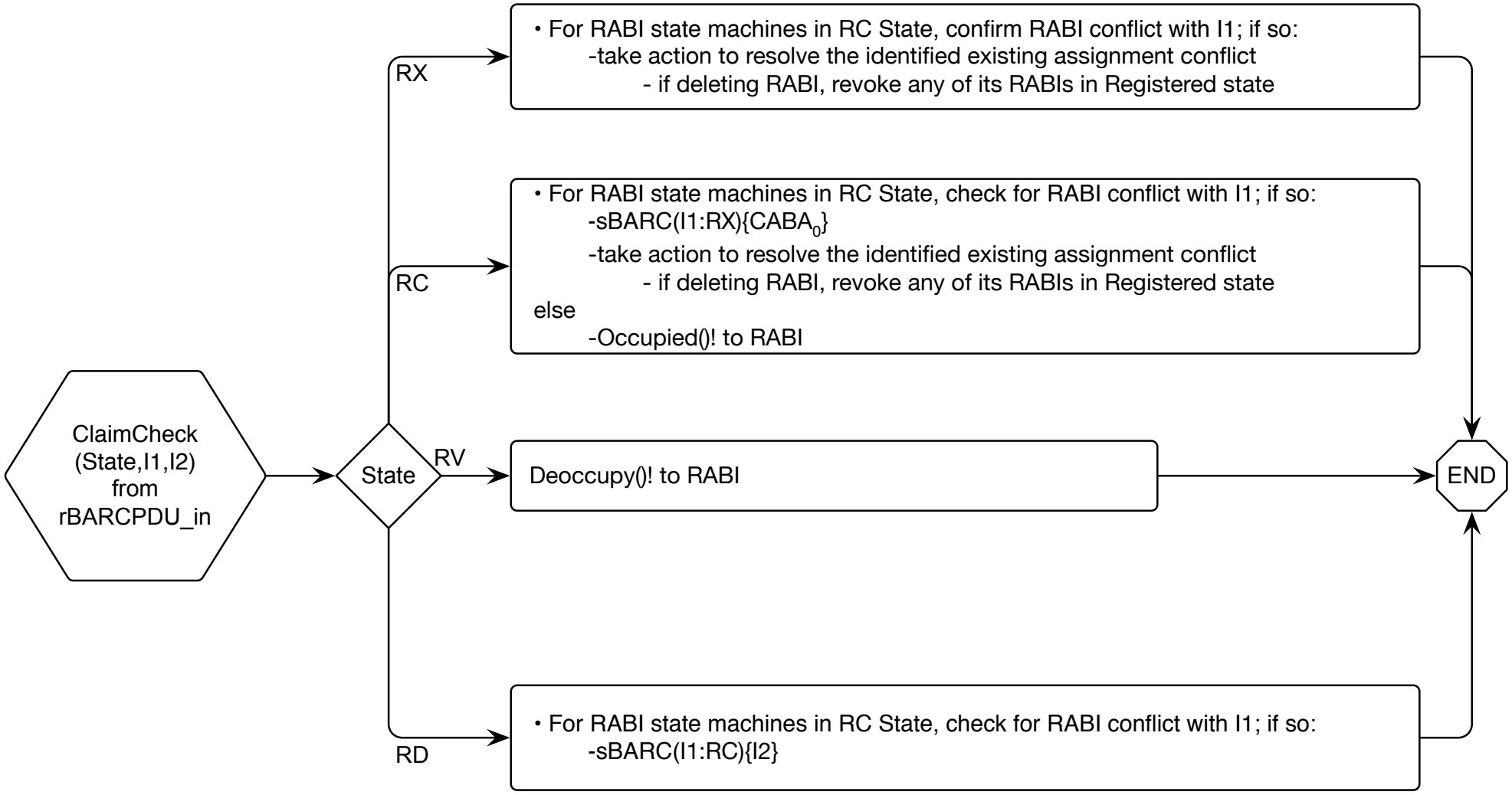
RABI Claim Procedure



Registrar: RABI State Machine



BARC Registrar Application: ClaimCheck Process



VLANs

• All address assignments are specific to the VLAN (or VLANs) in which messaging is communicated and under which the assignment was completed.

• Usage of any address may need to be limited to the VLAN or VLANs under which it was obtained.

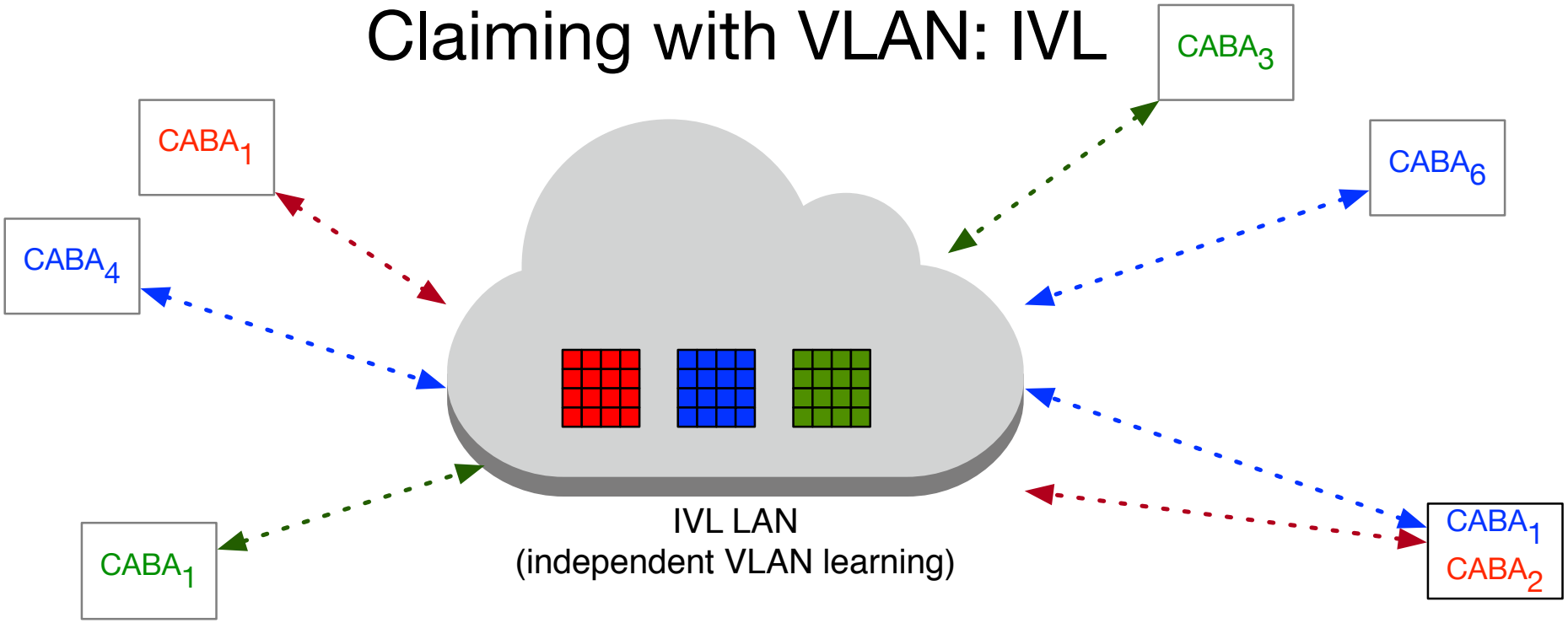
• Due to the possibility that the same unicast address may be assigned in different VLANs, Independent VLAN Learning (IVL) may be needed in bridges, per IEEE Std 802.1Q Annex F (F.1.2).

- This requirement could be relaxed in some cases
- e.g. when assigned unicast addresses are declared via MMRP (instead of learning)

• This issue is common to claiming protocols generally.

• Some approaches follow.

Claiming with VLAN: IVL



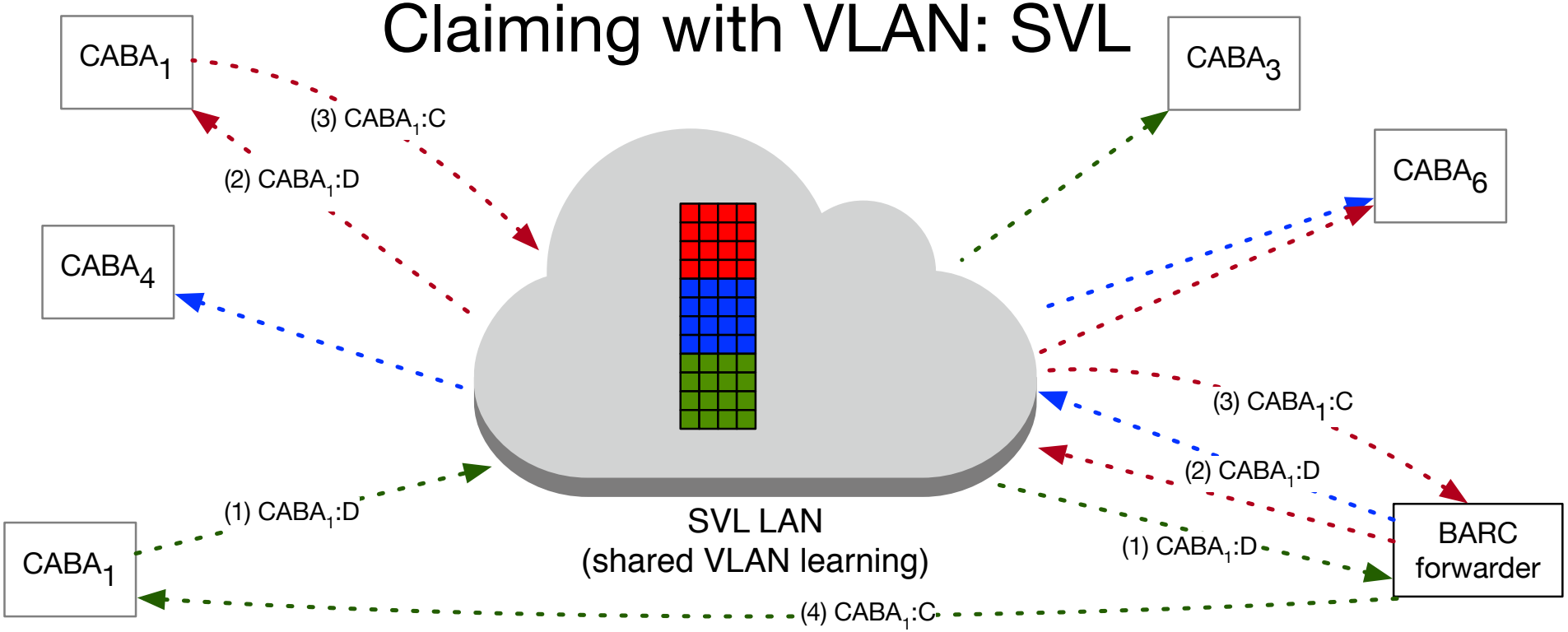
IEEE Std 802-2014 says “Local MAC addresses need to be unique on a LAN or bridged LAN unless the bridges support VLANs with independent learning.”

With IVL, each VLAN has an independent forwarding table.
- -but IVL is not always possible

BARC claiming on each VLAN is independent
- a duplicate address may occur in more than one VLAN; that is not harmful if managed carefully

A claimant with multiple VLANs needs to claim in each VLAN.
Claimed address is usable only in claimed VLAN:
- Claimant needs to bind address to VLAN
- For convenience, Claimant may claim the same address in each of its VLANs
  - -Still, requires multiple claim messages and multiple forwarding table entries.
  - -Device needing many VLANs should consider an EUI

Claiming with VLAN: SVL



With SVL, VLANs share a forwarding table.

BARC claiming on each VLAN is independent

an address could become a duplicate, existing in more than one VLAN

forwarding table is limited to one entry per address, so duplication is catastrophic.

To prevent duplication, BARC messaging must be carried across VLAN boundaries.

-extra benefit: single address claim is valid over multiple VLANs

To carry claiming messages across VLAN boundaries, a BARC forwarder could be introduced.

-receives BARC messages on all VLANs

-forwards BARC multicast to all VLANs (retaining originating source address)

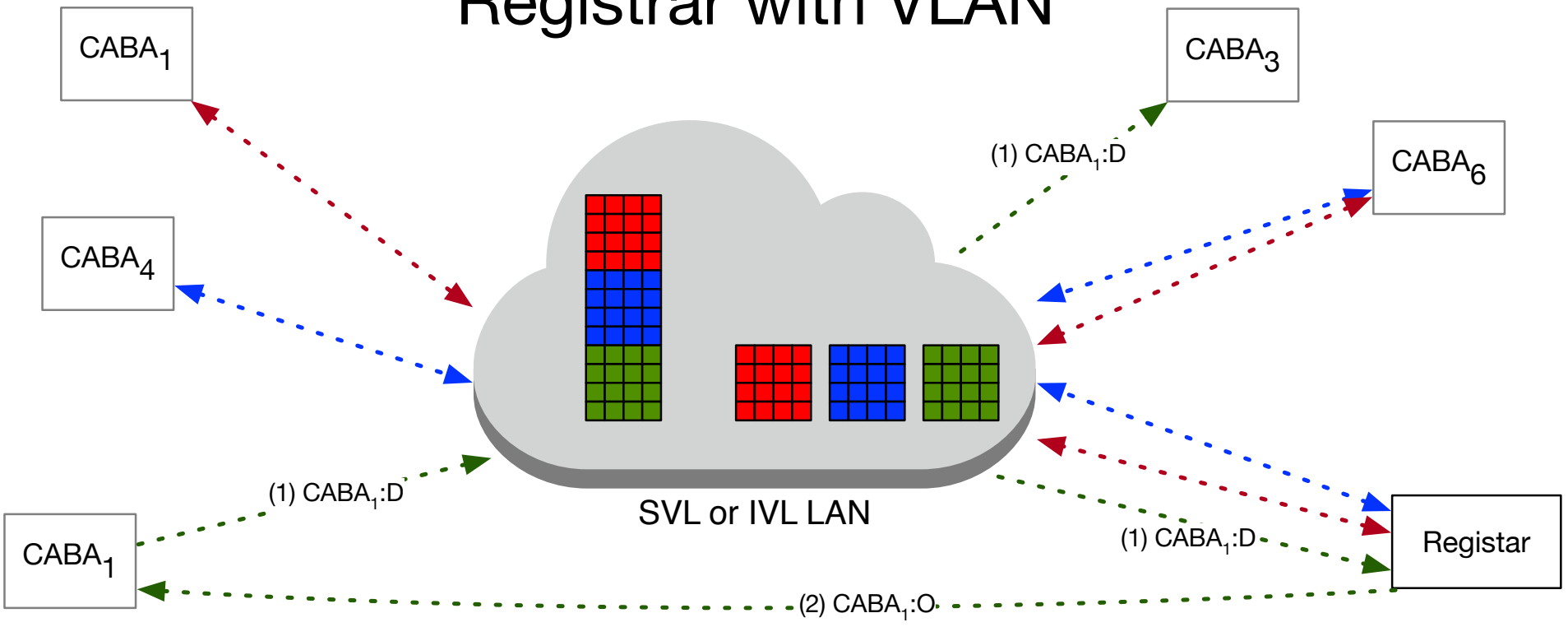
-forwards any BARC claim response to the claimant at the originating VLAN

-could result in loops, if a second BARC forwarder was present

- loops could *possibly* be prevented by labeling the forwarded PDU (e.g. SA) to prevent re-forwarding

-better to use a Registrar instead of a forwarder

Registrar with VLAN



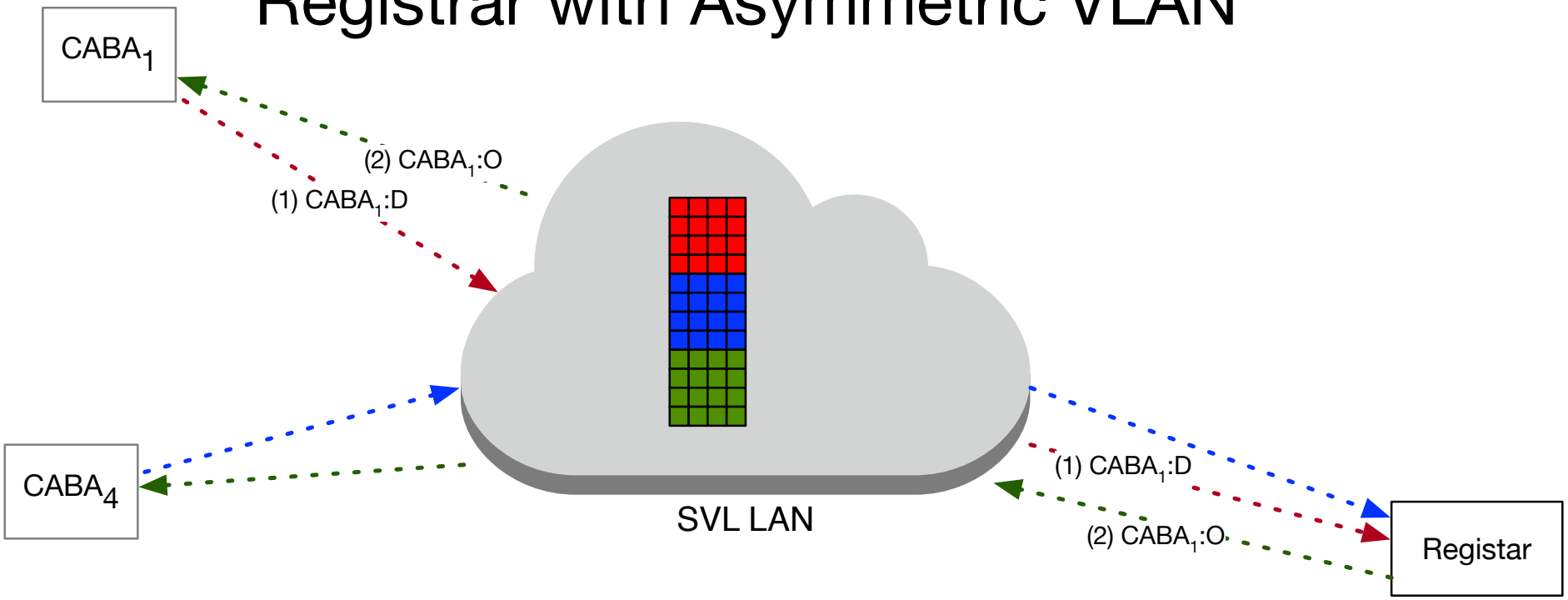
Network is configured with Registrar on all active VLANs on which BARC is used.

BARC claim from any VLAN is delivered to Registrar.
-Offer delivered on Claimant's VLAN

Registrar ensures that registered address is unique across all (or perhaps only some) of its VLANs.
-SVL or IVL will work

Registrar needs to remember over which VLANs the address was assigned.
-should be retained in State Machine

Registrar with Asymmetric VLAN



SVL is used for Asymmetric VLAN (IEEE Std 802.1Q Annex F.1.3)

Registrar can assign address to be unique across all VLANs available to the Registrar.

Summary

- Claimants operate with or without Registrars.

- Multiple registrars are supported, holding claims of disjoint RABIs.

- The block discretization provides:
  - a vast set of addresses to a LAN
  - a large set of temporary unicast addresses
  - operational efficiency and simplicity
  - both unicast and multicast addresses to Claimant
    - unicast and multicast subblocks share the same range, except for the I/G bit
      - could be exploited
    - devices needing both unicast and multicast addresses need make only one claim

- Could integrate with MMRP to limit propagation and eliminate learning of unicast AB content.
  - MMRP needs to efficiently handle address ranges
  - BARP could be specified as alternative MRP application
    (e.g. would understand an ABD)