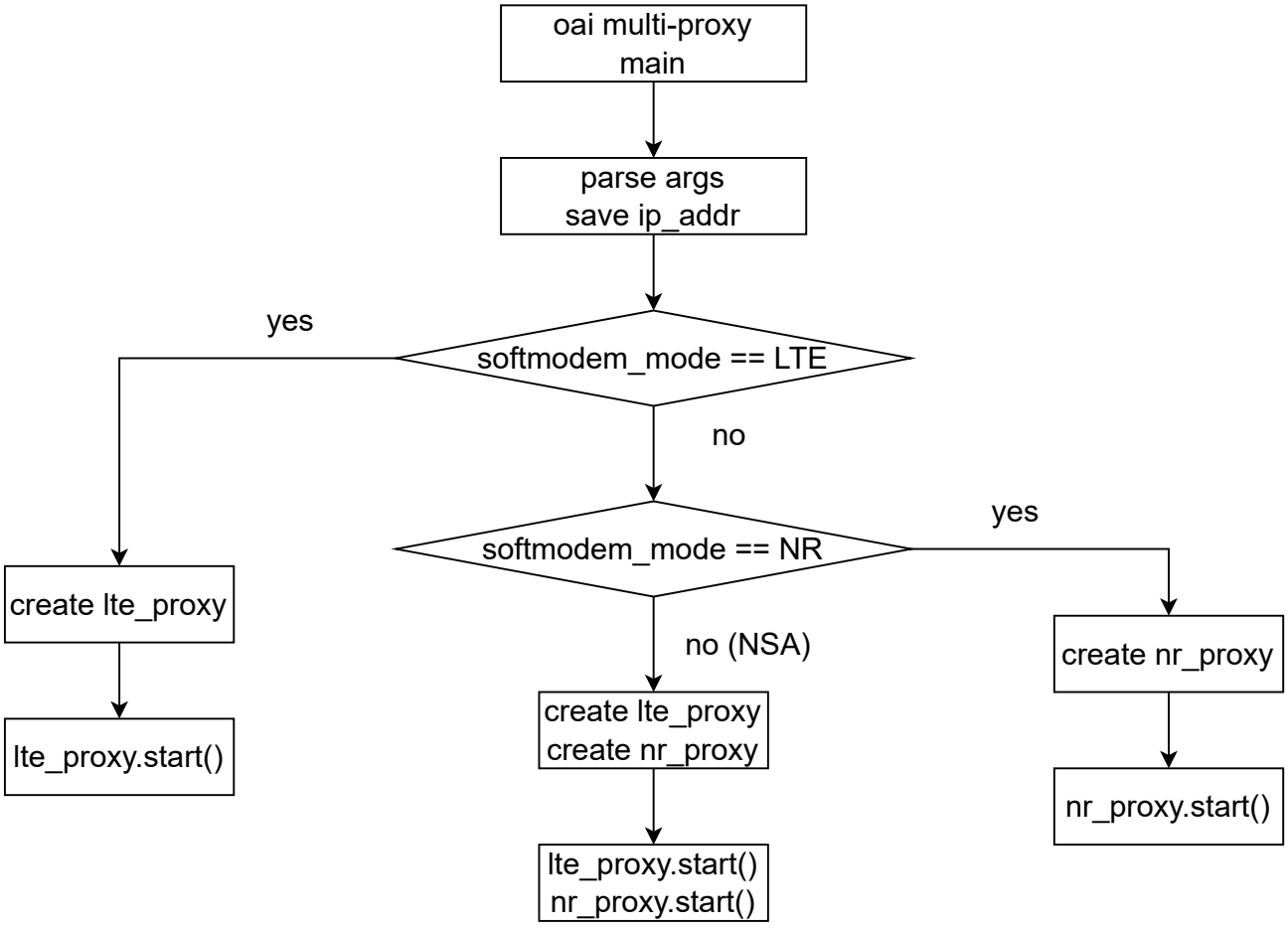
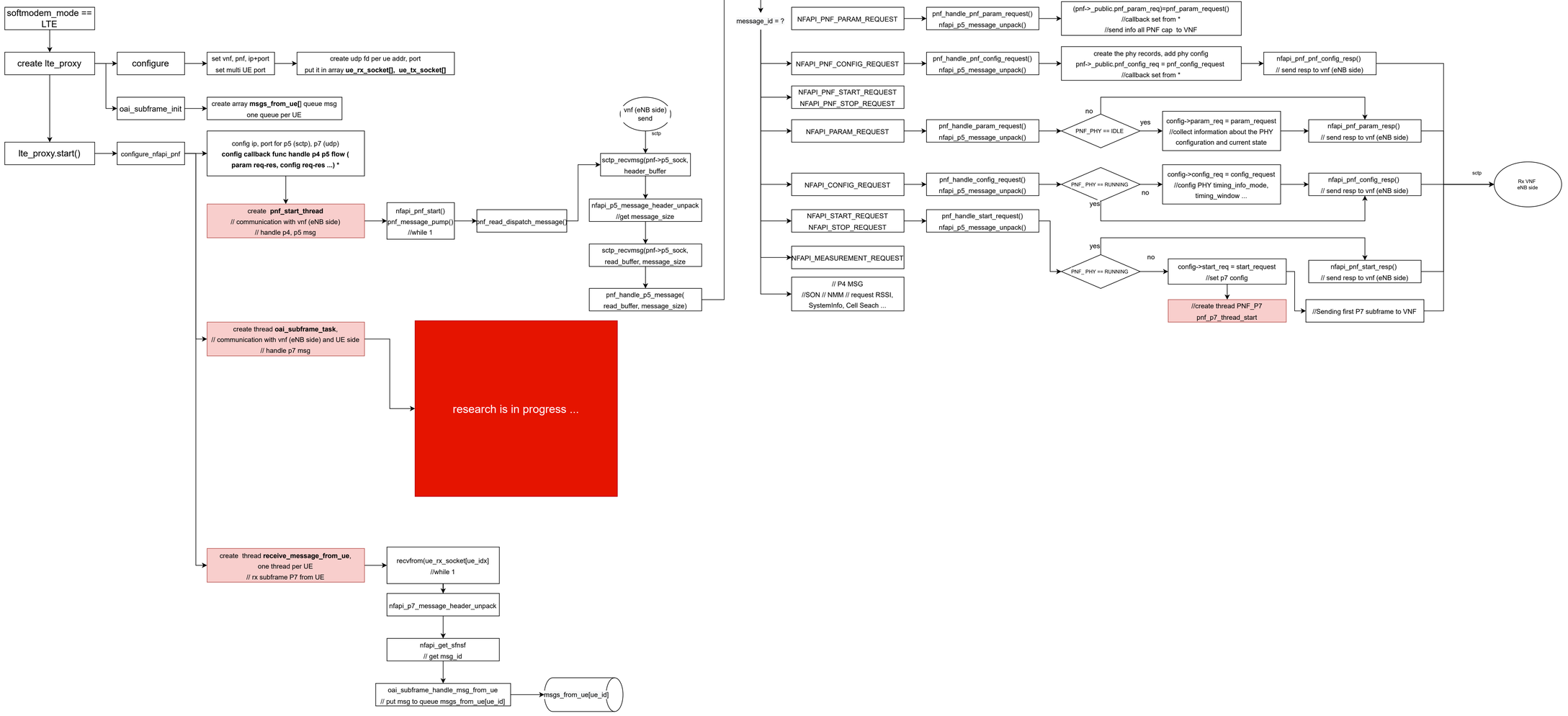
OAI multi-proxy



LTE mode


softmodem_mode == LTE
create lte_proxy
configure
set vnf, pnf, ip+port
set multi UE port
create udp fd per ue addr, port
put it in array ue_rx_socket[], ue_tx_socket[]
oai_subframe_init
create array msgs_from_ue[] queue msg
one queue per UE
lte_proxy.start()
configure_nfapi_pnf
config ip, port for p5 (sctp), p7 (udp)
config callback func handle p4 p5 flow (param req-res, config req-res ...) *
create pnf_start_thread
// communication with vnf (eNB side)
// handle p4, p5 msg
nfapi_pnf_start()
pnf_message_pump()
//while 1
pnf_read_dispatch_message()
vnf (eNB side)
send
sctp
sctp_recvmsg(pnf->p5_sock, header_buffer
nfapi_p5_message_header_unpack
//get message_size
sctp_recvmsg(pnf->p5_sock, read_buffer, message_size
pnf_handle_p5_message(read_buffer, message_size)
messageHeader.message_id = nfapi_p5_message_header_unpack
message_id = ?
NFAPI_PNF_PARAM_REQUEST
pnf_handle_pnf_param_request()
nfapi_p5_message_unpack()
(pnf->_public.pnf_param_req)=pnf_param_request()
//callback set from *
//send info all PNF cap to VNF
NFAPI_PNF_CONFIG_REQUEST
pnf_handle_pnf_config_request()
nfapi_p5_message_unpack()
create the phy records, add phy config
pnf->_public.pnf_config_req = pnf_config_request
//callback set from *
nfapi_pnf_pnf_config_resp()
// send resp to vnf (eNB side)
NFAPI_PNF_START_REQUEST
NFAPI_PNF_STOP_REQUEST
NFAPI_PARAM_REQUEST
pnf_handle_param_request()
nfapi_p5_message_unpack()
no
PNF_PHY == IDLE
yes
config->param_req = param_request
//collect information about the PHY configuration and current state
nfapi_pnf_param_resp()
// send resp to vnf (eNB side)
NFAPI_CONFIG_REQUEST
pnf_handle_config_request()
nfapi_p5_message_unpack()
PNF_ PHY == RUNNING
no
yes
config->config_req = config_request
//config PHY timing_info_mode, timing_window ...
nfapi_pnf_config_resp()
// send resp to vnf (eNB side)
NFAPI_START_REQUEST
NFAPI_STOP_REQUEST
pnf_handle_start_request()
nfapi_p5_message_unpack()
yes
PNF_ PHY == RUNNING
no
config->start_req = start_request
//set p7 config
nfapi_pnf_start_resp()
// send resp to vnf (eNB side)
//create thread PNF_P7
pnf_p7_thread_start
//Sending first P7 subframe to VNF
NFAPI_MEASUREMENT_REQUEST
// P4 MSG
//SON // NMM // request RSSI, SystemInfo, Cell Seach ...
sctp
Rx VNF
eNB side
create thread oai_subframe_task,
// communication with vnf (eNB side) and UE side
// handle p7 msg
research is in progress ...
create thread receive_message_from_ue,
one thread per UE
// rx subframe P7 from UE
recvfrom(ue_rx_socket[ue_idx]
//while 1
nfapi_p7_message_header_unpack
nfapi_get_sfnsf
// get msg_id
oai_subframe_handle_msg_from_ue
// put msg to queue msgs_from_ue[ue_id]
msgs_from_ue[ue_id]