

PaC State Machine

CONDITION	ACTION
Rx: PSR_EAP(1) (OFFLINE → WAIT EAP MSG IN DISC)	
Rx:PSR && PSR.exist_avp("EAP-Payload")	RtxTimerStop(); TxEAP(); SEPARATE=Unset;
Rx: PSR_NO_EAP(7) (OFFLINE → WAIT PAA)	
Rx:PSR && !PSR.exist_avp ("EAP-Payload") &&	RtxTimerStop(); PSA.insert_avp ("Session-Id");
MOBILITY==Set && resume_pana_sa() && PSR.exist_avp ("Cookie")	PSA.insert_avp("Nonce"); PANA_SA_RESUMED=Set; PSA.insert_avp("Cookie"); PSA.insert_avp("MAC"); Tx:PSA(); RtxTimerStart(); SEPARATE=Unset;
Rx:PSR && !PSR.exist_avp ("EAP-Payload") && MOBILITY==Set && resume_pana_sa() && !PSR.exist_avp ("Cookie")	RtxTimerStop(); PSA.insert_avp("Session-Id"); PSA.insert_avp("Nonce"); PSA.insert_avp("MAC"); Tx:PSA(); PANA_SA_RESUMED=Set;
Rx:PSR && !PSR.exist_avp ("EAP-Payload") && (MOBILITY==Unset !resume_pana_sa()) && PSR.S_flag==1 && SEPARATE==Set && PSR.exist_avp ("Cookie")	RtxTimerStop(); if (choose_isp()) PSA.insert_avp("ISP"); PSA.S_flag=1; PSA.insert_avp("Cookie"); Tx:PSA(); RtxTimerStart();
Rx:PSR && !PSR.exist_avp ("EAP-Payload") && (MOBILITY==Unset !resume_pana_sa()) && PSR.S_flag==1 && SEPARATE==Set && !PSR.exist_avp ("Cookie")	RtxTimerStop(); if (choose_isp()) PSA.insert_avp("ISP"); PSA.S_flag=1; Tx:PSA();

<pre> Rx:PSR && !PSR.exist_avp ("EAP-Payload") && (MOBILITY==Unset !resume_pana_sa()) && (PSA.S_flag!=1 SEPARATE==Unset) && PSR.exist_avp ("Cookie") </pre>	<pre> RtxTimerStop(); if (choose_isp()) PSA.insert_avp("ISP"); PSA.insert_avp("Cookie"); Tx:PSA(); RtxTimerStart(); SEPARATE=Unset; </pre>
<pre> Rx:PSR && !PSR.exist_avp ("EAP-Payload") && (MOBILITY==Unset !resume_pana_sa()) && (PSA.S_flag!=1 SEPARATE==Unset) && !PSR.exist_avp ("Cookie") </pre>	<pre> RtxTimerStop(); if (choose_isp()) PSA.insert_avp("ISP"); Tx:PSA(); SEPARATE=Unset; </pre>
AUTH_USER (1) (OFFLINE → OFFLINE)	
AUTH_USER	<pre> Tx:PDI(); RtxTimerStart(); </pre>
EAP_RESPONSE (1) (WAIT EAP MSG IN DISC → WAIT PAA)	
EAP_RESPONSE	<pre> PSA.insert_avp ("EAP-Payload") Tx:PSA(); </pre>
EAP_INVALID_MSG (1) (WAIT EAP MSG IN DISC → OFFLINE)	
EAP_INVALID_MSG	None();
Rx: PAR (1) (WAIT PAA → WAIT EAP MSG)	
Rx:PAR	<pre> RtxTimerStop(); TxEAP(); PANA_SA_RESUMED=Unset; </pre>
Rx: PFER_NO_ABORT_ON (3) (WAIT PAA → WAIT 1ST EAP RESULT)	
<pre> Rx:PFER && 1ST_EAP==Unset && SEPARATE==Set && PFER.RESULT_CODE== PANA_SUCCESS && PFER.S_flag==1 </pre>	<pre> 1ST_EAP=Success; TxEAP(); </pre>
<pre> Rx:PFER && 1ST_EAP==Unset && SEPARATE==Set && PFER.RESULT_CODE!= PANA_SUCCESS && PFER.S_flag==1 && ABORT_ON_1ST_EAP_FAILURE ==Unset && PFER.exit_avp ("EAP-Payload") </pre>	<pre> 1ST_EAP=Failure; TxEAP(); </pre>

<pre> Rx:PFER && 1ST_EAP==Unset && SEPARATE==Set && PFER.RESULT_CODE!= PANA_SUCCESS && PFER.S_flag==1 && ABORT_ON_1ST_EAP_FAILURE ==Unset && !PFER.exit_avp ("EAP-Payload") </pre>	<pre> 1ST_EAP=Failure; alt_reject(); </pre>
Rx: PFER_ABORT_ON (2) (WAIT_PAA → WAIT_1ST_EAP_RESULT)	
<pre> Rx:PFER && 1ST_EAP==Unset && SEPARATE==Set && PFER.RESULT_CODE!= PANA_SUCCESS && (PFER.S_flag==0 ABORT_ON_1ST_EAP_FAILURE ==Set) && PFER.exist_avp ("EAP-Payload") </pre>	<pre> 1ST_EAP=Failure; TxEAP(); </pre>
<pre> Rx:PFER && 1ST_EAP==Unset && SEPARATE==Set && PFER.RESULT_CODE!= PANA_SUCCESS && (PFER.S_flag==0 ABORT_ON_1ST_EAP_FAILURE ==Set) && !PFER.exist_avp ("EAP-Payload") </pre>	<pre> 1ST_EAP=Failure; alt_reject(); </pre>
Rx: PBR_RESUMED_SUCCESS (6) (WAIT_PAA → WAIT_EAP_RESULT)	
<pre> Rx:PBR && 1ST_EAP==Unset && SEPARATE==Unset && PBR.RESULT_CODE== PANA_SUCCESS && PANA_SA_RESUMED!=Set </pre>	<pre> TxEAP(); </pre>
<pre> Rx:PBR && 1ST_EAP==Success && PBR.RESULT_CODE== PANA_SUCCESS && PBR.exist_avp ("EAP-Payload"); </pre>	<pre> TxEAP(); </pre>
<pre> Rx:PBR && 1ST_EAP==Success && PBR.RESULT_CODE== PANA_SUCCESS && !PBR.exist_avp ("EAP-Payload"); </pre>	<pre> alt_reject(); </pre>

Rx:PBR && 1ST_EAP==Success && PBR.RESULT_CODE!= PANA_SUCCESS && PBR.exist_avp ("EAP-Payload")	TxEAP();
Rx:PBR && 1ST_EAP==Success && PBR.RESULT_CODE!= PANA_SUCCESS && !PBR.exist_avp ("EAP-Payload")	alt_reject();
Rx:PBR && 1ST_EAP==Failure && PBR.RESULT_CODE== PANA_SUCCESS	TxEAP();
Rx:PBR_MAC_NONCE_KEYID(1) (WAIT PAA → OPEN)	
Rx:PBR && 1ST_EAP==Unset && SEPARATE==Unset && PBR.RESULT_CODE==PANA_SUCCESS && PANA_SA_RESUMED==Set && PBR.exist_avp("Nonce") && PBR.exist_avp("Key-Id") && PBR.exist_avp("MAC")	PBA.insert_avp("Key-Id"); PBA.insert_avp("MAC"); TxPBA(); Authorize(); SessionTimerStart();
Rx:PBR_NO_SUCCESS(4) (WAIT PAA → WAIT EAP RESULT CLOSE)	
Rx:PBR && 1ST_EAP==Unset && SEPARATE==Unset && PBR.RESULT_CODE!= PANA_SUCCESS && PBR.exist_avp ("EAP-Payload")	TxEAP();
Rx:PBR && 1ST_EAP==Unset && SEPARATE==Unset && PBR.RESULT_CODE!= PANA_SUCCESS && !PBR.exist_avp ("EAP-Payload")	alt_reject();
Rx:PBR && 1ST_EAP==Failure && PBR.RESULT_CODE!= PANA_SUCCESS && PBR.exist_avp ("EAP-Payload")	TxEAP();
Rx:PBR && 1ST_EAP==Failure && PBR.RESULT_CODE!= PANA_SUCCESS && !PBR.exist_avp ("EAP-Payload")	alt_reject()

EAP_RESP_FAIL_SUCC_INV(2) (WAIT EAP MSG → WAIT PAA)	
EAP_RESPONSE	if (key_available()) PAN.insert_avp("MAC"); PAN.S_flag=PAR.S_flag; PAN.N_flag=PAR.N_flag; Tx: PAN();
EAP_INVALID_MSG EAP_SUCCESS EAP_FAILURE	None();
EAP_SUCC_FAIL(3) (WAIT EAP RESULT → OPEN)	
EAP_SUCCESS && PBR.exist_avp ("Key-Id")	PBA.insert_avp("MAC"); PBA.insert_avp("Key-Id"); Tx:PBA(); Authorize(); SessionTimerStart();
EAP_SUCCESS && !PBR.exist_avp("Key-Id")	if (key_available()) PBA.insert_avp("MAC"); Tx:PBA(); SessionTimerStart(); Authorize();
EAP_FAILURE	if (key_available()) PBA.insert_avp("MAC"); Tx:PBA();
EAP_INVALID_MSG_RES(1) (WAIT EAP RESULT → WAIT PAA)	
EAP_INVALID MSG	None();
EAP_SUCC_FAIL_CLOSED(3) (WAIT EAP RESULT CLOSE → CLOSED)	
EAP_SUCCESS && PBR.exist_avp("Key-Id")	PBA.insert_avp("MAC"); PBA.insert_avp("Key-Id"); Tx:PBA(); Disconnect();
EAP_SUCCESS && !PBR.exist_avp ("Key-Id")	if (key_available()) PBA.insert_avp("MAC"); Tx:PBA(); Disconnect();
EAP_FAILURE	Tx:PBA(); Disconnect();
EAP_INVALID_MSG_CLOSE(1) (WAIT EAP RESULT CLOSE → WAIT PAA)	
EAP_INVALID MSG	None();
EAP_FAIL_SUCC_KEYID_INV(3) (WAIT 1ST EAP RESULT → WAIT PAA)	
EAP_SUCCESS && PFER.exist_avp ("Key-Id")	PFEA.insert_avp("Key-Id"); PFEA.S_flag=1; PFEA.N_flag=PFER.N_flag; PFEA.insert_avp("MAC"); Tx:PFEA();

(EAP_SUCCESS && !PFER.exist_avp ("Key-Id")) EAP_FAILURE	if (key_available()) PFEA.insert_avp("MAC"); PFEA.S_flag=1; PFEA.N_flag=PFER.N_flag; Tx:PFEA();
EAP_INVALID_MSG	None();
EAP_SUCC_FAIL_CLOSE(3) (WAIT 1ST EAP RESULT CLOSE → CLOSED)	
EAP_SUCCESS && PFER.exist_avp ("Key-Id")	PFEA.insert_avp("Key-Id"); PFEA.S_flag=0; PFEA.N_flag=0; PFEA.insert_avp("MAC"); Tx:PFEA(); Disconnect();
(EAP_SUCCESS && !PFER.exist_avp ("Key-Id")) EAP_FAILURE	if (key_available()) PFEA.insert_avp("MAC"); PFEA.S_flag=0; PFEA.N_flag=0; Tx:PFEA(); Disconnect();
EAP_INVALID_MSG_1ST(1) (WAIT 1ST EAP RESULT CLOSE → WAIT PAA)	
EAP_INVALID_MSG	None();
Rx: PRAR(1) (OPEN → OPEN)	
Rx:PRAR	if (key_available()) PRAA.insert_avp("MAC"); Tx:PRAA();
FAST_REAUTH(1) (OPEN → WAIT PRAA)	
FAST_REAUTH	if (key_available()) PRAR.insert_avp("MAC"); Tx:PRAR(); RtxTimerStart();
EAP_REAUTH(1) (OPEN → WAIT PAA)	
EAP_REAUTH	PDI.insert_avp("Session-Id"); RtxTimerStart(); 1ST_EAP=Unset; PANA_SA_RESUMED=Unset; Tx:PDI(); RtxTimerStart();
Rx: PAR_OPEN(1) (OPEN → WAIT EAP MSG)	
Rx:PAR	SEPARATE=Set Unset; 1ST_EAP=Unset; PANA_SA_RESUMED=Unset; TxEAP();
Rx: PTR(1) (OPEN → CLOSED)	
Rx:PTR	if (key_available()) PTA.insert_avp("MAC"); Tx:PTA(); Disconnect();
TERMINATE(1) (OPEN → SESS_TERM)	

TERMINATE	<pre> if (key_available()) PTR.insert_avp("MAC"); Tx:PTR(); RtxTimerStart(); </pre>
UPDT_POPA_DEVID (1) (OOPEN → WAIT_PAUA)	
UPDATE_DEVICE_ID UPDATE_POPA	<pre> if (UPDATE_DEVICE_ID) PAUR.insert_avp ("Device-Id"); if (UPDATE_POPA) PAUR.insert_avp ("IP-Address"); if (key_available()) PAUR.insert_avp("MAC"); Tx:PAUR(); RtxTimerStart(); </pre>
Rx: PRAA (1) (WAIT_PRAA → OPEN)	
Rx:PRAA	None();
Rx: PAUA (1) (WAIT_PAUA → OPEN)	
Rx:PAUA	RtxTimerStop();
Rx: PTA (1) (SESS_TERM → CLOSED)	
Rx:PTA	Disconnect();