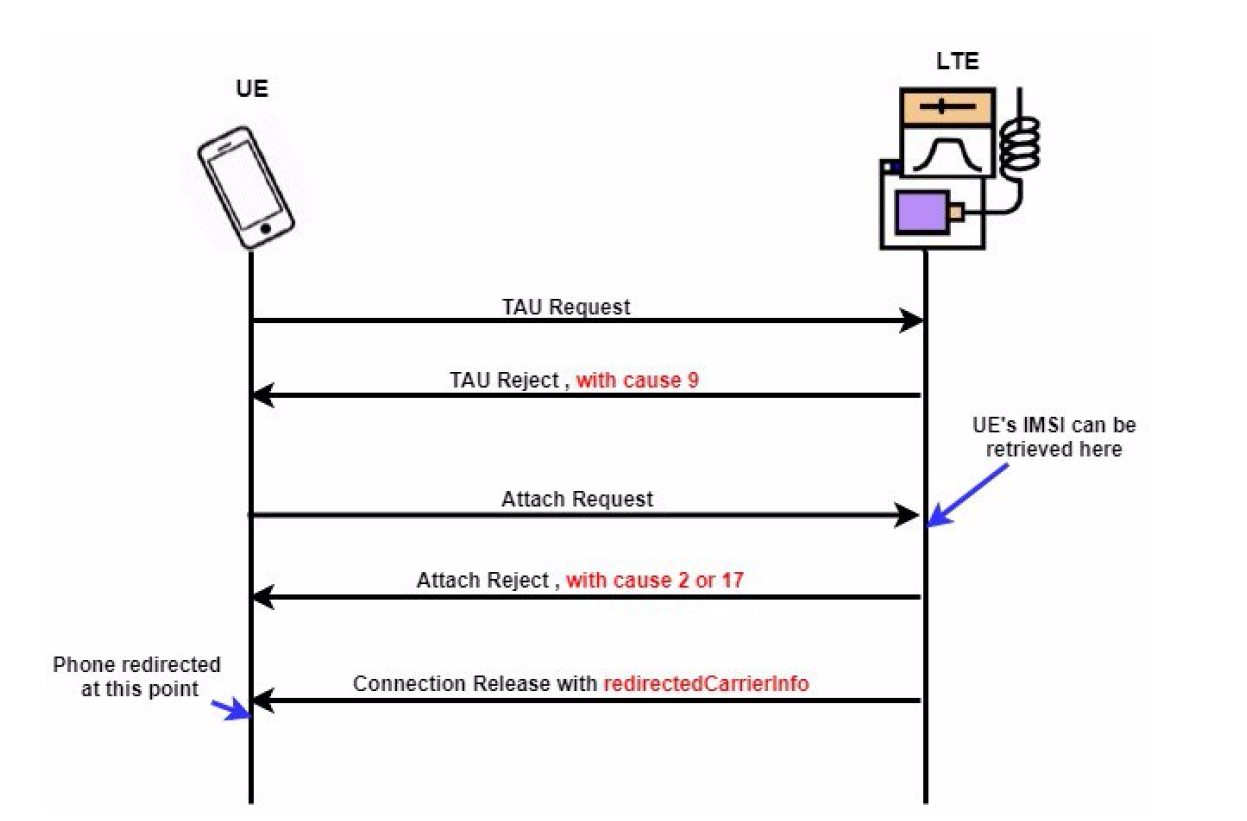
[Open5GS and OAI gNB | openverso-charts (gradiant.github.io)](https://gradiant.github.io/openverso-charts/open5gs-oaignb.html)

[Hakim badis's Home Page (univ-mlv.fr)](http://igm.univ-mlv.fr/~badis/)

[doc/TESTBenches.md · develop · oai / openairinterface5G · GitLab (eurecom.fr)](https://gitlab.eurecom.fr/oai/openairinterface5g/-/blob/develop/doc/TESTBenches.md)

[ci-scripts/conf\_files/du.band7.tm1.100PRB.usrpb210.conf · develop · oai / openairinterface5G · GitLab (eurecom.fr)](https://gitlab.eurecom.fr/oai/openairinterface5g/-/blob/develop/ci-scripts/conf_files/du.band7.tm1.100PRB.usrpb210.conf)

LTE REDIRECTION : TAU REJECT FLOW  + ATTACH REJECT OR SERVICE REJECT



https://gitlab.modalai.com/srs-opensource/srsLTE/blob/master/lib/include/srslte/asn1/liblte\_mme.h

Changer : LIBLTE\_MME\_EMM\_CAUSE\_IMPLICITLY\_DETACHED 0x0A en LIBLTE\_MME\_EMM\_CAUSE\_UE\_IDENTITY\_CANNOT\_BE\_DERIVED\_BY\_THE\_NETWORK 0x09

AUTRES CODES A CONNAITRE :

LIBLTE\_MME\_EMM\_CAUSE\_IMSI\_UNKNOWN\_IN\_HSS 0x02

LIBLTE\_MME\_EMM\_CAUSE\_ILLEGAL\_UE 0x03

LIBLTE\_MME\_EMM\_CAUSE\_NETWORK\_FAILURE 0x11

SRSLTE

------

https://github.com/srsran/srsRAN\_4G/blob/master/srsepc/src/mme/nas.cc

tau\_rej.emm\_cause

service\_rej.emm\_cause

à modifier bastien error :

https://github.com/bbaranoff/srslte\_to\_gsm/blob/main/redirection\_srslte\_gsm.patch

https://github.com/bbaranoff/srslte\_to\_gsm

OPEN LTE:

---------

attach\_rej.emm\_cause

+ LIBLTE\_MME\_TRACKING\_AREA\_UPDATE\_REJECT\_MSG\_STRUCT ta\_update\_rej;

+ LIBLTE\_BYTE\_MSG\_STRUCT msg;

+ ta\_update\_rej.emm\_cause = user->get\_emm\_cause();

Bastien mandeha tsara :

https://github.com/bbaranoff/telco\_story

OAI :

-----

TAU Reject :

------------

<https://github.com/brchiu/openair-cn-new/tree/master/src/nas>

<https://github.com/lfarizav/openair-cn>

openair-cn/SRC/NAS/EMM/nas\_proc.c:

openair-cn/blob/develop/src/nas/emm/TrackingAreaUpdate.c :

i f ( ue\_ctx ) {

r c = emm\_proc\_tracking\_area\_update\_re ject

( ue\_id , EMM\_CAUSE\_IMPLICITLY\_DETACHED) ;

OAILOG\_FUNC\_RETURN (LOG\_NAS\_EMM, r c ) ;

} e l s e {

r c = emm\_proc\_tracking\_area\_update\_re ject

( ue\_id , EMM\_CAUSE\_IMPLICITLY\_DETACHED) ;

OAILOG\_FUNC\_RETURN (LOG\_NAS\_EMM, r c ) ;

}

SERVICE REJECT :

---------------

<https://github.com/brchiu/openair-cn-new/blob/master/src/nas/nas_proc.c>

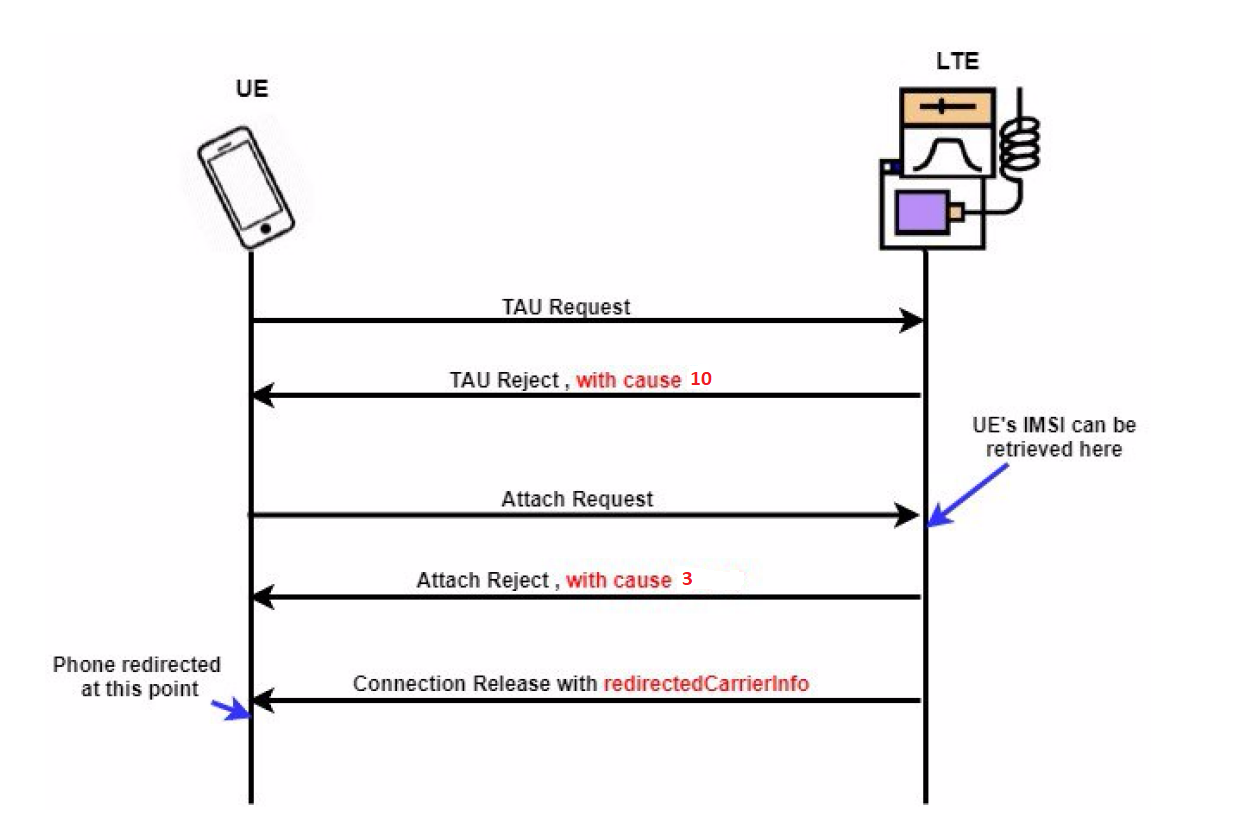
openair-cn/SRC/NAS/nas\_proc.c:

switch ( s6a\_error ) {

case DIAMETER\_AUTHENTICATION\_DATA\_UNAVAILABLE:

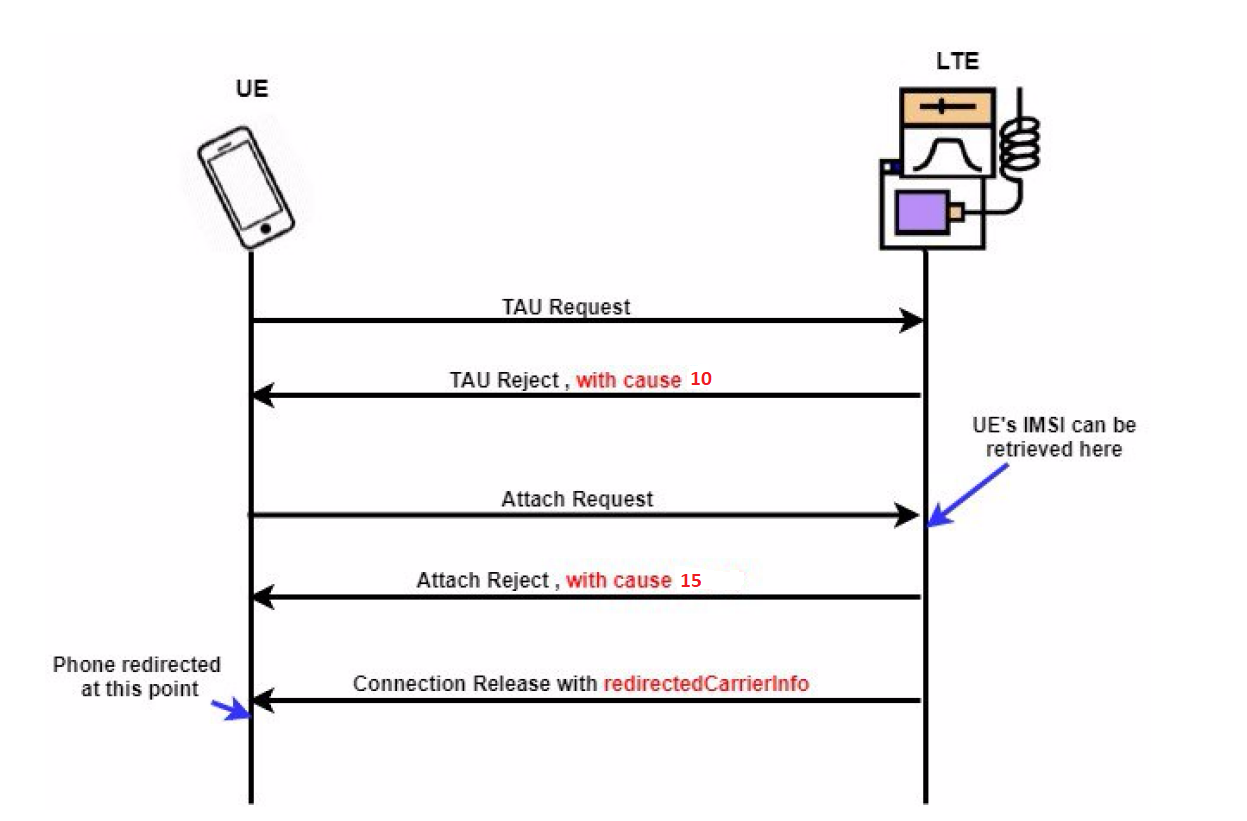
return NAS\_CAUSE\_NO\_SUITABLE\_CELLS\_IN\_TRACKING\_AREA

FOR imsicatcher default : ILLEGAL UE (by default by OAI-5G-NR)



DOS : attach reject : 2,3,7,8,14, 17

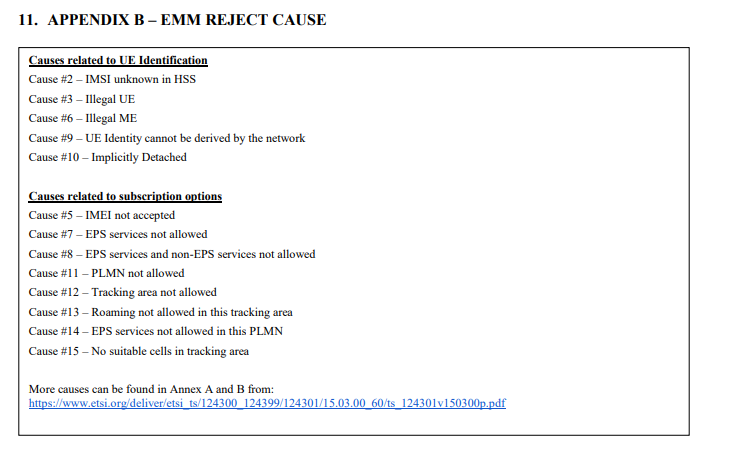
IMSI-CATCHER WITHOUT PERTURBATION CODE FLOW (REAL\_IMSICATCHER):



FOR 5G :

<https://github.com/bbaranoff/redir5Gted2Gsm/blob/main/sib.conf>

[Imsi Catcher de haut niveau avec le suivi GPS - Chine Signal Imsi Catcher, signal brouilleur (made-in-china.com)](https://fr.made-in-china.com/co_ptcjammer/product_High-Level-Imsi-Catcher-with-GPS-Tracking_enueunihy.html)



REDIRECTED OAI 4G :

<https://icepng.github.io/2020/09/23/RRCConnectionRelease/>

<https://www.programmersought.com/article/67481124166/>

UMTS :

https://github.com/RangeNetworks/OpenBTS-UMTS/blob/master/ASN/rrc.asn1

RRCConnectionReject

https://github.com/RangeNetworks/OpenBTS-UMTS/blob/master/UMTS/URRCMessages.cpp

When the RNC decides to reject the RRC connection request due to ICSU overload. After this, the RNC sends an

RRC: RRC CONNECTION REJECT message to the UE.

Search request : https://github.com/RangeNetworks/OpenBTS-UMTS/blob/master/UMTS/URRC.cpp

handleRrcConnectionRequest

// AttachRequest, AttachComplete,

// DetachRequest, DetachAccept (todo),

// RoutingAreaUpdateRequest, RoutingAreaUpdateComplete

FOR HACKING : sending routing are update reject : https://github.com/RangeNetworks/OpenBTS-UMTS/blob/master/SGSNGGSN/GPRSL3Messages.cpp

RRC connection request : https://github.com/RangeNetworks/OpenBTS-UMTS/blob/master/UMTS/URRCMessages.h

https://github.com/RangeNetworks/OpenBTS-UMTS/blob/master/ASN/rrc.asn1

https://github.com/RangeNetworks/OpenBTS-UMTS/blob/master/UMTS/URRCMessages.h

https://sourceforge.net/p/openbts/mailman/message/36090903/

https://github.com/RangeNetworks/OpenBTS-UMTS/blob/master/UMTS/URRCMessages.cpp

https://github.com/RangeNetworks/OpenBTS-UMTS/blob/master/ASN/rrc.asn1

https://github.com/RangeNetworks/OpenBTS-UMTS/tree/master/UMTS

https://github.com/RangeNetworks/OpenBTS-UMTS/blob/master/UMTS/URRC.cpp

https://www.sqimway.com/rrc\_umts.html

https://github.com/RangeNetworks/OpenBTS-UMTS/blob/master/ASN/makefile

https://github.com/RangeNetworks/OpenBTS-UMTS/blob/master/ASN/rrc.asn1

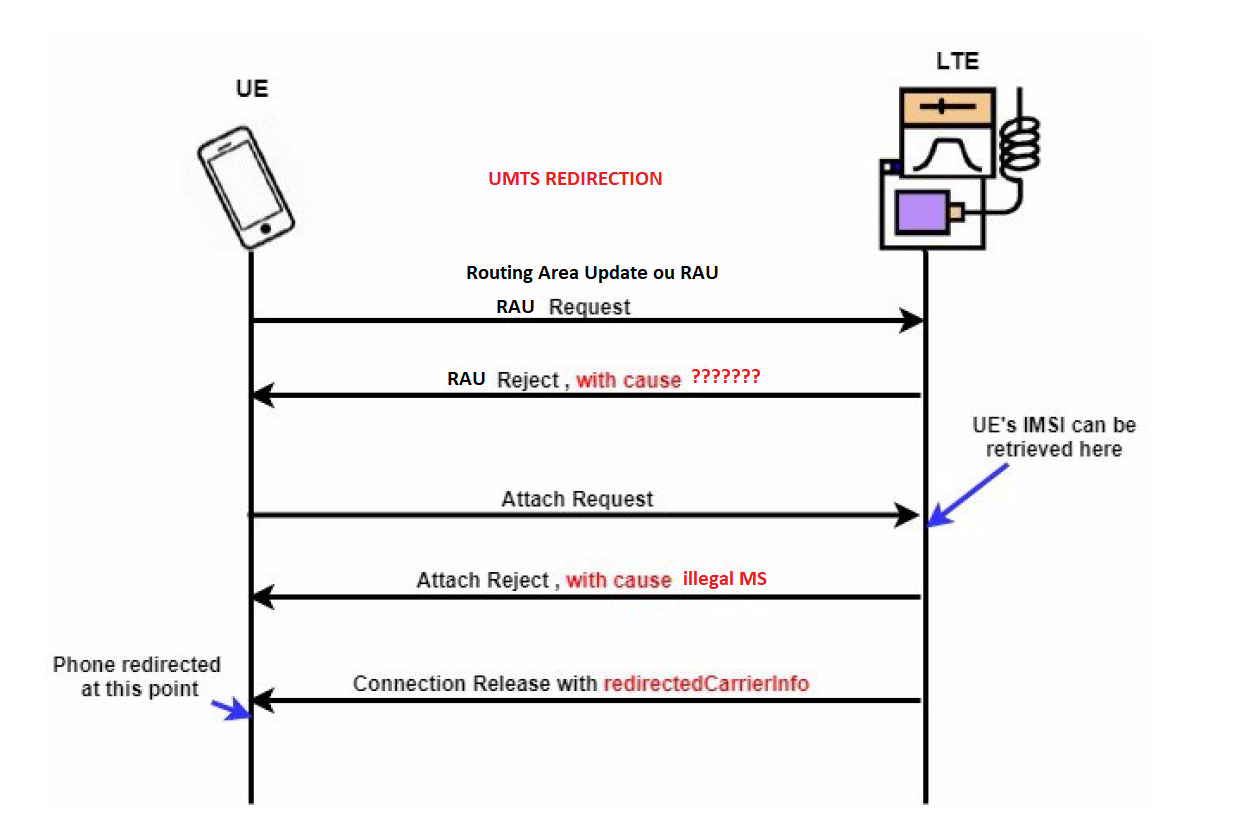
https://www.hackers-arise.com/scada-hacking

https://github.com/RangeNetworks/OpenBTS-UMTS/tree/master/UMTS

https://github.com/RangeNetworks/OpenBTS-UMTS/blob/master/UMTS/URRC.cpp

https://github.com/RangeNetworks/OpenBTS-UMTS/blob/master/UMTS/URRCTrCh.cpp

https://github.com/RangeNetworks/OpenBTS-UMTS/blob/master/UMTS/URRCMessages.cpp



Routing area Reject : Find on google : RoutingAreaUpdateReject + openbts-umts

FOR HACKING : sending routing are update reject :

* <https://github.com/RangeNetworks/OpenBTS-UMTS/blob/master/SGSNGGSN/GPRSL3Messages.cpp>
* <https://github.com/RangeNetworks/OpenBTS-UMTS/blob/master/SGSNGGSN/Sgsn.cpp>
* <https://discourse.myriadrf.org/t/openbts-umts-with-lilme-sdr/2033>

Chercher : sendRrcConnectionRelease dans : <https://github.com/RangeNetworks/OpenBTS-UMTS/blob/master/UMTS/URRCMessages.cpp>

Regarder dans : <https://github.com/RangeNetworks/OpenBTS-UMTS/blob/master/ASN/rrc.asn1> chercher RRCConnectionRelease-r3-IEs Le fichier asn generé etant : <https://github.com/RangeNetworks/OpenBTS-UMTS/blob/master/UMTS/URRCMessages.cpp> avec comme nom : RRCConnectionRelease-r3-IEs.c

CHOIX 1 : REMPLACER TOUS LES RELEASE EN REJECT

Lire le fichier asn et le fichier redirection est dans la fonction : sendRrcConnectionRelease :

ASN::RRCConnectionRelease\_r3\_IEs\_t \*ies = &msg.message.choice.rrcConnectionRelease.choice.r3.rrcConnectionRelease\_r3;

Cet variable ies, il est possible de modifier la cause du rejet et le canal GSM à rediriger dans la partie OPTIONAL de l’asn est dans : RRCConnectionReject-v690ext-IEs

Pour l’inspiration est dans le fichier asn de l’umts dans le lien tout à l’heure :

RRCConnectionReject-r3-IEs ::= SEQUENCE {

|  |
| --- |
| -- TABULAR: Integrity protection shall not be performed on this message. |
| -- User equipment IEs |
| initialUE-Identity InitialUE-Identity, |
| rrc-TransactionIdentifier RRC-TransactionIdentifier, |
| rejectionCause RejectionCause, |
| waitTime WaitTime, |
| redirectionInfo RedirectionInfo OPTIONAL |
| } |
|  |
| RRCConnectionReject-v690ext-IEs ::= SEQUENCE { |
| redirectionInfo-v690ext GSM-TargetCellInfoList OPTIONAL |
| } |

Redirection

// Allocate and set values for the redirect\_info IE

ASN::RedirectInfo\_t \*redirect\_info = new ASN::RedirectInfo\_t();

redirect\_info->set\_redirectingReason(ASN::RedirectingReason(1)); // Set redirecting reason to "User Busy"

redirect\_info->set\_redirectingPartyID("0123456789"); // Set redirecting party ID

redirect\_info->set\_redirectionNumber("9876543210"); // Set redirection number

// Add the redirect\_info IE to the ies structure

ies->redirect\_info = redirect\_info;

::::::::::::::::::::::::::::::

void URRCMessages::sendRrcConnectionRelease(int32\_t ueIdx, uint8\_t releaseCause, uint8\_t redirectInfo, uint16\_t arfcn) {

Debug(\_LAYER\_RRC, "sendRrcConnectionRelease(): entering. UE Index: %d, Release Cause: %d, Redirect Info: %d, ARFCN: %d.\n", ueIdx, releaseCause, redirectInfo, arfcn);

if (g\_UE\_list.find(ueIdx) == g\_UE\_list.end()) {

Error("sendRrcConnectionRelease(): UE Index %d does not exist.\n", ueIdx);

return;

}

UE& ue = g\_UE\_list[ueIdx];

ASN::RRCConnectionReject message;

message.criticalExtensions.present = ASN::RRCConnectionReject\_\_criticalExtensions\_PR\_rrcConnectionReject\_r3;

ASN::RRCConnectionReject\_r3\_IEs\_t \*ies = &message.criticalExtensions.choice.rrcConnectionReject\_r3;

ies->releaseCause.present = ASN::RRCConnectionRelease\_r3\_IEs\_\_releaseCause\_PR\_unspecified;

ies->redirect\_info\_v690ext = NULL; // set redirect\_info\_v690ext to NULL by default

switch (releaseCause) {

case 1: {

ies->releaseCause.present = ASN::RRCConnectionRelease\_r3\_IEs\_\_releaseCause\_PR\_normalRelease;

break;

}

case 2: {

ies->releaseCause.present = ASN::RRCConnectionRelease\_r3\_IEs\_\_releaseCause\_PR\_rrcConnectionReject;

break;

}

case 3: {

ies->releaseCause.present = ASN::RRCConnectionRelease\_r3\_IEs\_\_releaseCause\_PR\_rrcConnectionReleaseWithRedirection;

break;

}

default: {

Error("sendRrcConnectionRelease(): Invalid release cause: %d.\n", releaseCause);

return;

}

}

if (ies->releaseCause.present == ASN::RRCConnectionRelease\_r3\_IEs\_\_releaseCause\_PR\_rrcConnectionReleaseWithRedirection) {

ies->redirect\_info\_v690ext = (ASN::RedirectInformation\_v690ext\_t\*)calloc(1, sizeof(ASN::RedirectInformation\_v690ext\_t));

ies->redirect\_info\_v690ext->redirectedCarrierInfo.carrierFreq = arfcn;

ies->redirect\_info\_v690ext->redirectedCarrierInfo.cellReselectionPriority = 7;

ies->redirect\_info\_v690ext->redirectedCarrierInfo.q\_RxLevMin = -110;

// Set the channel number (EARFCN) for GSM

ies->redirect\_info\_v690ext->redirectedCarrierType.present = ASN::RedirectedCarrierInfo\_v8a0\_IEs\_\_redirectedCarrierType\_PR\_gsm;

ies->redirect\_info\_v690ext->redirectedCarrierType.choice.gsm.buf = (uint8\_t\*)calloc(1, sizeof(uint8\_t));

ies->redirect\_info\_v690ext->redirectedCarrierType.choice.gsm.buf[0] = arfcn - 935;

ies->redirect\_info\_v690ext->redirectedCarrierType.choice.gsm.size = 1;

// Set the redirect info

ies->redirect\_info\_v690ext->redirectingReason.present = ASN::RedirectInformation\_v690ext\_IEs\_\_redirectingReason\_PR\_servingPLMNRateControl;

ies->redirect\_info\_v

<https://github.com/RangeNetworks/OpenBTS-UMTS/blob/master/UMTS/URRC.h>

Modifier le code : sendRrcConnectionRelease en sendRrcConnectionReject

CHOIX 2 : ajouter dans la fonction sendRrcConnectionRelease la redirection

Regarder dans asn de RRCConnectionRelease dans la partie : laterNonCriticalExtensions

<https://www.sqimway.com/rrc_umts.html>

bool URRC::sendRrcConnectionRelease(GSM::Time ms, uint8\_t trx, uint8\_t ts, uint8\_t ulTsOffset, uint32\_t channel, uint32\_t mobile\_id, bool abort, uint8\_t wait)

{

ASN::RRCConnectionReject \*rrc = new ASN::RRCConnectionReject();

ASN::RRCConnectionReject\_r3\_IEs\_t \*ies = &rrc->value.u.rrcConnectionReject\_r3;

ies->rrc\_TransactionIdentifier = transactionNumber;

transactionNumber = (transactionNumber+1)&7;

ies->wait\_Time = wait;

if (abort) {

ies->criticalExtensions.present = ASN::RRCConnectionReject\_r3\_IEs\_\_criticalExtensions\_PR\_c1;

ies->criticalExtensions.choice.c1.present = ASN::RRCConnectionReject\_r3\_IEs\_\_criticalExtensions\_\_c1\_PR\_rrcConnectionReject\_r3;

ies->criticalExtensions.choice.c1.choice.rrcConnectionReject\_r3.rrc\_TransactionIdentifier = ies->rrc\_TransactionIdentifier;

ies->criticalExtensions.choice.c1.choice.rrcConnectionReject\_r3.wait\_Time = wait;

}

bool success = false;

uint8\_t \*buffer;

uint32\_t buffer\_len;

if (rrc->toByteArray(&buffer, buffer\_len)) {

success = sendRrc(rrc->messageId(), buffer, buffer\_len, ms, trx, ts, ulTsOffset, channel, mobile\_id);

}

free(buffer);

delete rrc;

return success;

}

ASN::RRCConnectionReject\_r3\_IEs\_t \*ies = &reject->u.r3.protocolSpecific.rejectionSpecificInformation.u.spare3;

ies->redirect\_info = NULL; // Add redirect\_info here

ASN\_SEQUENCE\_ADD(&ies->spare2, &spare2\_member);

// Add GSM channel ARFCN number 11

ASN::GSM\_ChannelList\_t \*gsm\_channel\_list = &ies->redirect\_info->cell\_change\_order;

gsm\_channel\_list->present = ASN::GSM\_ChannelList\_t\_PR\_gsm\_ARFCN\_List;

gsm\_channel\_list->choice.gsm\_ARFCN\_List = (ASN::GSM\_ARFCN\_List\_t\*)calloc(1, sizeof(\*gsm\_channel\_list->choice.gsm\_ARFCN\_List));

gsm\_channel\_list->choice.gsm\_ARFCN\_List->list.count = 1;

gsm\_channel\_list->choice.gsm\_ARFCN\_List->list.array[0] = 11;

CHATGPT :

void sendRrcConnectionRelease(UEInfo \*uep) {

// Create the RB Setup Message.

ASN::DL\_DCCH\_Message\_t msg;

memset(&msg, 0, sizeof(msg));

msg.message.present = ASN::DL\_DCCH\_MessageType\_PR\_rrcConnectionRelease;

msg.message.choice.rrcConnectionRelease.present = ASN::RRCConnectionRelease\_PR\_r3;

ASN::RRCConnectionRelease\_r3\_IEs\_t \*ies =

&msg.message.choice.rrcConnectionRelease.choice.r3.rrcConnectionRelease\_r3;

// Set the transaction ID

unsigned transactionId = uep->newTransactionId();

ies->rrc\_TransactionIdentifier = transactionId;

// Set the release cause to normal event

ies->releaseCause = toAsnEnumerated(ASN::ReleaseCause\_normalEvent);

// Set the non-critical extensions

ies->m.v690NonCriticalExtensionsPresent = true;

// Add the RRCConnectionRelease-v690ext-IEs extension

ies->v690NonCriticalExtensions.rrcConnectionRelease\_r6\_add\_extPresent = true;

// Add the RedirectionInfo-r6 extension

ASN::RedirectionInfo\_r6\_t &redirectionInfo = ies->v690NonCriticalExtensions.rrcConnectionRelease\_r6\_add\_ext.redirectionInfo\_r6;

redirectionInfo.m.criticalExtensionsPresent = false;

redirectionInfo.redirectionType = ASN::RedirectionType\_geranToUtran;

// Add the FrequencyInfo extension

ASN::FrequencyInfo\_t &frequencyInfo = redirectionInfo.frequencyInfo;

frequencyInfo.m.frequencyInfoTDDPresent = false;

frequencyInfo.m.frequencyInfoFDDPresent = true;

frequencyInfo.frequencyInfoFDD.ul\_CarrierFrequency = uep->ulFrequency();

frequencyInfo.frequencyInfoFDD.dl\_CarrierFrequency = uep->dlFrequency();

// Add the InterRATInfo-r6 extension

ASN::InterRATInfo\_r6\_t &interRATInfo = redirectionInfo.interRAT\_Info\_r6;

interRATInfo.m.halfRateInterRAT\_ParamsPresent = false;

interRATInfo.m.fddInterRAT\_ParamsPresent = true;

interRATInfo.fddInterRAT\_Params.interRAT\_Handover\_Info.m.criticalExtensionsPresent = false;

interRATInfo.fddInterRAT\_Params.interRAT\_Handover\_Info.gsm\_TargetCellInfoList.m.cellParametersIdPresent = true;

interRATInfo.fddInterRAT\_Params.interRAT\_Handover\_Info.gsm\_TargetCellInfoList.cellParametersId = 1;

interRATInfo.fddInterRAT\_Params.interRAT\_Handover\_Info.gsm\_TargetCellInfoList.m.arfcnValuePresent = true;

interRATInfo.fddInterRAT\_Params.interRAT\_Handover\_Info.gsm\_TargetCellInfoList.arfcnValue = uep->dlFrequency();

ByteVector result(1000);

if (!encodeDcchMsg(uep, SRB2, &msg, result, descrRrcConnectionRelease)) {

return;

}

// Prepare to receive the reply to this message:

UeTransaction(uep, UeTransaction::ttRrcConnectionRelease, 0, transactionId, stIdleMode);

uep->ueWriteHighSide(SRB2, result, descrRrcConnectionRelease);

}

OTHER :

void sendRrcConnectionRelease(UEInfo \*uep) {

// Create the RB Setup Message.

ASN::DL\_DCCH\_Message\_t msg;

memset(&msg, 0, sizeof(msg));

msg.message.present = ASN::DL\_DCCH\_MessageType\_PR\_rrcConnectionRelease;

msg.message.choice.rrcConnectionRelease.present = ASN::RRCConnectionRelease\_PR\_r3;

ASN::RRCConnectionRelease\_r3\_IEs\_t \*ies =

&msg.message.choice.rrcConnectionRelease.choice.r3.rrcConnectionRelease\_r3;

// RRC\_TransactionIdentifier\_t rrc\_TransactionIdentifier;

unsigned transactionId = uep->newTransactionId();

ies->rrc\_TransactionIdentifier = transactionId;

// N\_308\_t \*n\_308 /\* OPTIONAL \*/;

// N308 is the number of times UE sends response RrcConnectionReleaseComplete.

// It is mandatory when this message is sent on DCCH.

ies->n\_308 = RN\_CALLOC(ASN::N\_308\_t); // It is a long.

\*ies->n\_308 = 1; // must be 1..8

// ReleaseCause\_t releaseCause;

// Causes we might use are: normalEvent, userInactivity, pre\_emptiveRelease.

ies->releaseCause = toAsnEnumerated(ASN::ReleaseCause\_normalEvent);

// Add extensions

ies->rrcConnectionRelease\_r3\_add\_ext = RN\_CALLOC(ASN::RRCConnectionRelease\_r3\_add\_ext\_IEs\_t);

ies->rrcConnectionRelease\_r3\_add\_ext->present = ASN::RRCConnectionRelease\_r3\_add\_ext\_IEs\_PR\_c1;

// Add v690NonCriticalExtensions

ASN::RRCConnectionRelease\_v690ext\_IEs\_t \*v690ext = &ies->rrcConnectionRelease\_r3\_add\_ext->choice.c1.v690ext\_IEs;

v690ext->present = ASN::RRCConnectionRelease\_v690ext\_IEs\_PR\_c1;

v690ext->choice.c1.present = ASN::RRCConnectionRelease\_v690ext\_IEs\_\_c1\_PR\_redirectionInfo\_r6;

// Add RedirectionInfo-r6

ASN::RedirectionInfo\_r6\_t \*redirectionInfo = &v690ext->choice.c1.choice.redirectionInfo\_r6;

redirectionInfo->present = ASN::RedirectionInfo\_r6\_PR\_redirection\_info\_r6;

// Add FrequencyInfo

ASN::FrequencyInfo\_t \*frequencyInfo = &redirectionInfo->frequencyInfo;

frequencyInfo->present = ASN::FrequencyInfo\_PR\_fdd;

// Add InterRATInfo-r6

ASN::InterRATInfo\_r6\_t \*interRatInfo = &redirectionInfo->interRAT\_Info\_r6;

interRatInfo->present = ASN::InterRATInfo\_r6\_PR\_geranToUtranInfo\_r6;

// Add GSM-TargetCellInfoList

ASN::GSM\_TargetCellInfoList\_t \*targetCellList = &interRatInfo->choice.geranToUtranInfo\_r6.gsm

void sendRrcConnectionRelease(UEInfo \*uep) {

// Create the RB Setup Message.

ASN::DL\_DCCH\_Message\_t msg;

memset(&msg, 0, sizeof(msg));

msg.message.present = ASN::DL\_DCCH\_MessageType\_PR\_rrcConnectionRelease;

msg.message.choice.rrcConnectionRelease.present = ASN::RRCConnectionRelease\_PR\_r3;

ASN::RRCConnectionRelease\_r3\_IEs\_t \*ies = &msg.message.choice.rrcConnectionRelease.choice.r3.rrcConnectionRelease\_r3;

// Set the RRC transaction identifier.

unsigned transactionId = uep->newTransactionId();

ies->rrc\_TransactionIdentifier = transactionId;

// Set the release cause to normal event.

ies->releaseCause = toAsnEnumerated(ASN::ReleaseCause\_normalEvent);

// Set the number of times the UE should send a response to this message.

ies->n\_308 = RN\_CALLOC(ASN::N\_308\_t);

\*ies->n\_308 = 1;

// Add the R3 extensions to the message.

ies->criticalExtensions.present = ASN::RRCConnectionRelease\_r3\_IEs\_\_criticalExtensions\_PR\_c1;

ies->criticalExtensions.choice.c1.present = ASN::RRCConnectionRelease\_r3\_IEs\_\_criticalExtensions\_\_c1\_PR\_rrcConnectionRelease\_r3\_add\_ext;

ASN::RRCConnectionRelease\_r3\_IEs\_\_criticalExtensions\_\_c1\_\_rrcConnectionRelease\_r3\_add\_ext\_t& r3\_add\_ext =

ies->criticalExtensions.choice.c1.choice.rrcConnectionRelease\_r3\_add\_ext;

// Set the R6 extensions.

r3\_add\_ext.nonCriticalExtension.present = ASN::RRCConnectionRelease\_r3\_IEs\_\_criticalExtensions\_\_c1\_\_rrcConnectionRelease\_r3\_add\_ext\_\_nonCriticalExtension\_PR\_v690NonCriticalExtensions;

ASN::V690NonCriticalExtensions\_t& v690NonCriticalExtensions = r3\_add\_ext.nonCriticalExtension.choice.v690NonCriticalExtensions;

v690NonCriticalExtensions.interRATInfo\_r6.present = ASN::InterRATInfo\_r6\_PR\_interRATInfoCDMA2000\_r6;

ASN::InterRATInfoCDMA2000\_r6\_t& interRATInfoCDMA2000\_r6 = v690NonCriticalExtensions.interRATInfo\_r6.choice.interRATInfoCDMA2000\_r6;

// Set the RedirectionInfo field.

interRATInfoCDMA2000\_r6.redirectInfo\_r6.present = ASN::RedirectInfo\_r6\_PR\_gsmToUtran;

ASN::RedirectInfo\_gsmToUtran\_t& redirectInfoGsmToUtran = interRATInfoCDMA2000\_r6.redirectInfo\_r6.choice.gsmToUtran;

// Set the GSM-TargetCellInfoList.

redirectInfoGsmToUtran.gsm\_TargetCellInfoList\_r6.present = ASN::GSM\_TargetCellInfoList\_r6\_PR\_setup;

// Set the FrequencyInfo.

ASN::FrequencyInfo\_t& frequencyInfo = redirectInfoGsmToUtran.frequencyInfo\_r6;

frequencyInfo.arfcn = ASN::ARFCN\_ValueEUTRA(uep->chanNum);

frequencyInfo.bandwidth.present = ASN::BandwidthEUTRA\_PR\_bw6;

frequencyInfo.bandwidth.choice.bw6 = ASN::BandwidthEUTRA::BandwidthEUTRA\_bw6;

//

OTHES :

void sendRrcConnectionRelease(UEInfo \*uep) {

// Create the RB Setup Message.

ASN::DL\_DCCH\_Message\_t msg;

memset(&msg,0,sizeof(msg));

msg.message.present = ASN::DL\_DCCH\_MessageType\_PR\_rrcConnectionRelease;

msg.message.choice.rrcConnectionRelease.present = ASN::RRCConnectionRelease\_PR\_r3;

ASN::RRCConnectionRelease\_r3\_IEs\_t \*ies =

&msg.message.choice.rrcConnectionRelease.choice.r3.rrcConnectionRelease\_r3;

// RRC\_TransactionIdentifier\_t

unsigned transactionId = uep->newTransactionId();

ies->rrc\_TransactionIdentifier = transactionId;

// N\_308\_t

ies->n\_308 = RN\_CALLOC(ASN::N\_308\_t);

\*ies->n\_308 = 1;

// ReleaseCause\_t

ies->releaseCause = toAsnEnumerated(ASN::ReleaseCause\_normalEvent);

// v690NonCriticalExtensions

ies->m.v690NonCriticalExtensionsPresent = true;

ASN::RRCConnectionRelease\_r3\_IEs\_t\_v690NonCriticalExtensions &v690ext = ies->v690NonCriticalExtensions;

// RRCConnectionRelease-v690ext-IEs

v690ext.m.r3\_IEsPresent = true;

ASN::RRCConnectionRelease\_r3\_IEs\_t\_v690NonCriticalExtensions\_r3\_IEs &r3\_ext = v690ext.r3\_IEs;

// RedirectionInfo-r6

r3\_ext.m.redirectionInfo\_r6Present = true;

ASN::RedirectionInfo\_r6 &redir\_info = r3\_ext.redirectionInfo\_r6;

redir\_info.m.redirectionTypePresent = true;

redir\_info.redirectionType = ASN::RedirectionType\_geranToUtran;

redir\_info.targetCellInfoList.m.gSM\_TargetCellInfoListPresent = true;

ASN::GSM\_TargetCellInfoList &gsm\_target\_list = redir\_info.targetCellInfoList.gSM\_TargetCellInfoList;

// Add GSM redirection target info

ASN::GSM\_TargetCellInfoList\_Elem\_t gsm\_target;

gsm\_target.m.cellIdentityPresent = true;

gsm\_target.cellIdentity.mcc = 123;

gsm\_target.cellIdentity.mnc = 456;

gsm\_target.cellIdentity.lac = 789;

gsm\_target.cellIdentity.ci = 12345;

gsm\_target.routingAreaCode = 123;

gsm\_target.frequencyInfo.m.fddPresent = true;

gsm\_target.frequencyInfo.fdd.downlinkFrequency = 975;

gsm\_target.frequencyInfo.fdd.uplinkFrequencyOffset = ASN::UARFCN\_Value\_t::UplinkOffset\_t::UARFCN\_Value\_t\_\_uplinkOffset\_f1;

gsm\_target.m.hcs\_CellSelectInfoPresent = true;

gsm\_target.hcs\_CellSelectInfo.temporaryOffset = 1;

gsm\_target.hcs\_CellSelectInfo.penaltyTime = ASN::PenaltyTime\_t::penaltyTime\_v5;

gsm\_target.hcs\_CellSelectInfo.hcs\_RSCP\_Evaluation = 3;

gsm\_target.hcs\_CellSelectInfo.q\_HCS = 4;

gsm\_target.m.q\_RxLevMinPresent = true;

gsm\_target.q\_RxLev

OTHERS :

void sendRrcConnectionRelease(UEInfo \*uep) {

// Create the RB Setup Message.

ASN::DL\_DCCH\_Message\_t msg;

memset(&msg,0,sizeof(msg));

msg.message.present = ASN::DL\_DCCH\_MessageType\_PR\_rrcConnectionRelease;

msg.message.choice.rrcConnectionRelease.present = ASN::RRCConnectionRelease\_PR\_r3;

ASN::RRCConnectionRelease\_r3\_IEs\_t \*ies =

&msg.message.choice.rrcConnectionRelease.choice.r3.rrcConnectionRelease\_r3;

// RRC\_TransactionIdentifier\_t

unsigned transactionId = uep->newTransactionId();

ies->rrc\_TransactionIdentifier = transactionId;

// N\_308\_t

ies->n\_308 = RN\_CALLOC(ASN::N\_308\_t);

\*ies->n\_308 = 1;

// ReleaseCause\_t

ies->releaseCause = toAsnEnumerated(ASN::ReleaseCause\_normalEvent);

// v690NonCriticalExtensions

ies->m.v690NonCriticalExtensionsPresent = true;

ASN::RRCConnectionRelease\_r3\_IEs\_t\_v690NonCriticalExtensions &v690ext = ies->v690NonCriticalExtensions;

// RRCConnectionRelease-v690ext-IEs

v690ext.m.r3\_IEsPresent = true;

ASN::RRCConnectionRelease\_r3\_IEs\_t\_v690NonCriticalExtensions\_r3\_IEs &r3\_ext = v690ext.r3\_IEs;

// RedirectionInfo-r6

r3\_ext.m.redirectionInfo\_r6Present = true;

ASN::RedirectionInfo\_r6 &redir\_info = r3\_ext.redirectionInfo\_r6;

// Set the redirection type to GERAN to UTRAN

redir\_info.m.redirectionTypePresent = true;

redir\_info.redirectionType = ASN::RedirectionType\_geranToUtran;

// Set the target cell info list for GSM redirection

redir\_info.targetCellInfoList.m.gSM\_TargetCellInfoListPresent = true;

ASN::GSM\_TargetCellInfoList &gsm\_target\_list = redir\_info.targetCellInfoList.gSM\_TargetCellInfoList;

// Add GSM redirection target info

ASN::GSM\_TargetCellInfoList\_Elem\_t gsm\_target;

gsm\_target.m.cellIdentityPresent = true;

gsm\_target.cellIdentity.mcc = 123;

gsm\_target.cellIdentity.mnc = 456;

gsm\_target.cellIdentity.lac = 789;

gsm\_target.cellIdentity.ci = 12345;

gsm\_target.routingAreaCode = 123;

gsm\_target.frequencyInfo.m.fddPresent = true;

gsm\_target.frequencyInfo.fdd.downlinkFrequency = 975;

gsm\_target.frequencyInfo.fdd.uplinkFrequencyOffset = ASN::UARFCN\_Value\_t::UplinkOffset\_t::UARFCN\_Value\_t\_\_uplinkOffset\_f1;

gsm\_target.m.hcs\_CellSelectInfoPresent = true;

gsm\_target.hcs\_CellSelectInfo.temporaryOffset = 1;

gsm\_target.hcs\_CellSelectInfo.penaltyTime = ASN::PenaltyTime\_t::penaltyTime\_v5;

gsm\_target.hcs\_CellSelectInfo.hcs\_RSCP\_Evaluation = 3;

g

OTHERS :

void sendRrcConnectionRelease(UEInfo \*uep) {

// Create the RB Setup Message.

ASN::DL\_DCCH\_Message\_t msg;

memset(&msg,0,sizeof(msg));

msg.message.present = ASN::DL\_DCCH\_MessageType\_PR\_rrcConnectionRelease;

msg.message.choice.rrcConnectionRelease.present = ASN::RRCConnectionRelease\_PR\_r3;

ASN::RRCConnectionRelease\_r3\_IEs\_t \*ies =

&msg.message.choice.rrcConnectionRelease.choice.r3.rrcConnectionRelease\_r3;

// RRC\_TransactionIdentifier\_t

unsigned transactionId = uep->newTransactionId();

ies->rrc\_TransactionIdentifier = transactionId;

// N\_308\_t

ies->n\_308 = RN\_CALLOC(ASN::N\_308\_t);

\*ies->n\_308 = 1;

// ReleaseCause\_t

ies->releaseCause = toAsnEnumerated(ASN::ReleaseCause\_normalEvent);

// v690NonCriticalExtensions

ies->m.v690NonCriticalExtensionsPresent = true;

ASN::RRCConnectionRelease\_r3\_IEs\_t\_v690NonCriticalExtensions &v690ext = ies->v690NonCriticalExtensions;

// RRCConnectionRelease-v690ext-IEs

v690ext.m.r3\_IEsPresent = true;

ASN::RRCConnectionRelease\_r3\_IEs\_t\_v690NonCriticalExtensions\_r3\_IEs &r3\_ext = v690ext.r3\_IEs;

// RedirectionInfo-r6

r3\_ext.m.redirectionInfo\_r6Present = true;

ASN::RedirectionInfo\_r6 &redir\_info = r3\_ext.redirectionInfo\_r6;

redir\_info.m.redirectionTypePresent = true;

redir\_info.redirectionType = ASN::RedirectionType\_geranToUtran;

redir\_info.targetCellInfoList.m.gSM\_TargetCellInfoListPresent = true;

ASN::GSM\_TargetCellInfoList &gsm\_target\_list = redir\_info.targetCellInfoList.gSM\_TargetCellInfoList;

// Add GSM redirection target info

ASN::GSM\_TargetCellInfoList\_Elem\_t gsm\_target;

gsm\_target.m.cellIdentityPresent = true;

gsm\_target.cellIdentity.mcc = 123;

gsm\_target.cellIdentity.mnc = 456;

gsm\_target.cellIdentity.lac = 789;

gsm\_target.cellIdentity.ci = 12345;

gsm\_target.routingAreaCode = 123;

gsm\_target.frequencyInfo.m.fddPresent = true;

gsm\_target.frequencyInfo.fdd.downlinkFrequency = 975;

gsm\_target.frequencyInfo.fdd.uplinkFrequencyOffset = 0;

gsm\_target.m.hcs\_CellSelectInfoPresent = true;

gsm\_target.hcs\_CellSelectInfo.temporaryOffset = 1;

gsm\_target.hcs\_CellSelectInfo.penaltyTime = ASN::PenaltyTime\_t::penaltyTime\_v5;

gsm\_target.hcs\_CellSelectInfo.hcs\_RSCP\_Evaluation = 3;

gsm\_target.hcs\_CellSelectInfo.q\_HCS = 4;

gsm\_target.m

gsm\_target.m.q\_RxLevMinPresent = true;

gsm\_target.q\_RxLevMin = -105;

gsm\_target.m.q\_RxLevMinOffsetPresent = true;

gsm\_target.q\_RxLevMinOffset = -12;

gsm\_target.m.cellEnvironmentPresent = true;

gsm\_target.cellEnvironment = ASN::CellEnvironment\_urban;

gsm\_target.m.reportingCellStatusPresent = true;

gsm\_target.reportingCellStatus = ASN::ReportingCellStatus\_notMeasured;

// Set the number of targets to 1

gsm\_target\_list.n = 1;

gsm\_target\_list.elem = &gsm\_target;

// Serialize the message.

std::vector<uint8\_t> buffer = ASN::serialize(msg);

// Send the message.

uep->sendMessage(buffer.data(), buffer.size());

// Remove UE context.

uep->release();

}

5G SIB :

[5G NR System Information Block Type 8 -SIB8 - RRC Signalling (techplayon.com)](https://www.techplayon.com/5g-nr-system-information-block-type-8-sib8/)

[5G NR System Information Type#1 - SIB 1 - RRC Signalling - Techplayon](https://www.techplayon.com/5g-nr-system-information-type1-sib-1/)