Towards Verification of the PANA Authentication and Authorisation Protocol using Coloured Petri Nets

Steven Gordon

SIIT, Thammasat University
Thailand

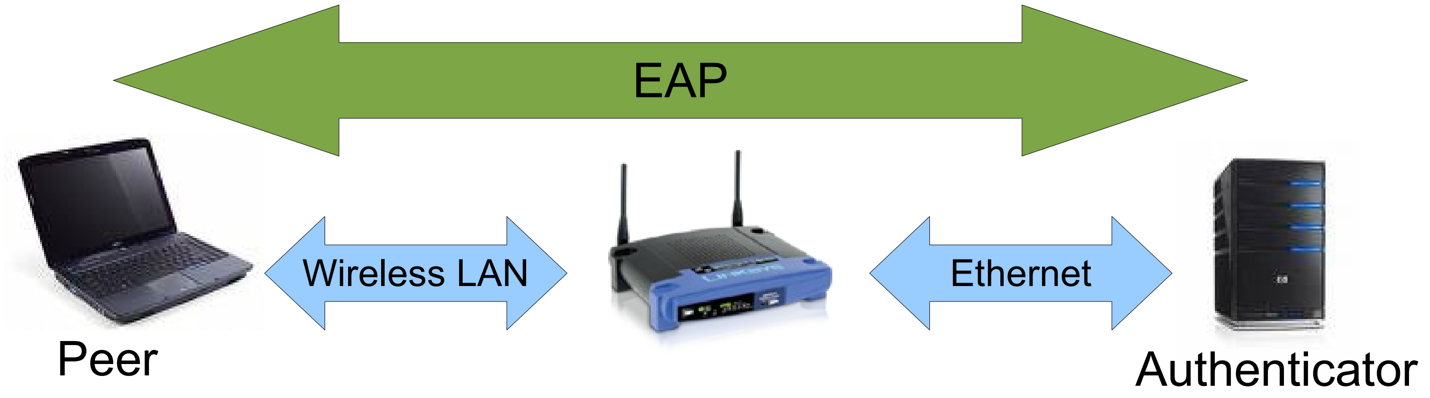
Outline

- What are EAP and PANA?
- Modelling PANA with CPNs: Why? How?
- Analysing PANA
  - Are there any deadlocks?
  - What are the effects of retransmissions?
  - How does PANA interact with users (EAP)?
- Observations and future work

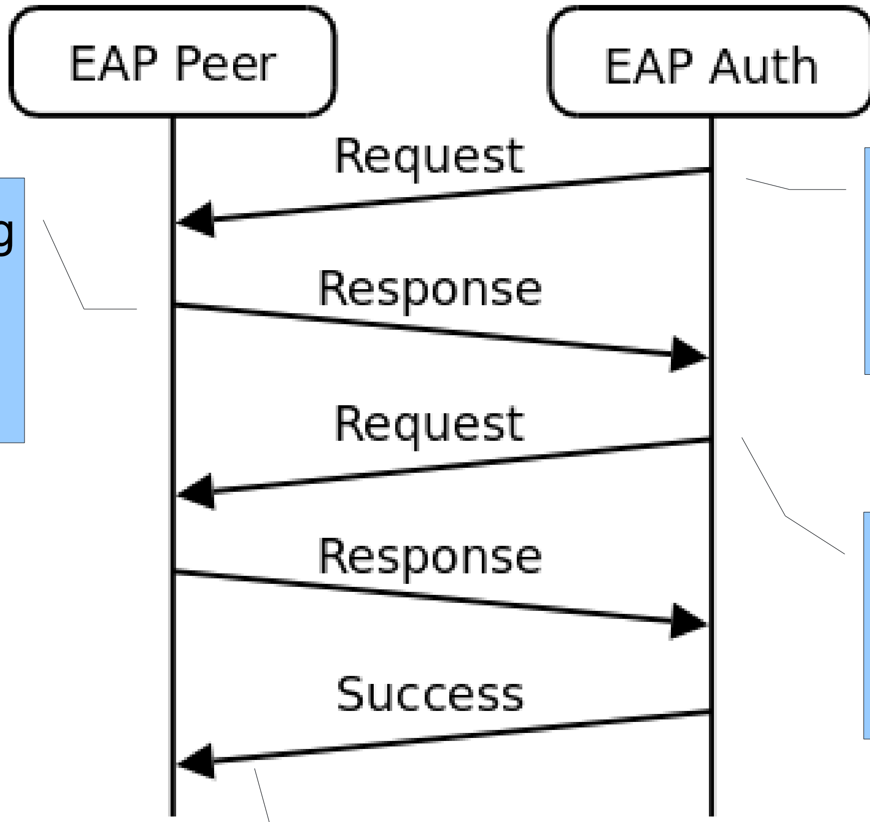
EAP and PANA

Authentication using EAP

- Extensible Authentication Protocol (EAP)
  - Designed for network access authentication when IP access is not available
    - PPP (dialup), IEEE 802.11 wireless LANs
  - Request/Response messages exchanged between *authenticator* and *peer*



EAP Messages



Peer responds depending on type of request: Identity-Request, MD5-Challenge, ...

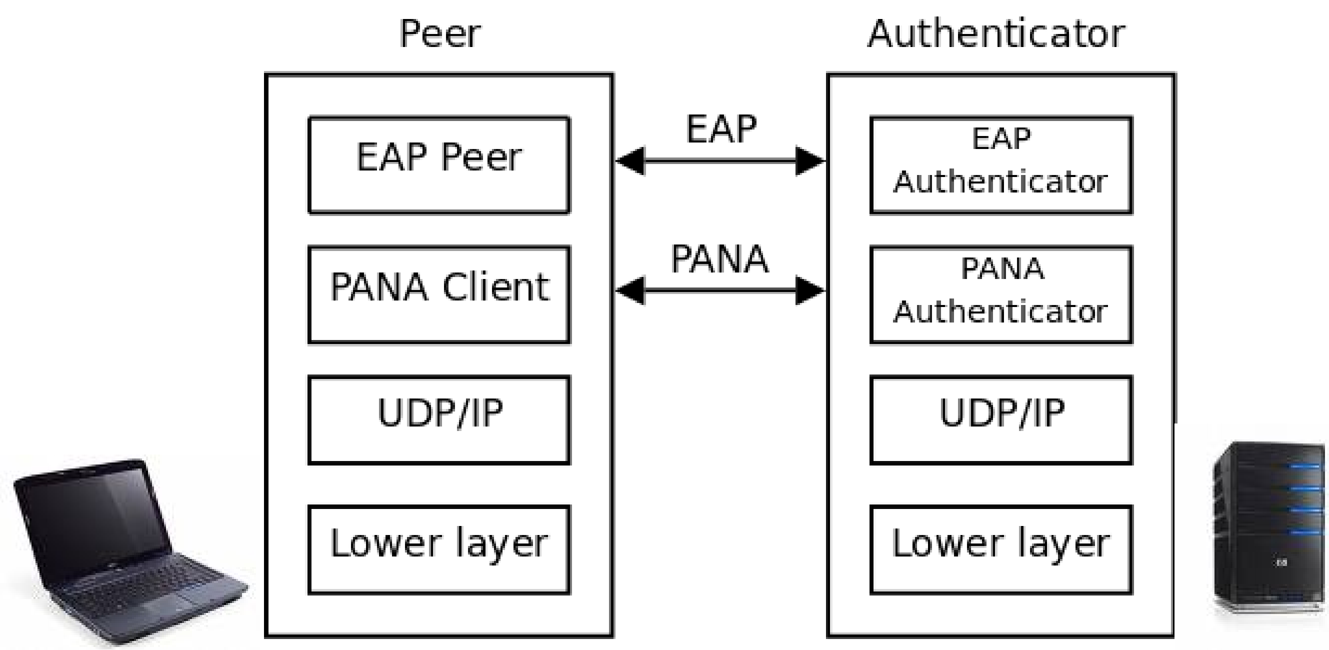
Authenticator initiates with an Authentication Request

Requests continue until Authenticator has sufficient information from Peer

Success or Failure message sent depending on responses from Peer

PANA and EAP

- Protocol for carrying Authentication for Network Access (PANA)
  - Designed to carry EAP messages over IP networks



Development of PANA by IETF

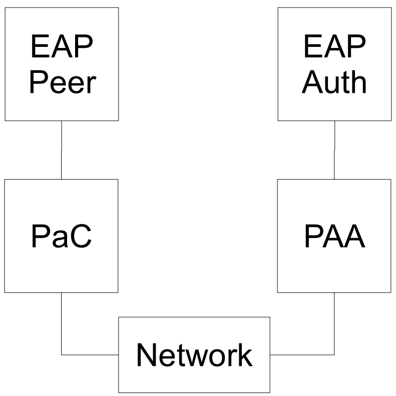
- EAP
  - Protocol description: IETF RFC 3478, June 2004
- PANA
  - Protocol description: IETF RFC 5191, May 2008
    - Started 2003, 18 versions
  - State machines: IETF RFC 5609, Aug 2009
    - Started 2005, 13 versions
  - (CPN modelling and analysis of PANA
    - Started 2007, total of 3-4 months of effort)

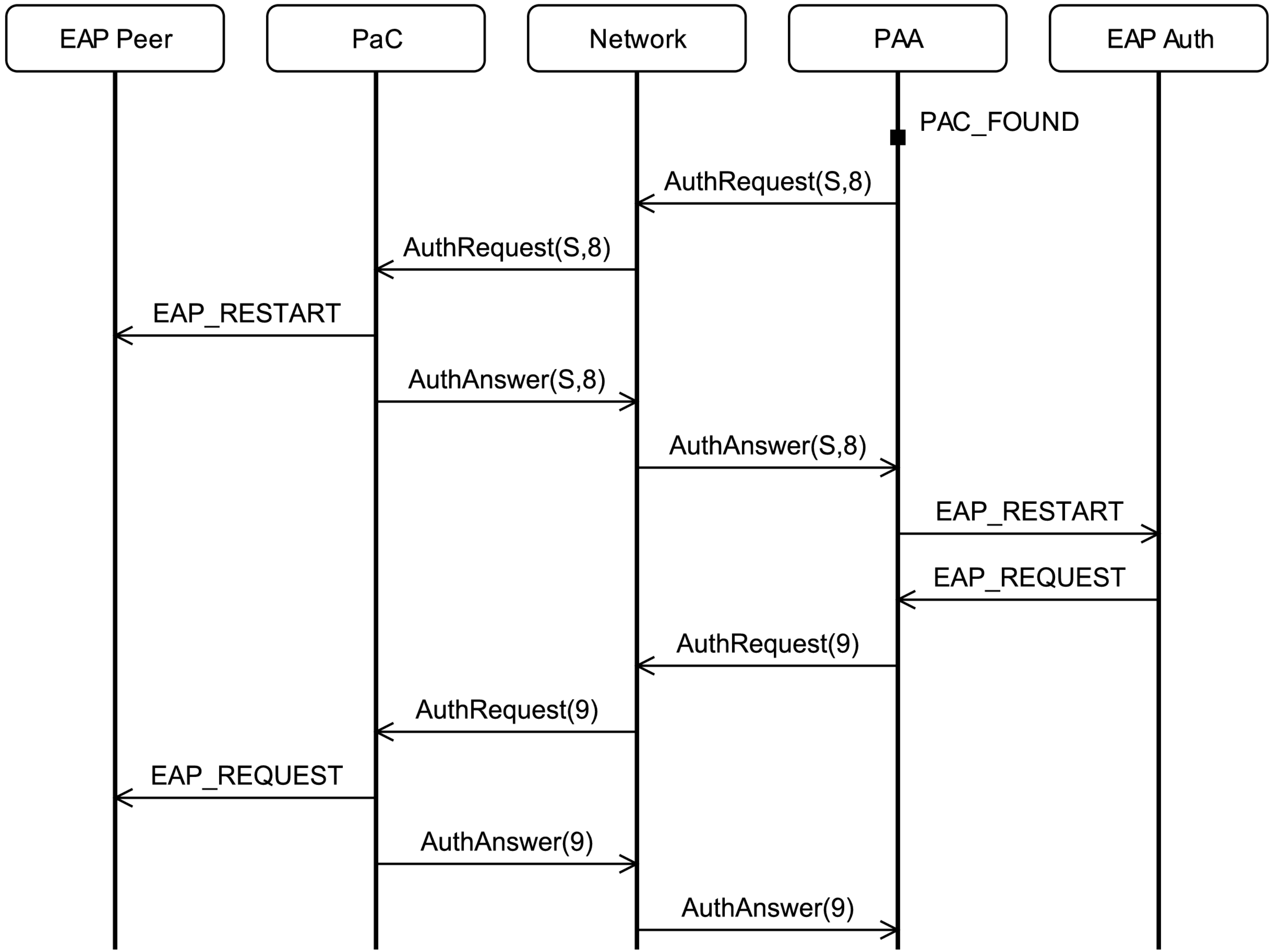
IETF: Internet Engineering Task Force
RFC: Request For Comments

PANA Operation

- PANA messages are used to:
  - Initialise and maintain the EAP session
  - Carry EAP messages (Request, Response, Success, Failure) between Peer and Authenticator
- PANA maintains its own session. Four phases:
  1) Authentication and Authorisation: perform the EAP authentication
  2) Access: once authentication, maintain the session
  3) Re-authentication: if session is about to expire
  4) Termination: either PaC or PAA may terminate

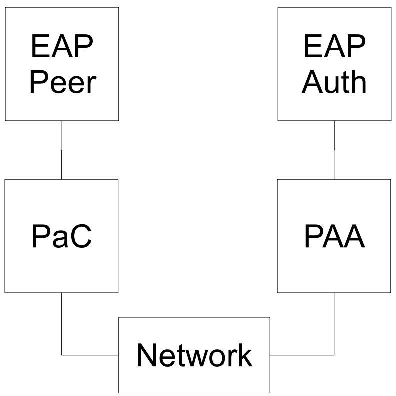
PANA Authentication Message Sequence Chart (No Piggybacking) – Part 1

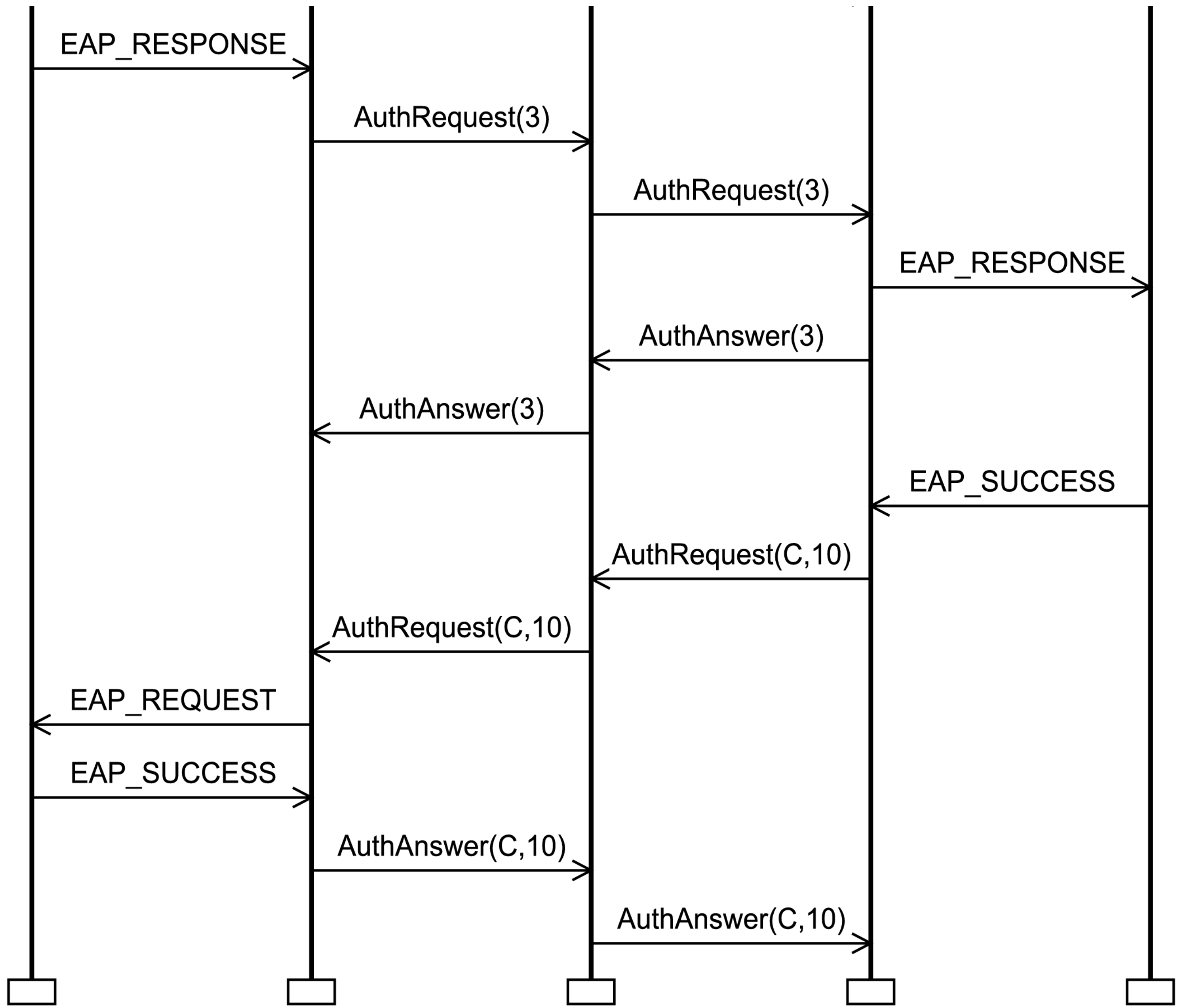




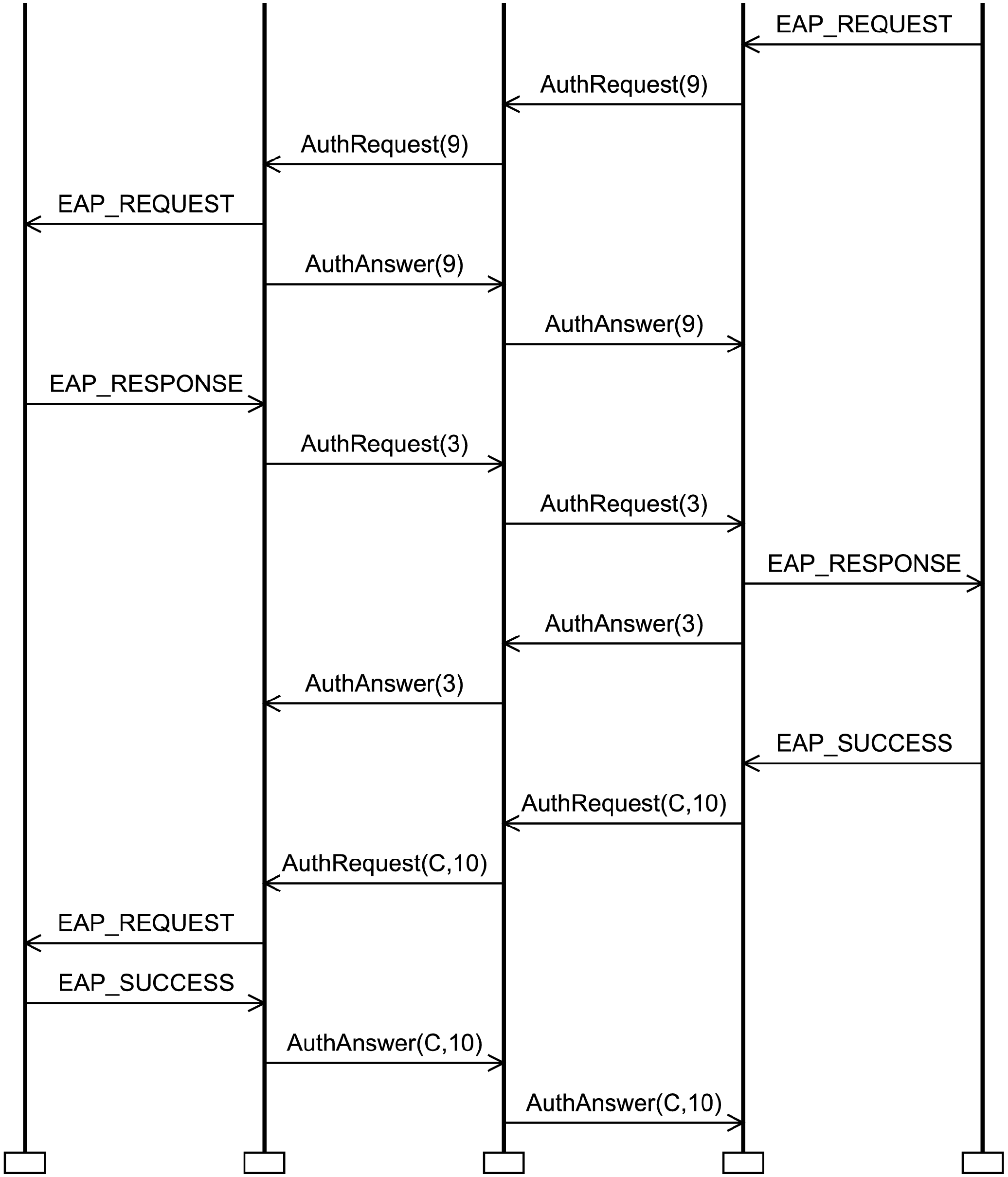
PaC: PANA Client
PAA: PANA Authenticator

PANA Authentication Message Sequence Chart (No Piggybacking) – Part 2

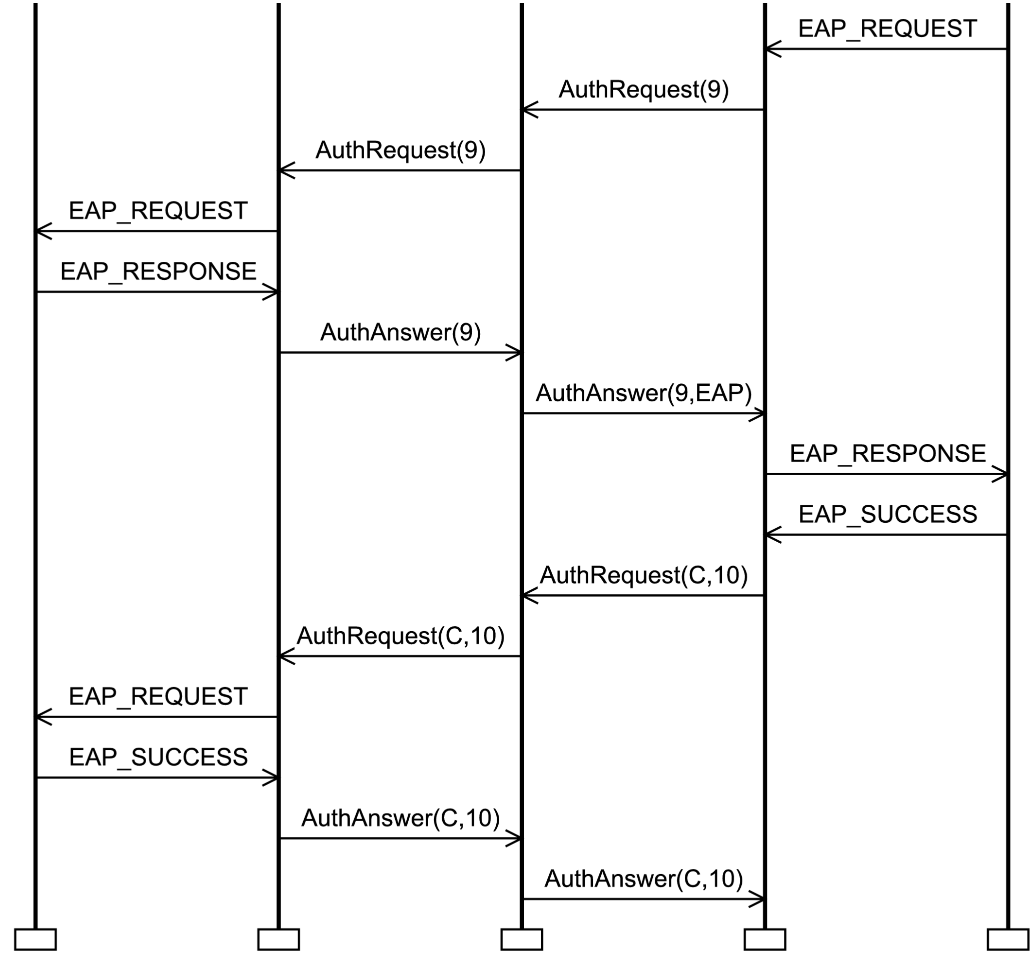




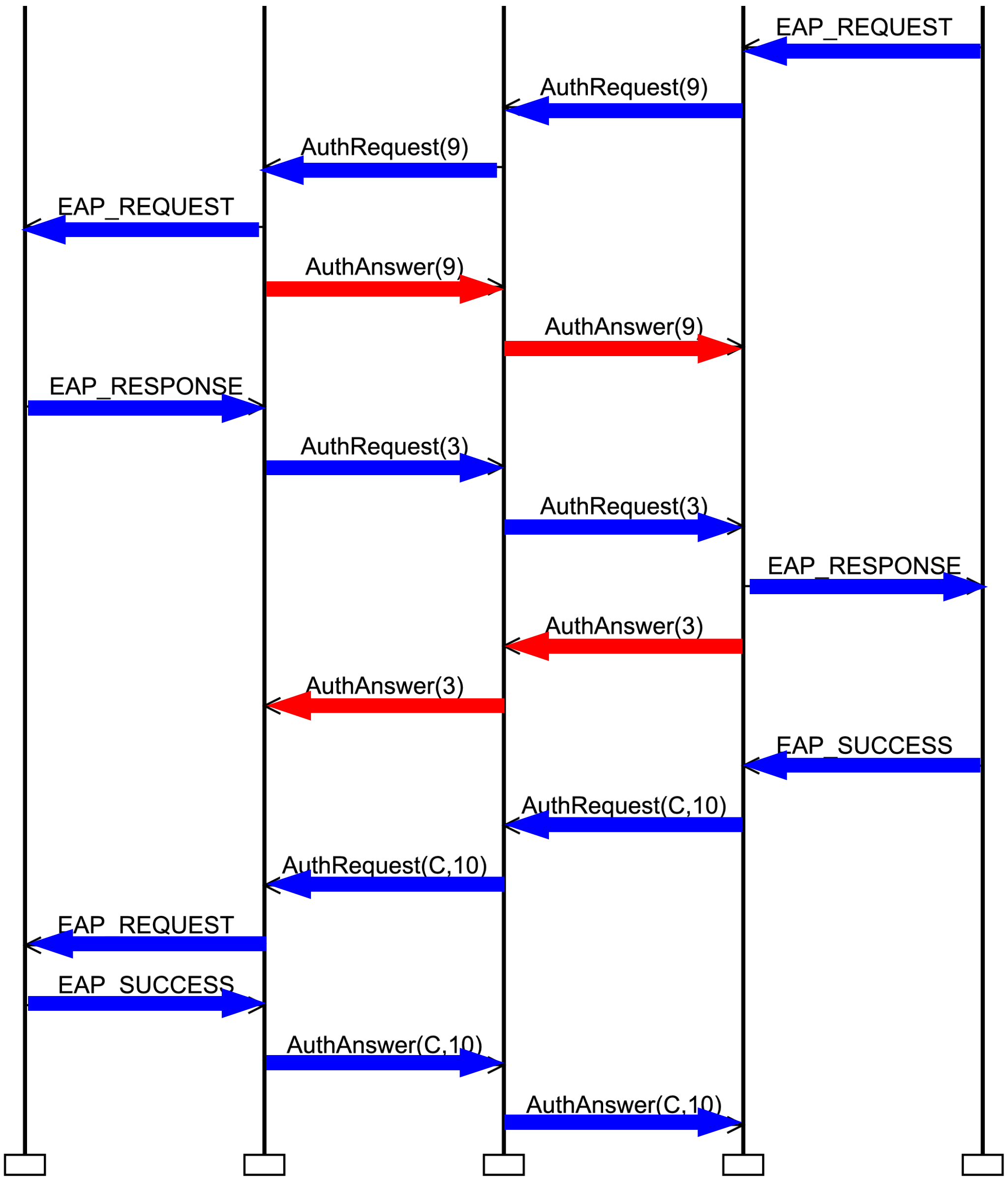
Piggybacking Off



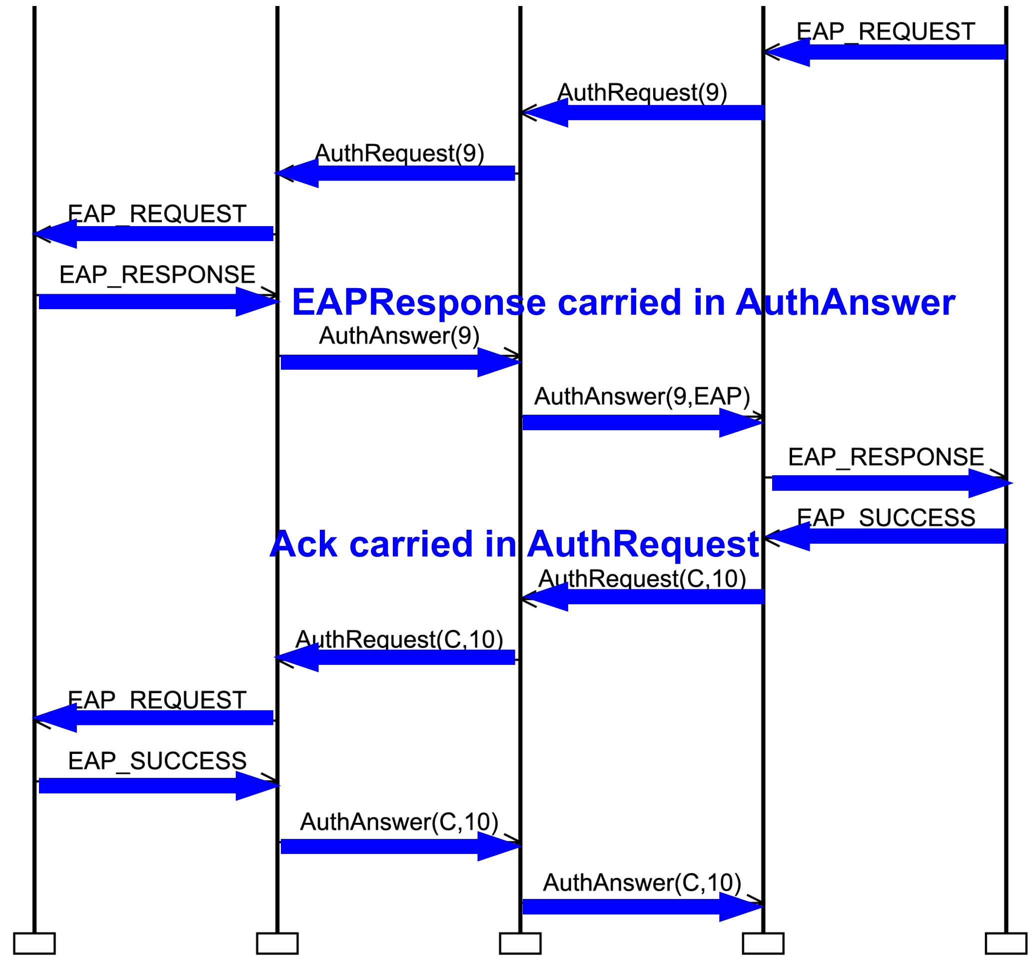
Piggybacking On



Piggybacking Off



Piggybacking On

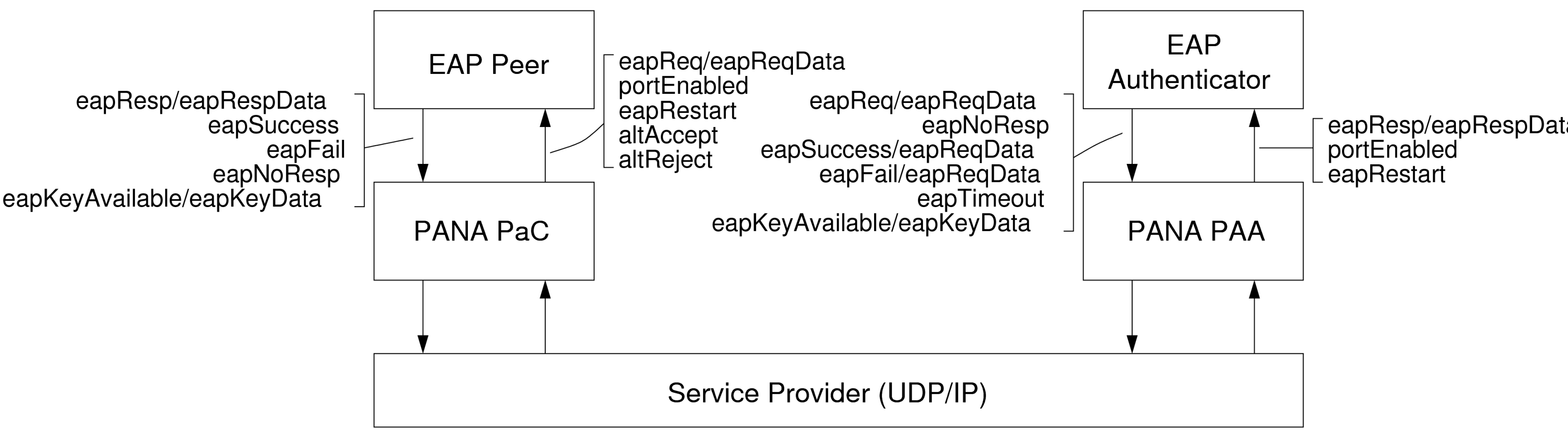


Other PANA Features

- Initiation of PANA session
  - By either PaC or PAA (method out of scope of PANA)
  - Optimised Initiation: if On, PAA can send EAP Request in initial AuthRequest message
- Each Request contains a 32-bit sequence number; responding Answer contains the same number
- Requests can be retransmitted if no answer; abort session if too many retransmits
- PANA messages use Attribute Value Pairs to carry EAP messages, authentication data and other security information

EAP/PANA Interface

- Protocols may define service offered to higher layer (e.g. EAP) independent from protocol operation
  - What messages can be exchanged between PANA and EAP, and in what order?
    - No explicit definition in PANA RFCs
    - Extract information from PANA RFC and EAP RFC: describe variables for communication between PANA/EAP



EAP/PANA Interface

No.	Entity	EAP	PANA	Primitive
1	Peer/PAC	-	AUTH_USER	CAuthUser
2	Peer/PAC	eapRestart	EAP_RESTART	CRestart
3	Peer/PaC	eapReq	EAP_REQUEST	CRequest
4	Peer/PaC	eapResp	EAP_RESPONSE	CResponse
5	Peer/PaC	eapSuccess	EAP_SUCCESS	CSuccess
6	Peer/PaC	eapFail	EAP_FAILURE	CFailure
7	Peer/PaC	-	-	CTimeout
8	Peer/PaC	-	ABORT	CAbort
9	Auth/PAA	-	PAC_FOUND	APacFound
10	Auth/PAA	eapRestart	EAP_RESTART	ARestart
11	Auth/PAA	eapReq	EAP_REQUEST	ARequest
12	Auth/PAA	-	-	AResponse
13	Auth/PAA	eapSuccess	EAP_SUCCESS	ASuccess
14	Auth/PAA	eapFail	EAP_FAILURE	AFailure
15	Auth/PAA	-	EAP_TIMEOUT	ATimeout
16	Auth/PAA	-	ABORT	AAbort
17	Peer/PaC	-	DISCARD	CDiscard
18	Auth/PAA	-	DISCARD	ADiscard

Modelling PANA with CPNs

Motivation

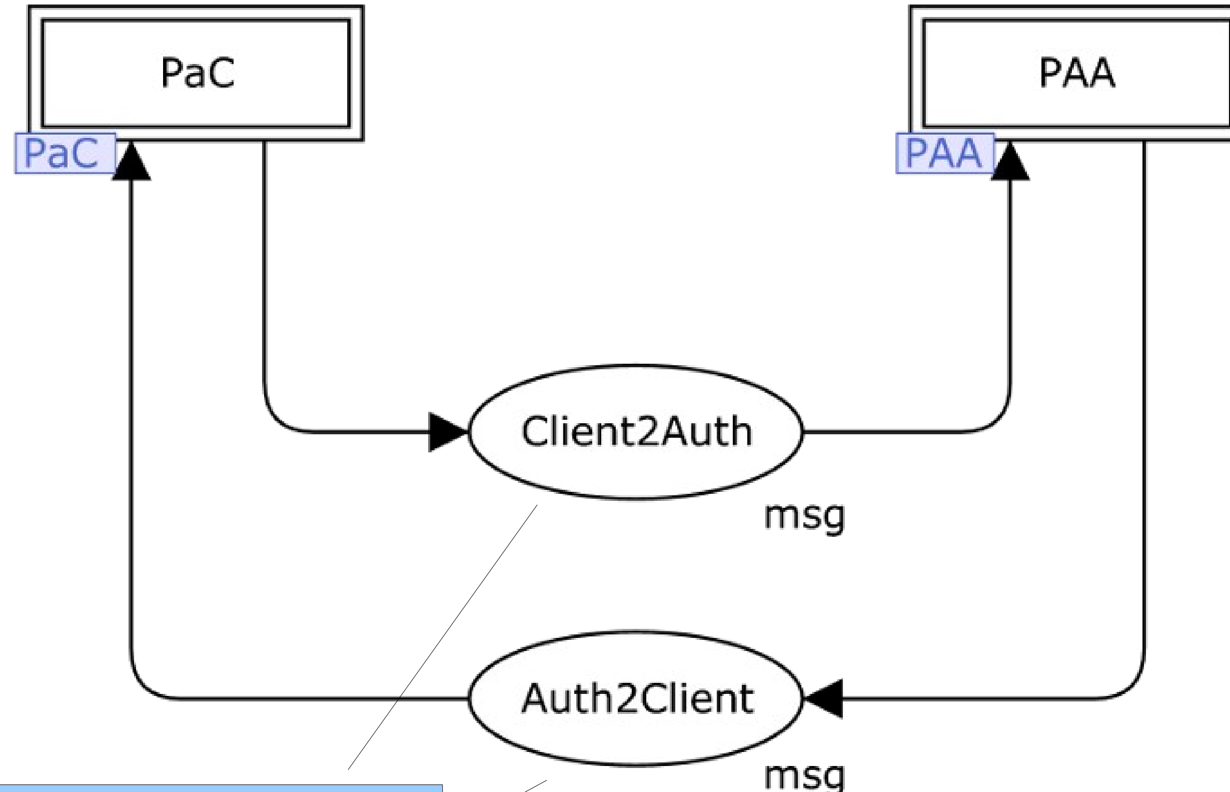
- Verify the PANA specification is accurate and unambiguous
- Gain understanding of PANA's operation
- Gain experience in CPN Tools

Methodology

- Apply steps from a Protocol Engineering Methodology using CPNs and CPN Tools
  1. Formal modelling of PANA protocol using CPNs
     - State-based modelling approach using PANA state machines
  2. Simulation of the PANA protocol
     - Stepping through selected sequences
     - Generation of Message Sequence Charts
  3. State space analysis of PANA
     - Absence of deadlocks, bounds
  4. Generation of PANA protocol language
     - To understand interactions between PANA and EAP

PANA Hierarchy and Top Level

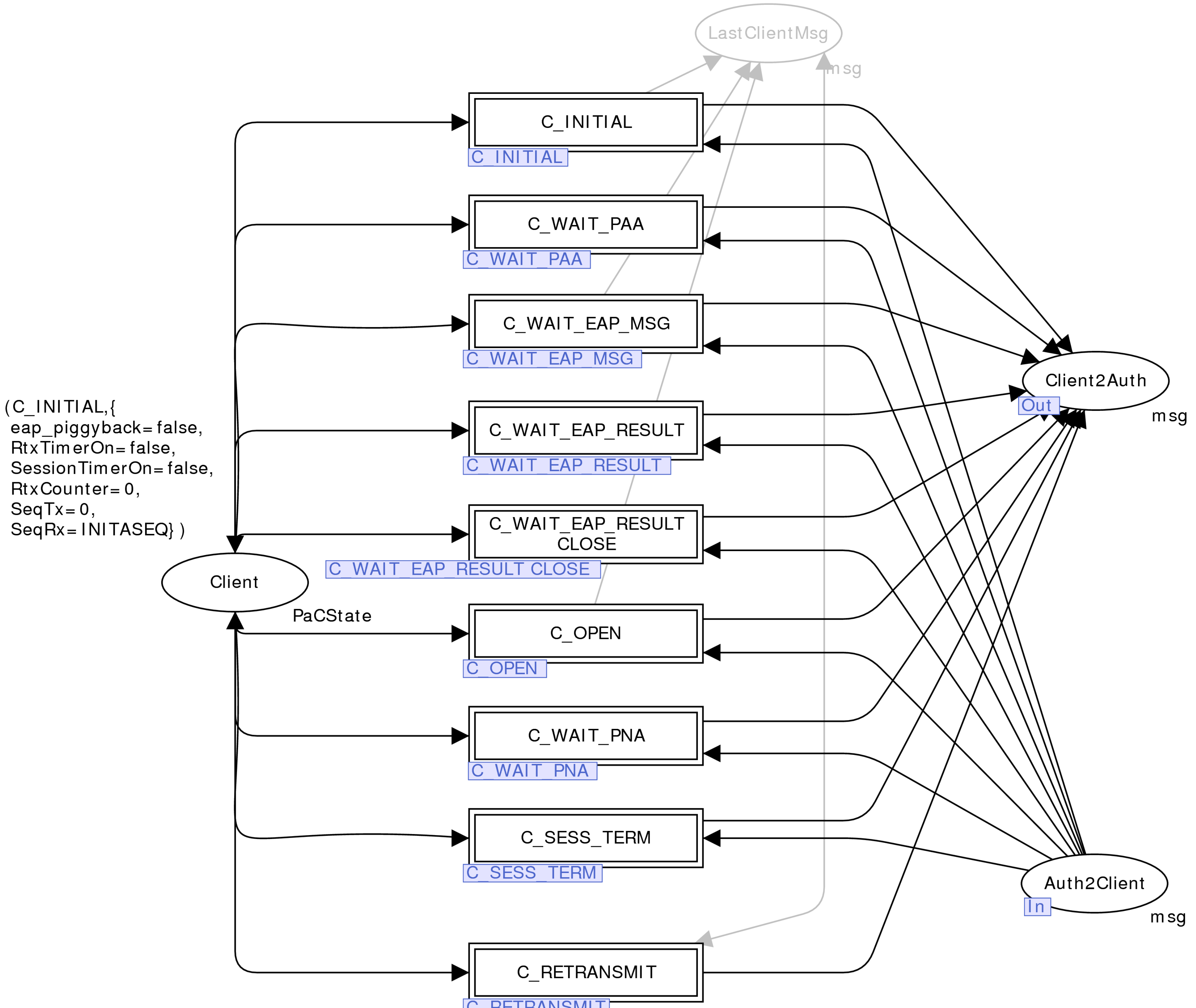
- ∨PANA
  - ∨PaC
    - ∨C_INITIAL
      - C_INITIAL_Piggyback
      - C_INITIAL_No_Piggyback
    - C_WAIT_PAA
    - C_WAIT_EAP_MSG
    - C_WAIT_EAP_RESULT
    - C_WAIT_EAP_RESULT CLOSE
    - C_OPEN
    - C_WAIT_PNA
    - C_SESS_TERM
    - C_RETRANSMIT
  - ∨PAA
    - A_INITIAL
    - A_WAIT_EAP_MSG
    - A_WAIT_SUCC_PAN
    - A_WAIT_FAIL_PAN
    - A_OPEN
    - A_WAIT_PNA_PING
    - A_WAIT_PAN_OR_PAR
    - A_SESS_TERM
    - A_RETRANSMIT



Communication places
Unordered, reliable

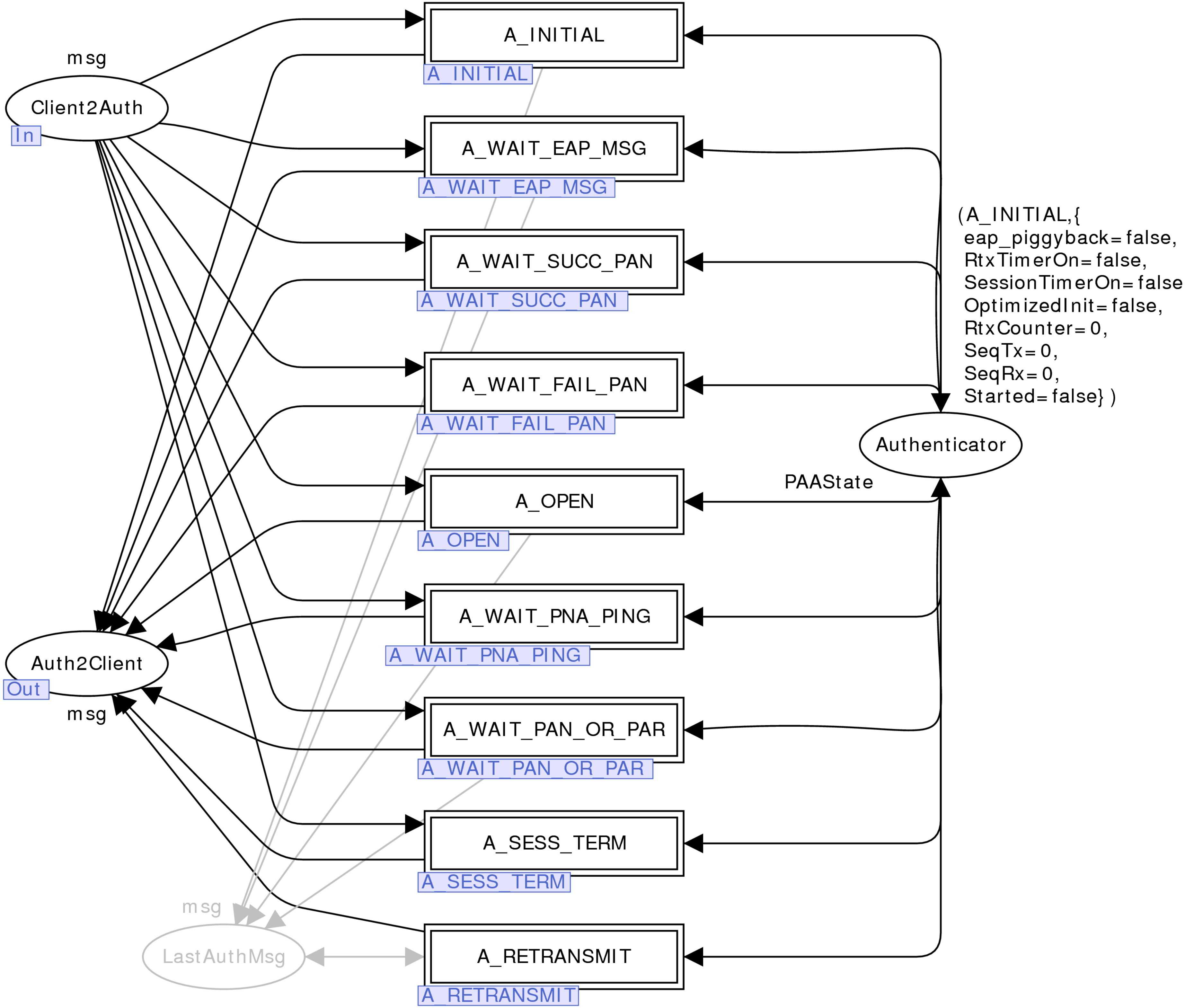
States of PAA
Each page models events/actions of corresponding state
Each state is represented by state table in RFC 5609

PaC



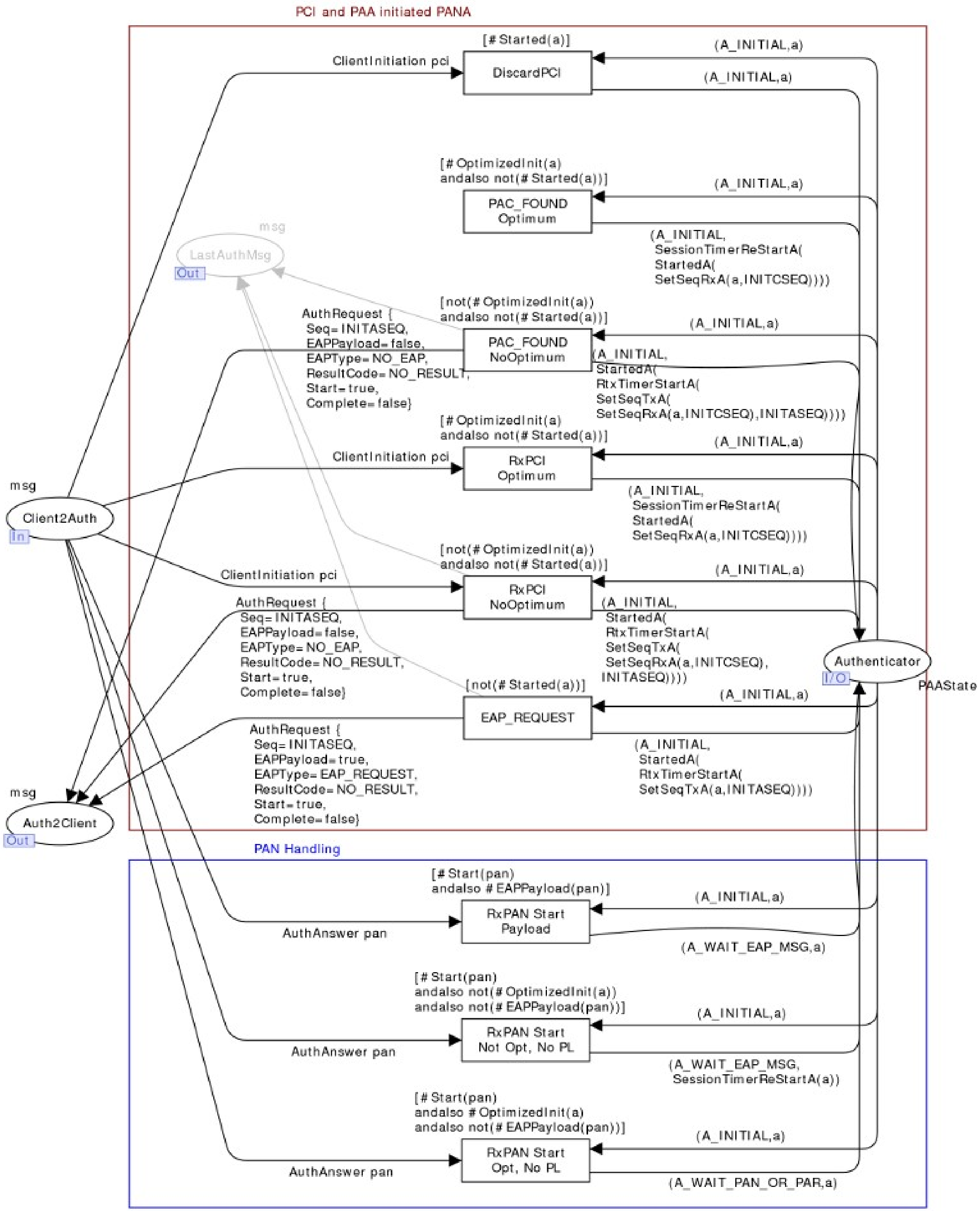
LastClientMsg
msg
C_INITIAL
C_INITIAL
C_WAIT_PAA
C_WAIT_PAA
C_WAIT_EAP_MSG
C_WAIT_EAP_MSG
C_WAIT_EAP_RESULT
C_WAIT_EAP_RESULT
C_WAIT_EAP_RESULT CLOSE
C_WAIT_EAP_RESULT CLOSE
C_OPEN
C_OPEN
C_WAIT_PNA
C_WAIT_PNA
C_SESS_TERM
C_SESS_TERM
C_RETRANSMIT
C_RETRANSMIT
(C_INITIAL,{
eap_piggyback=false,
RtxTimerOn=false,
SessionTimerOn=false,
RtxCounter=0,
SeqTx=0,
SeqRx=INITASEQ})
Client
PaCState
Client2Auth
Out
msg
Auth2Client
In
msg

PAA



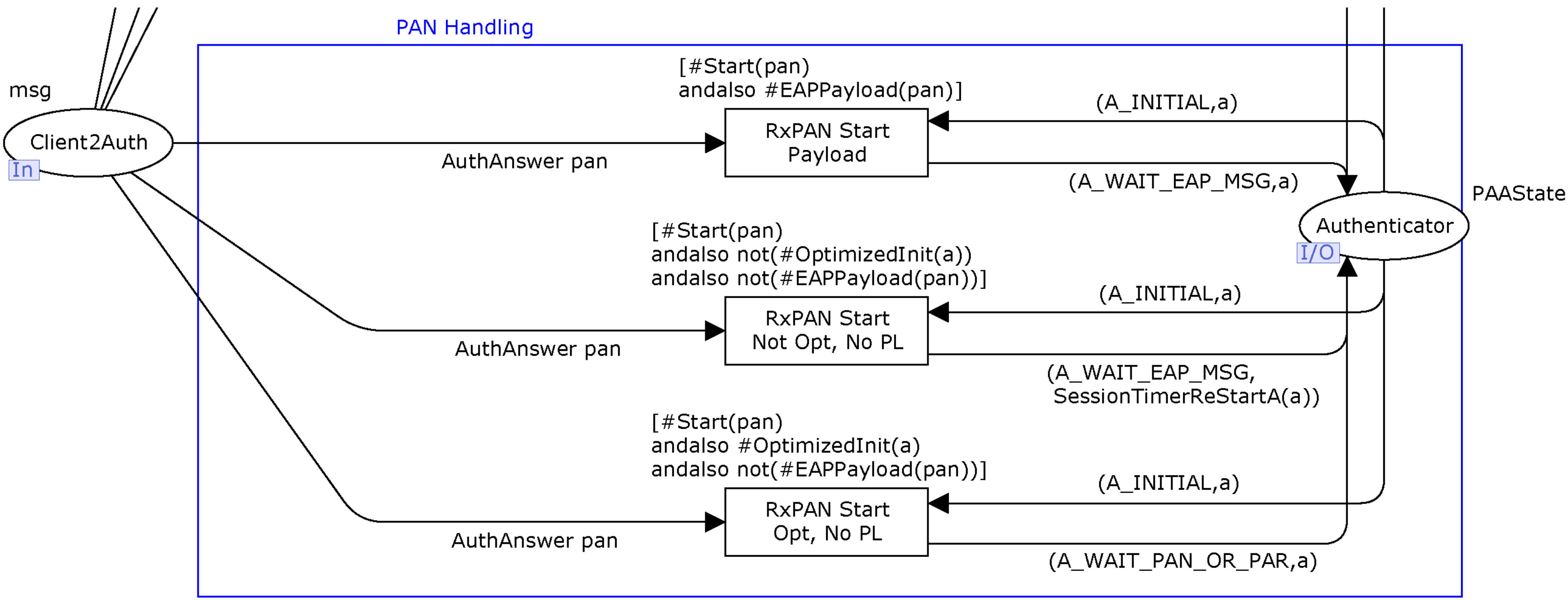
msg
Client2Auth
In
A_INITIAL
A_INITIAL
A_WAIT_EAP_MSG
A_WAIT_EAP_MSG
A_WAIT_SUCC_PAN
A_WAIT_SUCC_PAN
A_WAIT_FAIL_PAN
A_WAIT_FAIL_PAN
A_OPEN
A_OPEN
A_WAIT_PNA_PING
A_WAIT_PNA_PING
A_WAIT_PAN_OR_PAR
A_WAIT_PAN_OR_PAR
A_SESS_TERM
A_SESS_TERM
A_RETRANSMIT
A_RETRANSMIT
Auth2Client
Out
msg
msg
LastAuthMsg
(A_INITIAL,{
eap_piggyback=false,
RtxTimerOn=false,
SessionTimerOn=false,
OptimizedInit=false,
RtxCounter=0,
SeqTx=0,
SeqRx=0,
Started=false})
Authenticator
PAAState

PAA INITIAL state (A_INITIAL)



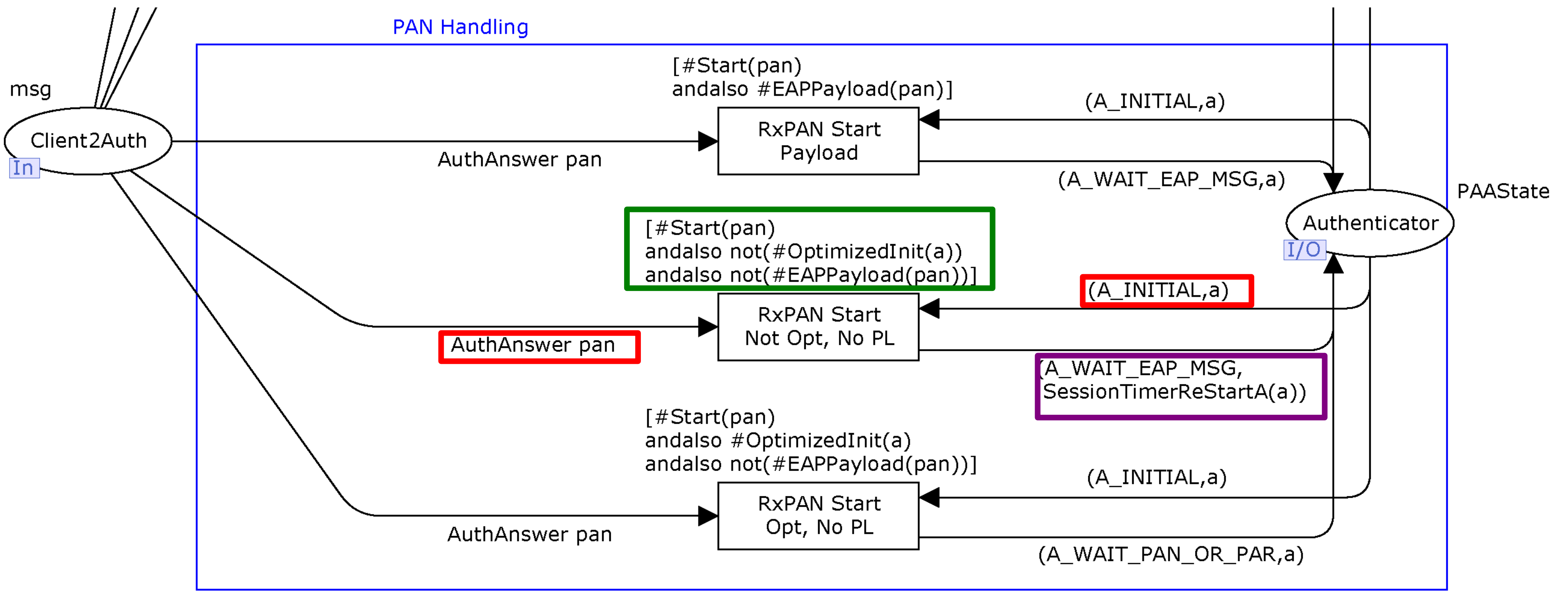
Selection of PANA State Table and CPN for PAA in INITIAL state

```
Exit Condition            Exit Action                 Exit State
------------------------+---------------------------+------------
Rx:PAN[S] &&              if (PAN.exist_avp           WAIT_EAP_MSG
((OPTIMIZED_INIT ==          ("EAP-Payload"))
  Unset) ||                TxEAP();
PAN.exist_avp             else {
  ("EAP-Payload"))          EAP_Restart();
                            SessionTimerReStart
                             (FAILED_SESS_TIMEOUT);}

Rx:PAN[S] &&              None();                     WAIT_PAN_OR_PAR
(OPTIMIZED_INIT ==
  Set) &&
! PAN.exist_avp
 ("EAP-Payload")
```

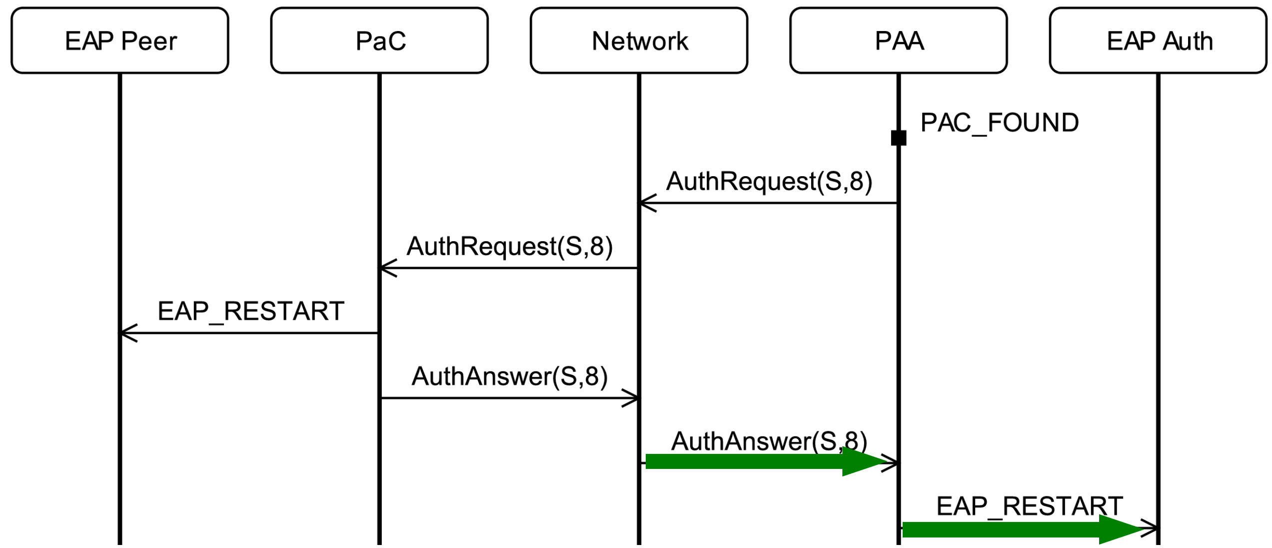


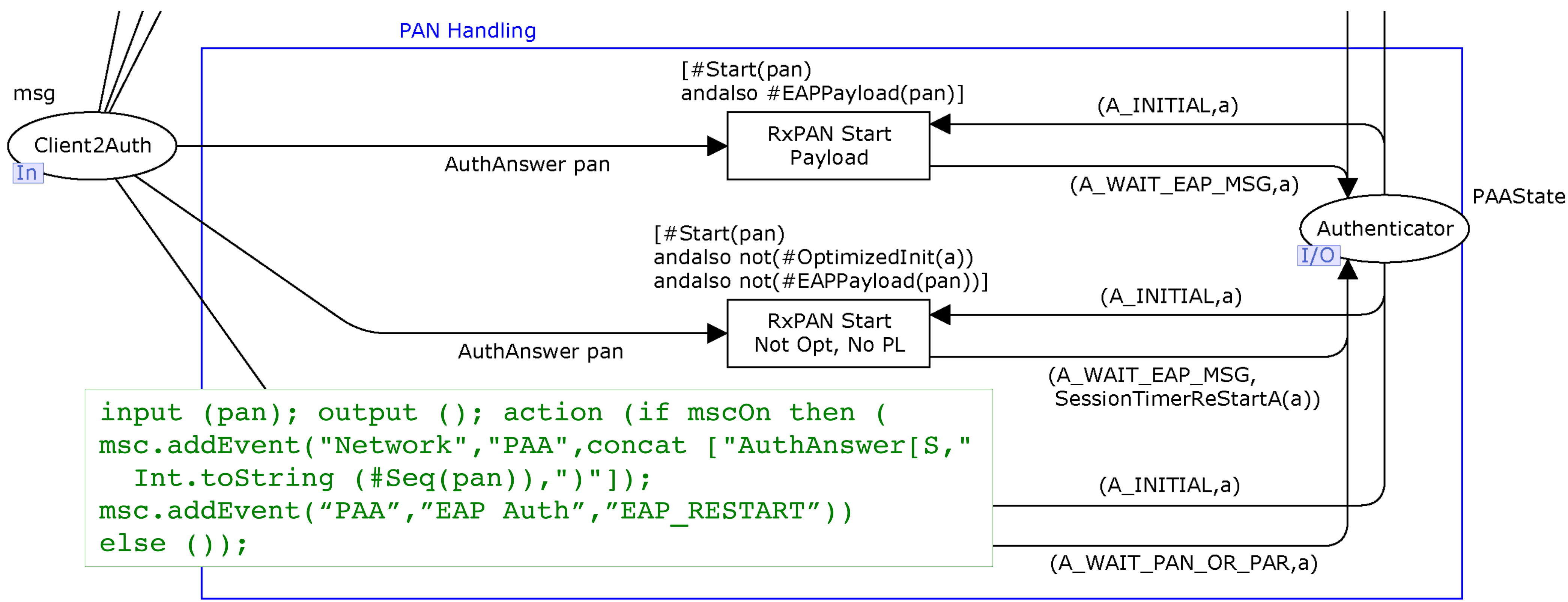
Selection of PANA State Table and CPN for PAA in INITIAL state

```
Exit Condition              Exit Action                 Exit State
------------------------+---------------------------+------------
Rx:PAN[S] &&                if (PAN.exist_avp           WAIT_EAP_MSG
((OPTIMIZED_INIT ==            ("EAP-Payload"))
  Unset) ||                   TxEAP();
PAN.exist_avp               else {
  ("EAP-Payload"))            EAP_Restart();
                              SessionTimerReStart
                               (FAILED_SESS_TIMEOUT);}

Rx:PAN[S] &&                None();                     WAIT_PAN_OR_PAR
(OPTIMIZED_INIT ==
  Set) &&
! PAN.exist_avp
 ("EAP-Payload")
```



Message Sequence Chart and Code Segment using BRITNeY and MSC Library





Analysis of PANA

State Space Analysis

- Aims
  - Gain understanding of PANA operation
  - Prove the absence of deadlocks in protocol
  - Generate PANA protocol language

- Configurations
  - Piggybacking: Off, On
  - Optimised Initiation: Off, On
  - Maximum retransmit limit at PaC: 0, 1, 2, 3, ...
  - Maximum retransmit limit at PAA: 0

State Space Analysis: No Retransmissions

Parameter		State Space			Bounds	
Piggyback	Optimised Init	States	Arcs	Terminals	Client2Auth	Auth2Client
Off	Off	15531	34047	6866	5	3
On	Off	3436	7212	1265	3	2
Off	On	12079	26360	5292	5	3
On	On	2085	4233	775	3	2

- Only Authentication/Authorisation phase considered
- Packet loss is not possible
- Single EAP Request sent by Authenticator
- 4-bit sequence numbers

Terminal States

- Define expected terminal states based on the protocol state of the PaC/PAA, e.g. Open, Closed, Initial, ...

		PaC						
		Initial	WaitPAA	WaitEAP Msg	WaitEAP Result	WaitEAPR esultClose	Open	Closed
PAA	Initial							PaC Abort
	WaitEAPMsg							
	WaitSuccPAN							
	WaitFailPAN							
	WaitPANPAR							
	Open		PaC enters WaitPAA				Success	PaC Abort
	Closed		PAA Abort				PAA Abort	Failed or Abort

Terminal States

- No unexpected terminal states, but refinement of definition of expected terminal states is necessary

Number of terminal states Piggyback: Off or On Optimised Init: Off or On		PaC						
		Initial	WaitPAA	WaitEAP Msg	WaitEAP Result	WaitEAPR esultClose	Open	Closed
PAA	Initial	0	0	0	0	0	0	0 0 1 1
	WaitEAPMsg	0	0	0	0	0	0	0
	WaitSuccPAN	0	0	0	0	0	0	0
	WaitFailPAN	0	0	0	0	0	0	0
	WaitPANPAR	0	0	0	0	0	0	0
	Open	0	11 8 7 4	0	0	0	61 41 59 23	456 61 333 38
	Closed	0	84 186 68 116	0	0	0	164 98 164 59	6090 871 4660 534

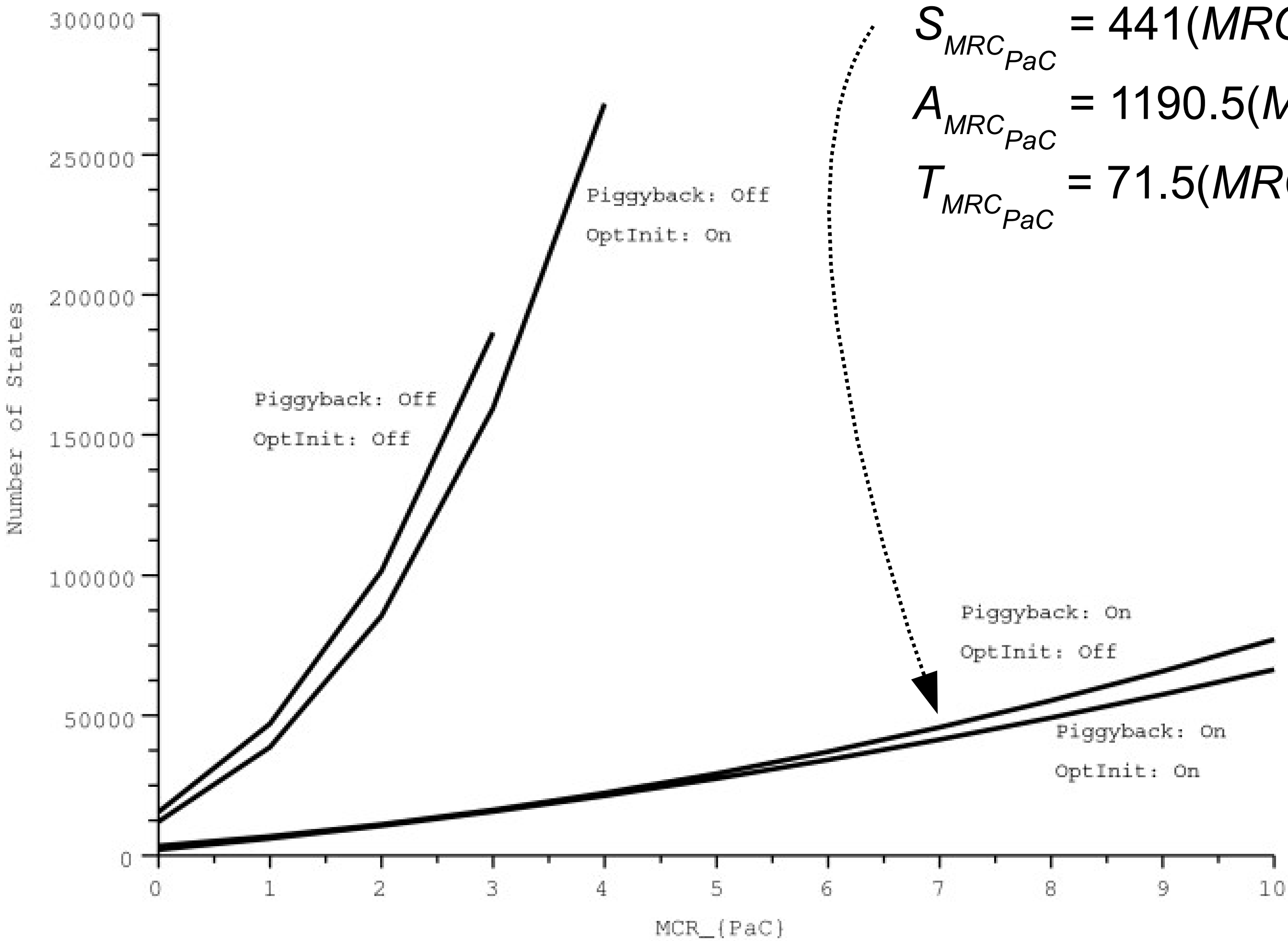
The Effect of Retransmissions

- Both PaC and PAA may retransmit messages up to a maximum number of times
  - PaC retransmission limit: MRC_{PaC}
  - PAA retransmission limit: MRC_{PAA}
  - Abort when limit is reached
- State space analysis for $MRC_{PAA} = 0$:
  - Piggyback Off, Optimised Init Off, MRC_{PaC}: 0, 1, 2, 3
  - Piggyback Off, Optimised Init On, MRC_{PaC}: 0, 1, 2, 3, 4
  - Piggyback On, Optimised Init Off, MRC_{PaC}: 0, 1, 2, 3, 4, 5, 10
  - Piggyback On, Optimised Init On, MRC_{PaC}: 0, 1, 2, 3, 4, 5, 10

The Effect of Retransmissions

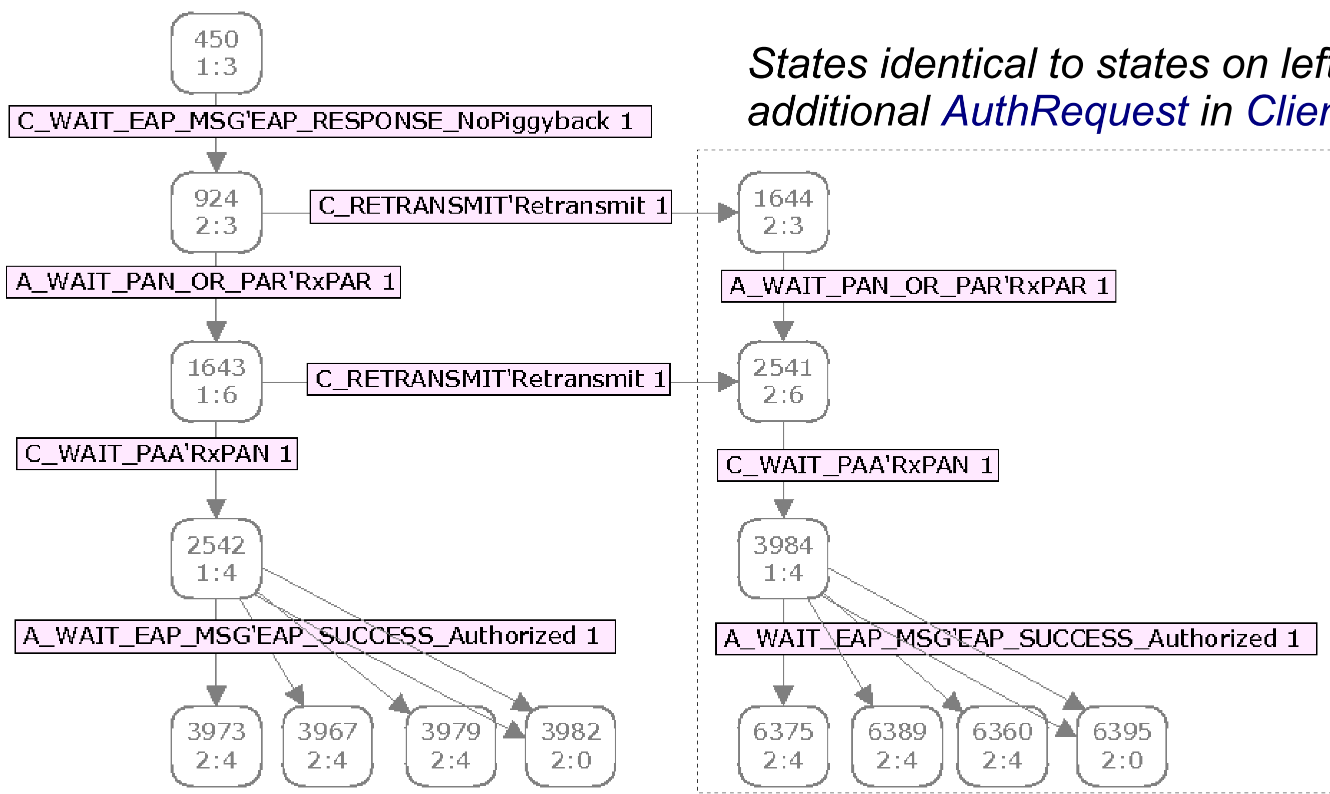
- Piggybacking on: state space number of states, arcs and terminals expressed as polynomials

$$S_{MRC_{PaC}} = 441(MRC_{PaC})^2+2958(MRC_{PaC})+3436$$
$$A_{MRC_{PaC}} = 1190.5(MRC_{PaC})^2+7029.5(MRC_{PaC})+7212$$
$$T_{MRC_{PaC}} = 71.5(MRC_{PaC})^2+813.5(MRC_{PaC})+1265$$



Piggybacking and Retransmissions

- Retransmissions (by PaC)
  - Piggyback On: PaC can only retransmit ClientInitiation message
  - Piggyback Off: PaC can retransmit ClientInitiation and AuthRequest messages



States identical to states on left, except additional AuthRequest in Client2Auth place

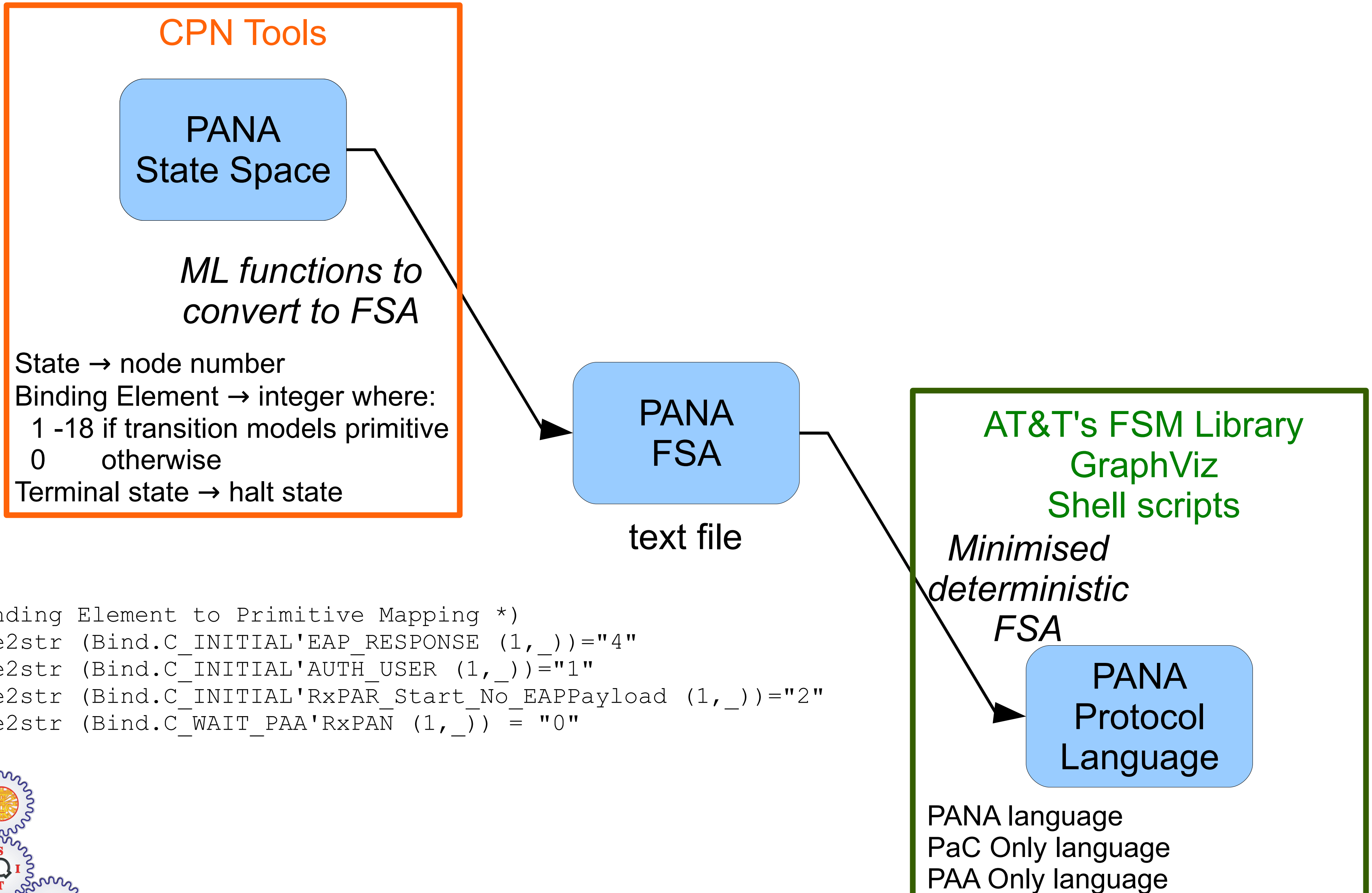
Observations from Analysis

- Terminal markings
  - Valid protocol states for PaC and PAA
  - Closer examination of state variables needed
- Integer bounds on places *Client2Auth* and *Auth2Client*
  - Number of messages sent before waiting for response
  - Bound on *Client2Auth* increases linearly with MRC_{PaC}
  - Useful in dimensioning buffer sizes
- State space size
  - Dependent on MRC_{PaC}
  - Patterns suggest state space reduction techniques can be applied

Language Analysis

- PANA protocol language: set of sequences of primitives exchanged between PANA and EAP
  - Should match the PANA service language
  - BUT PANA service language not defined!
- Aim: understand the PANA protocol language as step towards defining PANA service language
  - Currently only visual inspection of subsets of the full language

Language Analysis Methodology

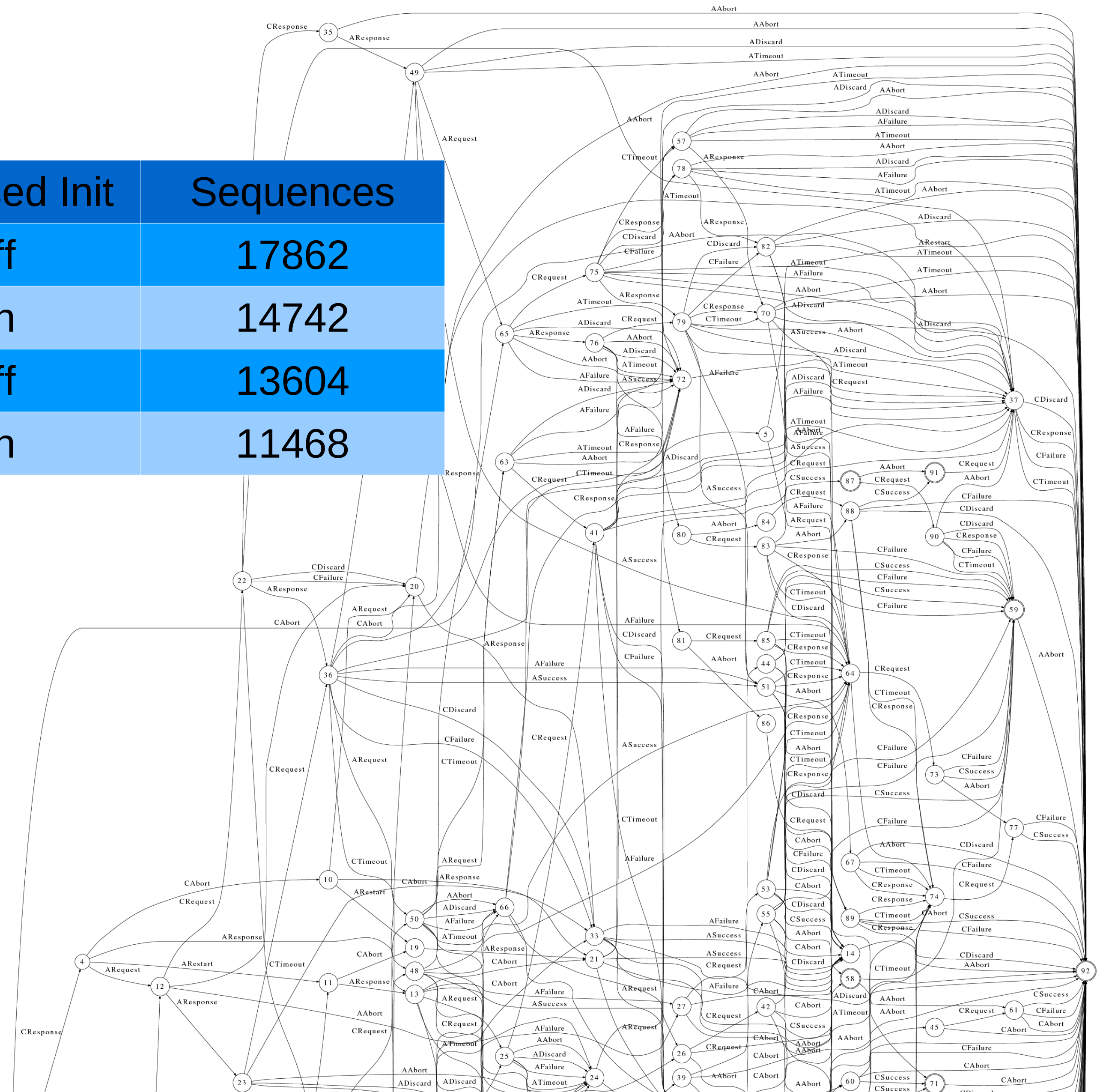



```
(* Binding Element to Primitive Mapping *)
fun be2str (Bind.C_INITIAL'EAP_RESPONSE (1,_))="4"
  | be2str (Bind.C_INITIAL'AUTH_USER (1,_))="1"
  | be2str (Bind.C_INITIAL'RxPAR_Start_No_EAPPayload (1,_))="2"
  | be2str (Bind.C_WAIT_PAA'RxPAN (1,_)) = "0"
...
```

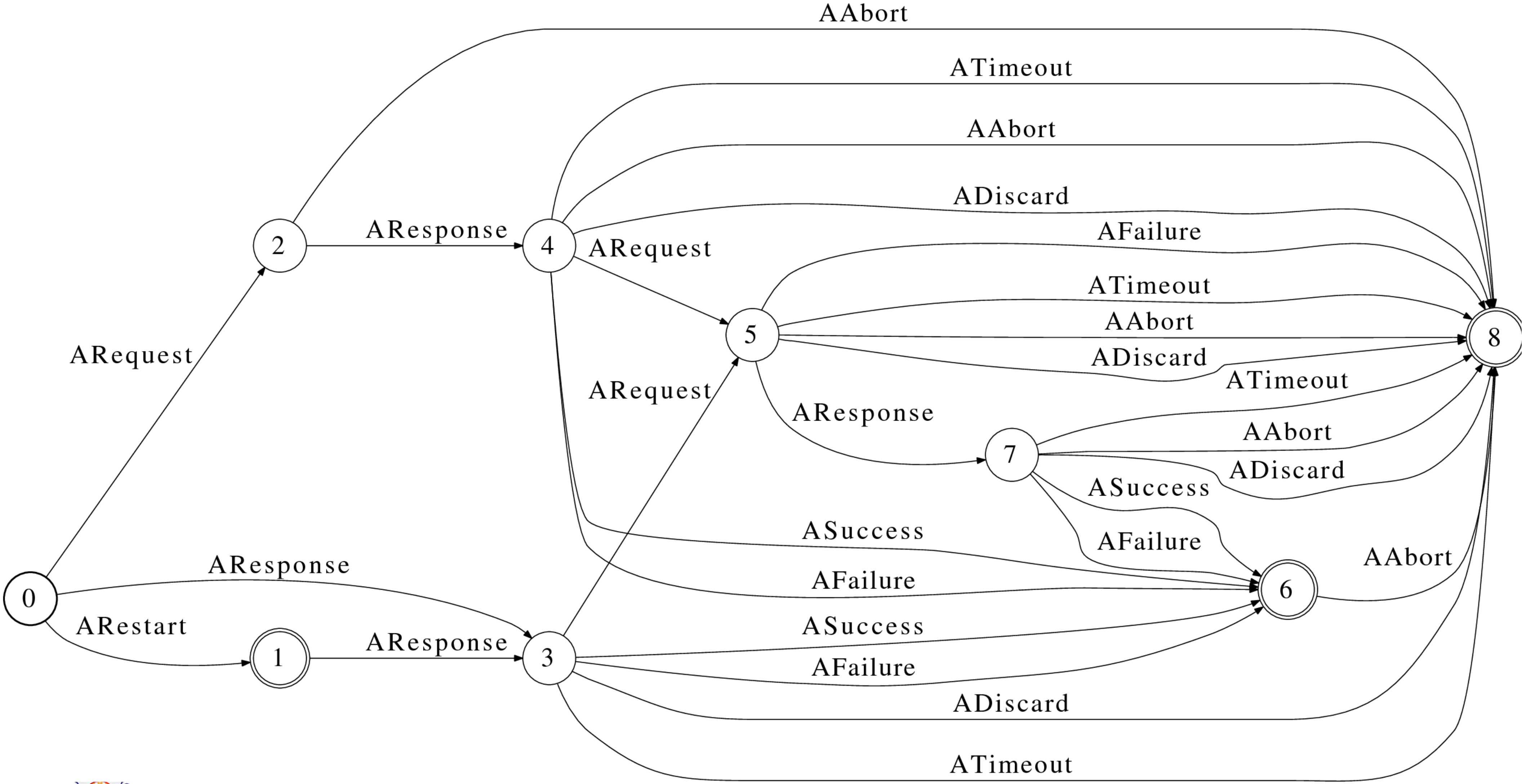
Full PANA Language

Number of sequences

Piggyback	Optimised Init	Sequences
Off	Off	17862
Off	On	14742
On	Off	13604
On	On	11468



PAA Only Language



Piggybacking On
Optimised Initiation On

Summary

- CPN modelling and analysis of an Internet authentication protocol
- Applied state-based modelling approach to PANA
  - Map state table entries from RFC to CPN transitions
- State space and language analysis
  - Increased confidence in the PANA RFC
  - Refinement of model and analysis techniques necessary

Future Work

- Precise definition of terminal markings
- Analysis of Access, Re-authentication and Termination phases
- Relaxing assumptions (reliable communications, single EAP Request)
- Prove state space results independent of retransmission limits
- Further develop tools for protocol engineering
  - Language analysis, CPNTools+SVN+Diff, generate state tables from model, ...

Thank You

(and the reviewers)