CELL SELECTION IMPROVEMENTS
LOW LEVEL DESIGN

(Edit via File/Properties/Summary!)
Rev 004

applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products

Applications

Amplifiers
amplifier.ti.com
Audio
www.ti.com/audio
Data Converters
dataconverter.ti.com
Automotive
www.ti.com/automotive
DSP
dsp.ti.com
Broadband
www.ti.com/broadband
Interface
interface.ti.com
Digital Control
www.ti.com/digitalcontrol
Logic
logic.ti.com
Military
www.ti.com/military
Power Mgmt
power.ti.com

Optical Networking
www.ti.com/opticalnetwork
Microcontrollers
microcontroller.ti.com
Security
www.ti.com/security

Telephony
www.ti.com/telephony

Video & Imaging
www.ti.com/video

Wireless
www.ti.com/wireless

Mailing Address:
Texas Instruments
Post Office Box 655303
Dallas, Texas 75265

Revision History
Date
Revision
Reason for Change
Changed By
Approval
Authority
Dec 06, 2004
000
Initial version of Document
K.Gopinath

001
Review by Sasken

DEC 21, 2004
002
after review of TIB

January 11,
2005
003_TIB
After reply of Sasken and review of TIB

Feb 01, 2005
003
Final version for Phase 1 of Cell Selection improvements feature

Apr 04, 2005
004
Final version of Phase 2 of cell selection improvements feature

NOTE: In the table shown above, the name of the person who made the changes as well as the approval authority is displayed in hidden text and does not print as part of the final document. Click on the paragraph symbol icon in the toolbar to show or hide this information.

Table of Contents

1 Purpose

This document describes the Low Level design for Cell Selection improvements feature according to the requirements specified in the corresponding High Level Design document cell_selection_005.doc. The High Level Design document for Cell Selection Improvements feature is attached to the Conquest issue RR_ENH_26751.

2 Scope

This document describes all the code changes required to implement Cell Selection Improvements feature. Changes to Data Structures, Global Variables, Macros, constants and Functions are identified and described. Pseudo-code is used wherever possible.

The document is divided into the following sections

Data Structures : Describes all new data structures introduced
Design Description : Describes the functional changes
Interface changes : Describes changes to MPH sap between RR AND ALR
Configurable Parameters : Describes all Dynamic Configuration Commands introduced for this feature
Deviations : Describes any deviations taken from HLD.

2.1 Terms and Definitions

Abbreviation/Term

Expansion/Definition
API
Application Programming Interface
ARFCN
Absolute Radio Frequency Channel Number
CQ
Conquest
DCS
Digital Communication System
FFS
Flash File System
PCM
Permanent Configuration Memory
PCS
Personal Communication System
RxLev
Signal Level of a carrier

2.2 References

[1] Cell_selection_005.doc - High Level Design document for Cell Selection Improvements feature

3 Data Structures

This section describes all the new data structures introduced/modified to implement Cell Selection Improvements feature.

3.1 Stuck in "Limited Service"

A fix has been proposed under CQ 24416.

3.2 CDMA Carriers

3.2.1 Black List

The "Black List" is intended to contain carriers that cannot be synchronized (like CDMA carriers), and will be maintained dynamically. The "Black List" can contain the following carriers

1. CDMA carriers
2. Non-BCCH GSM carriers

3.2.1.1 Chosen Data Structure

Carriers belonging to either European or American bands can be part of "Black List" at any given time in a given place. As a result two separate lists are used to represent the "Black List". The data structure used to represent the "Black List" is given below

```
 {
  T_LIST list [2];                /* Separate lists for Euro &
American region */
  U8 sfc [2] [512];             /* Separate sync fail counters for
Euro & American region  */
} T_CS_BLACK_LIST
```

Structure Members

Type
Size
Description
list
T_LIST

2 * T_LIST
Each bit represents one carrier in the range 0 - 1024 (1024/8 = 128).
Bit value
1   - Carrier is part of "Black List"
0   - Carrier is not part of "Black List"
sfc
U8
2 * 512 bytes
4 bits are used to represent synchronization failure counter for each carrier. This   accommodates two carriers in one byte and limits the max value of sync fail counter to 15.

3.2.1.2 Other Data Structures considered for Black List

The following two data structures were also considered for Black List before finally selecting the one described in section 3.2.1.1.

3.2.1.2.1 Fixed Length Array

```
{
   U16 arfcn;
   U8   sfc;
   U8   black_or_grey;
} T_CS_BLACK_LIST

{
 T_CS_BLACK_LIST black_list [ 200 ];
} T_CS_DATA
```

Disadvantages of this approach

* Restricts the number of Black List Carriers
* 4 bytes are required for each carrier
* Operations like addition and removal of carriers from Black List requires search through the entire list making this approach very inefficient in terms of run time. Using this data structure will defeat any improvement in cell selection time that we hope to achieve by this feature.
* Checking the presence of a carrier in Black List also requires search through the entire list.
* Totally new functions have to be implemented to support the above operations. This means more testing effort.
* The existing functions implemented in "rr_srv.c" file for manipulating GSM carriers represented in the form of T_LIST structure cannot be used. Totally new functions have to be implemented to support the above operations. This means more testing effort. The image size will also increase considerably.

3.2.1.2.2 Single Bitmap

```
{
 T_LIST arfcn;
 U8 sfc[512];
} T_CS_BLACK_LIST

{
 T_CS_BLACK_LIST  black_list;
} T_CS_DATA
```

Disadvantages of this approach

* All the GSM channels (Both European and American) have to be accommodated in the same T_LIST array.
* Since DCS 1800 and PCS 1900 bands use the same channel numbers, this would require re-ordering of channel numbers in RR for Black List that is different from standard GSM format. This would require another round of conversion of channel numbers between Black List format and standard GSM format.
* The existing functions implemented in "rr_srv.c" file for manipulating GSM carriers represented in the form of T_LIST structure cannot be used.
* Totally new functions have to be implemented to manipulate the new representation of carriers. This increases the image size as well as the testing effort.

3.2.1.3 Rationale for the chosen structure

* Separate list for European and American regions. This makes the implementation flexible and simple.
* Operations such as addition, removal and checking the presence of carriers in "Black List" take constant time. Run time efficient.
* All the existing functions implemented in "rr_srv.c" file for manipulating GSM carriers represented in the form of T_LIST structure can be used. This minimizes the testing effort as well as the image size.

3.2.2 White List

The "White List" contains carriers that are good candidates for "Full Service". Carriers belonging to only one region (European or American) will be part of "White List" at any given time in a given place. The following data structure will be used to represent "White List" and related information.

During the CR or CS process the MS reads the SI2, SI2bis and SI2ter for each synchronized carrier and stores this temporarily only in the arrays si2 to si2ter. But not each 'CR' cell becomes the serving cell afterwards (It can be a not suitable cell; wrong PLMN or access class, barred or insufficient path loss).

The SI2x information, collected during the CR/CS process, shall be copied from the temporary storage (arrays si2 to si2ter) only in case of successful cell (re-) selection.

For temporary use during BCCH reading process:

```
{
  U8   si2 [BA_BITMAP_SIZE];
  U8   si2bis [BA_BITMAP_SIZE];
  U8   si2ter [BA_BITMAP_SIZE];
  } T_CR_WHITE_LIST
```

Structure Members

Type

Size

Description

si2

U8

BA_BITMAP_SIZE
Stores the BA list received in System Information Type 2 message
si2bis
U8
BA_BITMAP_SIZE
Stores the BA list received in System Information Type 2BIS message.
si2ter
U8
BA_BITMAP_SIZE
Stores the BA list received in System Information Type 2TER message

After successful cell selection:

```
{
  U8 region;
  U16 last_sc_arfcn;
  T_Loc_area_ident   last_sc_lac;
  T_LIST   list;
  } T_CS_WHITE_LIST
```

Structure Members

Type
Size
Description
region
U8
1
Indicates the region, "White List" belongs to. Takes two values
0 - European region
1 - American region
last_sc_arfcn
U16
1
Used to store the last serving cell ARFCN in Full Service
last_sc_lac
T_Loc_area_ident
1
Used to store the Location area code of the last serving cell in Full Service. This will be used during initial cell selection to check whether MS finds "Full Service" in the same location area, where it was switched off.
list
T_LIST
1
Contains all carriers which are set in the si2, si2bis and si2ter
3.2.3 New Search Modes
Four new search modes are introduced to improve the Cell Selection Process.
1. FAST SEARCH

This Search mode is started after a Downlink Failure on a carrier with "Full Service" or a BCCH Read Failure on a carrier with "Full Service". The objective of "FAST SEARCH" is to camp on cell as fast as possible. Black List shall be used during FAST SEARCH.

2. NORMAL SEARCH

NORMAL SEARCH is carried out normally when a Cell Selection is required, for example, after expiry of TFAST_CS timer, MS has stayed in dedicated mode for more than 30 seconds, etc. Black List shall be used during NORMAL SEARCH

3. FULL SEARCH

Full search is used during initial cell selection, for example after power ON. It is also carried out after expiry of TNORMAL_CS timer. Only one attempt of FULL SEARCH is done and NORMAL SEARCH is restarted. Black list shall not be used during FULL SEARCH

4. BLACK LIST SEARCH

BLACK LIST SEARCH is used to identify all inactive carriers in current region and remove all such carriers from "Black List". BLACK LIST SEARCH is carried out after cell reselection to a different Location or Routing area.

No new data structures are required to implement the search modes. Realization of new search modes requires changes to MPH SAP interface between RR and ALR. Two new timers TFAST_CS and TNORMAL_CS are also introduced in RR to manage the search modes.

3.3 Multiple Frequency Bands in a Region

None

3.4 Region Selection Problem

None

3.5 BA list and Last Used Serving Cell storage and usage

See Implementation in Section 3.2.2 "White List".

3.6 Multiple requests from MM during Cell selection

None

3.7 Frequent searching of carriers during 2 Scans

None

3.8 Configurable Parameters

A new Data Structure shall be introduced in RR to hold all Dynamic Configuration commands related variables. This has the advantage of better identification and can be better enclosed by !defined(NCONFIG).

```
{
  U32 tfast_cs_val;
  U32  tnormal_cs_val;
  U8  upper_rxlev_thr;
  U8  medium_rxlev_thr;
  U8  lower_rxlev_thr;
  U8 bl_cs_en;
  U8 fcr ;
  U8  scr ;
  U8 fca ;
  U8  fho ;
  U8 iho ;
  U8 set_band ;
  U8 no_sys_time ;
  U8 dcs_offset ;
  U8 gsm_offset ;
  U8 nkc ;
 T_TIME lim_ser_nps_delay ;
} T_DYNAMIC_CONFIG
```

Structure Members

Type
Description
Dynamic Command
bl_cs_en
U8
Controls Black List Search
0 - Black List Search is disabled
1 - Black List Search is enabled
Default Value : 1
BL_CS
tfast_cs_val
U32
Value of TFAST_CS_timer in seconds
Default Value :
 TFAST_CS_VALUE (4min)
TFAST_CS
tnormal_cs_val
U32
Value of TNORMAL_CS timer in seconds
Default Value : TNORMAL_CS_VALUE(4min)
TNORMAL_CS
upper_rxlev_thr
U8
Upper RxLev threshold for GSM channels.
Default Value : UPPER_RXLEV_THRESHOLD
U_RXT
medium_rxlev_thr
U8
Medium RxLev threshold for GSM channels
Default Value : MEDIUM_RXLEV_THRESHOLD
M_RXT
Lower_rxlev_thr
U8
Lower RcLev threshold for GSM channels
Default Value
LOWER_RXLEV_THRESHOLD
L_RXT
lim_ser_nps_delay
T_TIME
Time delay between two consecutive Non-Prallel searches in Limited service. This delay is used to provide enough time for Emergency Calls
SET_NPS_DELAY

4 Design Description

This section describes all the Global variables, Macros and Functions introduced or modified to implement the Cell Selection

Improvements feature. Pseudo-code is used to describe the functionality wherever possible.

4.1 CDMA Carriers

4.1.1 Black List

This section describes all the Global variables, Macros and Functions introduced or modified to implement "Black List".

4.1.1.1 Global Variables

The following global variables will be introduced in T_CS_DATA structure in RR to manage "Black List".

```
{
   U8 initial_plmn_search;
   T_CS_BLACK_LIST black_list;
   U8 region;
} T_CS_DATA
```

Structure Members
Variable
Type
Size
Description
Initial_plmn_search
U8
1
Identifies the first "FUNC_PLMN_SEARCH" request from MM after power on. Takes three values as described below
INITIAL_PLMN_SEARCH_NOT_ACTIVE
First "FUNC_PLMN_SEARCH" not received yet

INITIAL_PLMN_SEARCH_ACTIVE
First "FUNC_PLMN_SEARCH" received

INITIAL_PLMN_SEARCH_DONE
First "FUNC_PLMN_SEARCH" already proc
essed

black_list
T_CS_BLACK_LIST
1
Stores the "Black List" information
region
U8
1
Stores the current region. This is derived from global "STD" variable and shall be updated whenever global "STD" changes. This is passed later as a parameter to "Black List" management functions.
Ex : cs_remove_BA_MA_from_black_list
0 - European region
1 - American region

It is very important to test the usage of "Black List" across power cycles in windows simulation environ-ment also. This can be achieved by the following sequence of primitives

RR_DEACTIVATE_REQ

This is sent whenever MS is switched off. All the RR data structures are initialized again in the function handling RR_DEACIVATE_REQ primitive. The "Black List" information is written to FFS and read back again from FFS.

RR_ACTIVATE_REQ

Activate the MS again. The "Black List" stored and read back during last RR_DEACTIVATE_REQ will be used from now on.

Since FFS is not available during Windows simulation testing, RAM shall be used to simulate the same. The following global variable shall be defined to simulate FFS for "Black List"

```
#if defined(_SIMULATION_)
T_LIST  win_black_list[2];
#endif
```

4.1.1.2 Macros

The following macros will be introduced in rr.h file to manage "Black List"

4.1.1.2.1 CS_GET_REGION_FROM_FREQ

Prototype:

```
 CS_GET_REGION_FROM_FREQ ( arfcn )
```

Description:

Returns the region (European or American) the requested carrier belongs to. This macro shall be used only from places where the ARFCN field contains Region and STD information

Input:

Absolute Radio Frequency Channel Number of a carrier containing "Band" and "Region" information

Definition:

```
#define CS_GET_REGION_FROM_FREQ

((arfcn&US_BIT)?AMERICAN_REGION:EUROPEAN_REGION)
```

4.1.1.2.2 CS_SET_BLACK_LIST_FLAG

Prototype:

```
CS_SET_BLACK_LIST_FLAG ( index )
```

Description:

This macro sets Bit : 2 of attributes [ ] field n T_CS_DATA structure for the carrier identified by the index. This bit indicates whether a carrier is a candidate for "Black List" or not. The BLACK_LIST_FLAG in the attribute filed shall be set during initial PLMN search(as indicated by the flag initial_plmn_search_active) whenever MS fails to synchronize to a carrier. After the completion of initial PLMN search, Black list flag shall be used to update the Black List database based on the outcome of initial PLMN search.

Input:

Index to the attributes [ ] field of channel in T_CS_DATA structure.

Definition:

```
#define CS_SET_BLACK_LIST_FLAG ( index )
                                                     ( rr_data-
>cs_data.attributes[index] | =CS_BLACK_LIST_FLAG )
```

4.1.1.2.3 CS_GET_BLACK_LIST_FLAG

Prototype:

CS_GET_BLACK_LIST_FLAG (index)

Description:

Returns the value of Bit : 2 of attributes [ ] field n T_CS_DATA structure for the carrier identified by the index. This bit indicates whether a carrier is a candidate for "Black List" or not. This macro will be called while updating Black list database following the completion of initial PLMN search.

Input:

Index to the attributes [ ] field of channel in T_CS_DATA structure.

Definition:

```
#define CS_GET_BLACK_LIST_FLAG ( index )
                                                     ( rr_data-
>cs_data.attributes[index] & CS_BLACK_LIST_FLAG )
```

4.1.1.3 Constants

The following constants will also be introduced in rr.h file to manage "Black List"

Macro

Value

Description

MAX_SYNC_FAILURES

5

Maximum number of sync failures after which a Reasonably strong carrier will be moved to "Black List".

CS_BLACK_LIST_FLAG

0x04

Mask bit for "Black List" flag in attributes[ ] field of T_CS_DATA

This bit will be used to identify carriers to which MS failed to synchronize during initial PLMN search. This information will be used later to update Black list database following the completion of initial PLMN search

MAX_SFC_PER_REGION

512

Array size of sync fail counter per region

4.1.1.4 enums

The following enums will also be introduced in rr.h file to manage "Black List"

enum

members

Value

Description

clear_black_list_e

CLR_BLACK_LIST_RAM
0
Clear "Black List" from RAM
Black list database shall be cleared from RAM

CLR_BLACK_LIST_FFS
1
Clear "Black List" from FFS
Black List database shall be cleared from FFS
initial_plm_search_e
INITIAL_PLMN_SEARCH_NOT_ACTIVE
0
Indicates that the initial PLMN search request from MM for Full service has not yet been received from FFS
Black List database shall be cleared from FFS

INITIAL_PLMN_SEARCH_ACTIVE
1
Indicates that initial PLMN search request from MM for Full service has been received and the PLMN search is currently active

INITIAL_PLMN_SEARCH_DONE
2
Indicates that the initial PLMN search request from MM for Full service has been processed

4.1.1.5 Function API for Managing "Black List"
The following functions will be implemented in RR to manage "Black List".
The following functions modify the entire "Black List".
Function API
Description
Reference
cs_clear_black_list
Clears the "Black List" from FFS
Clears the "Black List" from RAM

4.1.1.5.1
cs_store_black_list

Stores the "Black List" f to FFS during switch off.
Calls Function rr_csf_write_black_list ( ) to write the black list to FFS.
4.1.1.5.2
rr_csf_write_black_list
Writes the "Black List" to Flash File System
In windows simulation environment, "Black List" will be written to simulated FFS.
Synchronization failure counter information shall not be written to FFS.
4.1.1.5.3
rr_csf_read_black_list
Reads the "Black List" from Flash File System

In windows simulation environment, "Black List" will be read from simulated FFS.

4.1.1.5.4

The following functions are used to control the addition/deletion of carriers from "Black List".

Function API

Description

Reference

cs_add_to_black_list

Adds a carrier to "Black List"

4.1.1.5.5

cs_del_from_black_list

Removes a carrier from "Black List"

4.1.1.5.6

cs_remove_BA_MA_from_black_list

Updates the "Black List" by removing cells from BA/MA lists

4.1.1.5.7

cs_update_black_list

Updates the "Black List" following the outcome of first "FUNC_PLMN_SEARCH" after power on. Adds some cells according to the current attributes array values of the last selection process.

4.1.1.5.8

cs_inc_sync_fail_counter

Increments the synchronization failures counter of the requested carrier by one.

4.1.1.5.9

cs_reset_sync_fail_counter

Resets the synchronization failures counter of the requested carrier

4.1.1.5.10

cs_get_sync_fail_counter

Returns the synchronization failures counter of the requested carrier

4.1.1.5.11

cs_is_in_black_list

Checks whether a carrier is already part of "Black List" or not

4.1.1.5.12

cs_check_black_list_criteria

Evaluates the "Black List" criteria for a carrier

4.1.1.5.13

4.1.1.5.1 cs_clear_black_list

Prototype:

void cs_clear_black_list (U8 which)

Description:

This function is used to clear "Black List" database. The function clears the "Black List" database from RAM or FFS or both RAM and FFS. This function is called in the following cases

1) In response to "ERASE_BL" dynamic configuration command

2) After Initial PLMN search based on its outcome

Input Parameters:

which : RAM or FFS or both RAM and FFS

Returns:

None

Pseudo-code :

```
void cs_clear_black_list(U8 which)
{
       Reset the black list database including the sync fail counter in RAM/ FFS
}
```

4.1.1.5.2 cs_store_black_list

Prototype:

void cs_store_black_list (   )

Description:

This is a wrapper function for storing "Black List" information to FFS . This in turn calls rr_csf_write_black_list( ) to store "Black List" to FFS. This function is called during power off.

Input Parameters:

None

Return Value:

None

Pseudo-code :

```
void cs_store_black_list (   )
{
    Store the "Black List" from RAM to Flash File System during switch off
}
```

4.1.1.5.3 rr_csf_write_black_list

Prototype:

void rr_csf_write_black_list (T_LIST *black_list)

Description:

This function writes "Black List" information to FFS. In case of windows simulation environment, "Black List" is stored to simulated FFS area. This function is called during switch off.

Input Parameters:

lack_list   : Pointer to "Black List" information

Return Value:

None

Pseudo-code :

```
void rr_csf_write_black_list(T_LIST *black_list))
{ if ( Windows SIMULATION )
   Write the "Black List to simulated FFS for "Black List".
else
 {
    Check if the directory is created/ create one if not
    Write the Black List to Flash File System
    Handle the error
}
}
```

4.1.1.5.4 rr_csf_read_black_list

Prototype:

void rr_csf_read_black_list (   )

Description:

This function copies "Black List" information from FFS to RR internal "Black List" data structures. In case of windows simulation environment, "Black List" is read from simulated FFS area. This function is called after power on.

Input Parameters:

None

Return Value:

None

Pseudo-code :

```
void rr_csf_read_black_list (   )
{
  if ( Windows SIMULATION )
  Read  the "Black List from simulated FFS for "Black List".
else
   {    Check if the directory is created/ create one if not
        Read the Black List from Flash File System
       Handle the error
  }
}
```

4.1.1.5.5 cs_add_to_black_list

Prototype:

void cs_add_to_black_list (U8 region, U16 arfcn, U8 rxlev)

Description:

This function is used to add GSM channels to "Black List". The function checks the "Black List" criteria before adding it to the list. This function is called whenever MS fails to synchronize to a GSM channel.

Input Parameters:

region	:	European or American region of the carrier
arfcn	:	Absolute Radio Frequency Channel Number of a GSM carrier
rxlev	:	signal level of the channel

Returns:

None

Pseudo-code :

```
void cs_add_to_black_list(U8 region, U16 arfcn, U8 rxlev )
{
    Validate region
   Validate ARFCN
   if ( criteria for adding arfcn to "Black List" is satisfied  )
    {
       add  arfcn to Black list of the corresponding region
    }
}
```

4.1.1.5.6 cs_del_from_black_list

Prototype:

void cs_del_from_black_list (U8 region, U16 arfcn)

Description:

This function is used to delete a GSM channel from "Black List". The function deletes the channel from the "Black List" and also resets its SFC counter to zero. This function is called whenever MS successfully synchronizes to a GSM channel.

Input Parameters:

Region : European or American region of the carrier
arfcn : Absolute Radio Frequency Channel Number of a GSM carrier

Returns:

None

Pseudo-code :

```
void cs_del_from_black_list(U8 region,  U16 arfcn  )
{
   remove the carrier from the black list
   reset the sync fail counter for this carrier
 }
```

4.1.1.5.7 cs_remove_BA_MA_from_black_list

Prototype:

void cs_remove_BA_MA_from_black_list (U8 region, T_LIST *source_list)

Description:

This function is used to remove the GSM channels present in MA and BA lists from the "Black List" because such channels are valid carriers for this local environment. The function deletes these channels from the "Black List" and also resets their SFC counter to zero. This function is called whenever MS receives BA and MA list information in any RR message.

Input Parameters:

region : Indicates European / American region
0 - European region
1 - American region
source_list : Input BA / MA list

Returns:

None

Pseudo-code :

```
void cs_remove_BA_MA_from_black_list(U8 region,  T_LIST
*source_list )
{
        remove the argument list from the black list corresponding
to the region
}
```

4.1.1.5.8 cs_update_black_list

Prototype:

void cs_update_black_list (   )

Description:

This function is used to update "Black List" database after initial PLMN search. It first clears the current "Black list" database from RAM and then adds some cells to "Black List" according to the current attributes array values(BLACK_LIST_FLAG) of the Initial PLMN selection process. This function is called under the following cases

1. MS enters Limited or No service after Initial PLMN search
2. MS enters Full service in a different Location area from where it is switched off after initial PLMN search

Input Parameters:

None

Return Value:

None

Pseudo-code :

```
void cs_update_black_list (   )
{

    if( RR enters Full Service following first FUNC_PLMN_SEARCH
and the Location area  is not the same
        as the one before switch off ) OR
      ( RR enters Limited or No service following first
FUNC_PLMN_SEARCH )     {
        clear the "Black List" in RAM read from FFS after Power on

        Update the "Black List" based on the current search
information. Use Black List flag information in
        attributes [  ] field to identify "Black List" candidates.
    }
 }
```

4.1.1.5.9 cs_inc_sync_fail_counter

Prototype:

void cs_inc_sync_fail_counter (U8 region, U16 arfcn)

Description:

This function increments the SFC counter for "Reasonably strong" GSM carriers. The size of SFC counter is 4 bits. As a result two carriers are accommodated in one byte. This function first converts the ARFCN range from 1-1023 to 0-511 format and increments the SFC accordingly. The SFC format is shown below

Index
MSB 4 bits
LSB 4 bits
0
ARFCN : 2
ARFCN : 1
1
ARFCN : 4
ARFCN   3
.

.

510
ARFCN : 1022
ARFCN: 1021
511
ARFCN:0
ARFCN:1023
ARFCN: 0 = CHANNEL_0_INTERNAL

Input Parameters:

region : European or American region

arfcn : Absolute Radio Frequency Channel Number of a GSM carrier

Returns:
None

Pseudo-code :

```
void cs_inc_sync_fail_counter( U8 region, U16 arfcn )
{
   Validate the channel ARFCN 0-1023, 1024 (CHANNEL_0_INTERNAL)
  Convert ARFCN range from 1-1023 to 0-511
  Increment the 4 bit SFC counter for this ARFCN to a maximal value of 15
}
```

4.1.1.5.10 cs_reset_sync_fail_counter

Prototype:

void cs_reset_sync_fail_counter (U8 region, U16 arfcn)

Description:

This function resets the SFC counter of GSM carriers to zero. The size of SFC counter is 4 bits. As a result two carriers are accommodated in one byte. This function first converts the ARFCN range from 1-1023 to 0-511 format and resets the SFC accordingly. The SFC format is shown below

Index	MSB 4 bits	LSB 4 bits
0	ARFCN : 2	ARFCN : 1
1	ARFCN : 4	ARFCN 3
.		
.		
510	ARFCN : 1022	ARFCN: 1021
511	ARFCN:0	ARFCN: 1023

ARFCN: 0 = CHANNEL_0_INTERNAL

Input Parameters:

Region   : European or American region
arfcn   : Absolute Radio Frequency Channel Number of a GSM carrier

Returns:
None

Pseudo-code :

```
void cs_reset_sync_fail_counter( U8 region, U16 arfcn )
{
   Validate the channel ARFCN 0-1023, 1024 (CHANNEL_0_INTERNAL)
```

```
  Convert ARFCN range from 1-1023 to 0-511
 Reset  the 4 bit SFC counter for this arfcn
}
```

4.1.1.5.11 cs_get_sync_fail_counter

Prototype:

U8 cs_get_sync_fail_counter (U8 region, U16 arfcn)

Description:

This function returns the SFC counter of GSM carriers. The size of SFC counter is 4 bits. As a result two carriers are accommodated in one byte. This function first converts the ARFCN range from 1-1023 to 0-511 format and returns the SFC accordingly. The SFC format is shown below

Index	MSB 4 bits	LSB 4 bits
0	ARFCN : 2	ARFCN : 1
1	ARFCN : 4	ARFCN 3
.		
.		
510	ARFCN : 1022	ARFCN: 1021
511	ARFCN:0	ARFCN:1023

ARFCN: 0 = CHANNEL_0_INTERNAL

Input Parameters:

Region   : European or American region
arfcn   : Absolute Radio Frequency Channel Number of a channel along with "Band" and "Region" information

Returns:

Sync failure counter

Pseudo-code :

```
U8 cs_get_sync_fail_counter( U8 region, U16 arfcn )
{
   Validate the channel ARFCN 0-1023, 1024 (CHANNEL_0_INTERNAL)
  Convert ARFCN range from 1-1023 to 0-511
 Return  the 4 bit counter value for this arfcn
}
```

4.1.1.5.12 cs_is_in_black_list

Prototype:

```
BOOL  cs_in_black_list (U8 region, U16 arfcn )
```

Description:

This function is used to check whether a GSM channel is already black listed or not.. This check is necessary in order to avoid setting the same flag again.

Input Parameters:

Region   : European or American region of the carrier
arfcn   : Absolute Radio Frequency Channel Number of a GSM carrier

Return Value:

TRUE   - If the channel is present in the "Black List"
 FALSE - If the channel is not present in the "Black List"

Pseudo-code :

```
BOOL  cs_is_in_black_list(U8 region, U16 arfcn )
{
   Check its presence in the corresponding black list
   Return the presence status
}
```

4.1.1.5.13 cs_check_black_list_criteria

Prototype:

```
BOOL  cs_check_black_list_criteria (U8 region, U16 arfcn, U8 rxlev
)
```

Description:

This function checks the criteria for adding a GSM channel to "Black List". GSM channels are added to "Black List" only after they satisfy this criteria.   This function is called from cs_add_to_black_list( ) function.

Input Parameters:

region   : European or American region of the carrier
arfcn   : Absolute Radio Frequency Channel Number of a channel along with "Band" and "Region" information
rxlev   : signal level of the channel

Return Value:

 TRUE   - If the channel satisfies "Black List" criteria
 FALSE - If the channel doesn't satisfy "Black List" criteria

Pseudo-code :

```
BOOL  cs_check_black_list_criteria (U8 region, U16 arfcn, U8 rxlev
)
{
   if(channel is present in White List)
       return FALSE;
   if( channel signal level is above upper_level_threshold )
      return TRUE;
   if( channel signal level is between upper_level_threshold and
medium_level_threshold )
   {
      Increment synchronization failure count for this carrier
      if( synchronization failure count is equal to or more than
max sync failures)
        return TRUE;
      else
        return FALSE;
```

```
    }
    return FALSE;
}
```

4.1.1.6 Addition of carriers to Black List

* cs_add_black_list (U16 arfcn, U8 rxlev) function shall be called to add a channel to "Black List".
* cs_add_black_list (U16 arfcn, U8 rxlev) shall be called from function cs_mph_bsic_cnf(   ) following failure to decode Frequency / synchronization bursts for this channel. The function cs_check_black_list_crieria() decides if the carrier satisfies the criterion for entering the Black List.

4.1.1.7 Storing of Black List carriers on the FFS

* cs_store_black_list (   ) function shall be called to store "Black List" on FFS.
* cs_store_black_list (   ) function shall be called from function att_rr_deactivate_req( ) during switch off.
* cs_store_black_list (   ) function always stores "Black List" on FFS irrespective of RR service.

4.1.1.8 Reading of Black List carriers from FFS

* rr_csf_read_black_list_from_ffs (   ) function shall be used to copy "Black List" from FFS to RAM after power on.
* rr_csf_read_black_list_from_ffs (   ) function shall be called from function cs_init_process( )) after power on..

4.1.1.9 Erasing the Black List

* "Black List" read from FFS after power on shall be used provided the MS finds "Full Service" in the same Location Area where it was switched off. The "Black List" read from FFS shall be erased in all other cases. This functionality is implemented in the function cs_clear_black_list ( ).

4.1.1.10 Updating the Black List

* cs_update_black_list (   ) shall be called after first "FUNC_PLMN_SEARCH".
* Global variable initial_plmn_search is used to identify the first "FUNC_PLMN_SEARCH" .
* initial_plmn_search is initialized to INITIAL_PLMN_SEARCH_NOT_ACTIVEin function pei_init( ) after power on. It is set to INITIAL_PLMN_SEARCH_ACTIVE in function att_handle_rr_activate_req( ) when processing the first RR_ACTIVATE_REQ primitive from MM with "FUNC_PLMN_SERCH". The variable is set to INITIAL_PLMN_SEARCH_DONE after the completion of first "FUNC_PLMN_SEARCH" and will remain so till another power cycle.
* initial_plmn_search shall be set to INITIAL_PLMN_SEARCH_NOT_ACTIVE in rr_deactivate_req( ) primitive.

4.1.1.11 Removal of individual carriers from the Black List

While in NORMAL or FAST SEARCH, it is impossible to camp on a carrier on the Black List, no matter how its accessibility has changed since the carrier entered the Black List. Therefore, it is important to be able to modify the Black List when network conditions change.

* All carriers found in BA list are removed from "Black List". Function cs_remove_BA_MA_from_black_list( ) shall be used for this purpose. BA list is received in System Information Messages

2, 2bis, 2ter, 5, 5bis and 5ter. Hence function cs_remove_MA_BA_from_black_list( ) is called from the following functions to update the "Black List" with BA list.

att_copy_sys_info_2_par( ), att_copy_sys_info_2bis_par( ), att_copy_sys_info_2ter_par( )
att_copy_sys_info_5_par( ), att_copy_sys_info_5bis_par( ), att_copy_sys_info_5ter_par( )

* All carriers found in MA list are removed from "Black List". Function cs_remove_MA_BA_from_black_list( ) shall be used for this purpose. MA list is received in Assignment Command, Channel Mode Modify, Frequency Redefinition, Handover Command, Immediate Assignment , Immediate Assignment Extended and system information type 4 messages. Function cs_remove_MA_BA_from_black_list( ) shall be called from all the functions that process these messages.

* All inactive carriers as reported as inactive carriers in MPH_POWER_CNF primitive are also removed from Black List.

* Any carrier that is successfully synchronized during the Synchronization Phase shall be removed from "Black List". Function cs_del_black_list( ) shall be used for this purpose. cs_del_black_list( ) function shall be called from function cs_mph_bsic_cnf( ) following synchronization success on a carrier.

4.1.2 White List

This section describes all the Global variables, Macros and Functions introduced or modified to implement "White List". Currently, BCCH information uses PCM API for storage. FFS API shall replace this. The "White List" shall now be written directly to FFS.

4.1.2.1 Global variables

The following global variables will be introduced in T_CS_DATA structure in RR to manage "White List".

```
{
      T_CS_WHITE_LIST white_list;     // structure defined in section 3.2.2
} T_CS_DATA
```

Structure Members

Variable	Type	Size	Description
white_list	T_CS_WHITE_LIST	1	Stores the "White List" information

It is very important to test the usage of "white List" across power cycles. In windows simulation environ-ment this can be achieved by the following sequence of primitives

RR_DEACTIVATE_REQ

This is sent whenever MS is switched off. All the RR data structures are initialized again in the function handling RR_DEACIVATE_REQ primitive. The "White List" information is written to FFS and read back again from FFS.

RR_ACTIVATE_REQ

Activate the MS again. The "White List" stored and read back during last RR_DEACTIVATE_REQ will be used from now on.
Since FFS is not available during Windows simulation testing, RAM shall be used to simulate the same. The following global variable shall be defined to simulate FFS for "White List" .

```
#if defined(_SIMULATION)
T_CS_WHITE_LIST win_white_list;
#endif
```

The following variables shall be removed from T_RR_DATA and T_CS_DATA structures as they are now accommodated in T_CS_WHITE_LIST/T_CR_WHITE_LIST structures.

last_used_sc_arfcn

white_list_si2

white_list_si2bis

white_list_si2ter

4.1.2.2 Function API for managing "White List"

The following functions will be implemented/modified in RR to manage "White List".

Function API

Description

Reference

cs_set_bcch_info

This function already exists, but will be replaced by a function that copies the SIM BCCH info to T_CS_WHITE_LIST structure.

4.1.2.2.1

dat_convert_white_list

This function is already existing, but will be modified to store region and serving cell info as well

4.1.2.2.2

cs_store_white_list

Stores White List to FFS. This function replaces the function cs_store_bcch_info()

4.1.2.2.3

rr_csf_write_white_list

Writes the "White List" to FFS

4.1.2.2.4

rr_csf_read_white_list

Reads the "White List" from FFS to RAM

4.1.2.2.5

cs_clear_white_list

This replaces the function cs_clear_bcch_info( ).

4.1.2.2.6

cs_is_in_white_list()

Returns if a channel is present in white list

4.1.2.2.7

cs_use_white_list_info

Increases the priority of white list carriers in MPH_POWER_CNF array to high priority. This replaces the existing function cs_use_bcch_information( )

4.1.2.2.8

dat_store_neigh_cell_desc

Modified to update white List following any change in system information on serving cell

4.1.2.2.9

The following functions will be removed from RR, following the changes to "White List" information storage/usage in cell selection improvements feature.

1. void cs_use_bcch_information(void) - replaced by cs_use_white_list_info( )
2. void cs_use_last_used_sc (void) - removed
3. cs_collect_stored_bcch_info( ) -- removed
4. void cs_clear_bcch_info( ) - replaced by cs_clear_white_list ( )
5. void cs_store_bcch_info( ) - replaced by cs_store_white_list( )

4.1.2.2.1 cs_set_bcch_info

Prototype:

void cs_set_bcch_info (T_bcch_info * sim_bcch_info)

Description:

This function converts the SIM BCCH information to T_LIST format and merges it with the White List database.

Input Parameters:

sim_bcch_info - SI2 BA List information stored in SIM

Return Value:

None

Pseudo-code :

```
void cs_set_bcch_info ( T_bcch_info * sim_bcch_info )
{
   Merge the SIM BCCH info with the "White List" database
}
```

4.1.2.2.2 dat_convert_white_list

Prototype:

void dat_convert_white_list ( )

Description:

This function converts the BCCH information to T_LIST format and stores it in the White List database. This function is called whenever Full service is reached following Cell Selection or reselection

Input Parameters:

None

Return Value:

None

Pseudo-code :

```
void dat_convert_white_list ( )
{
   if ( RR is in Full Service )
  {
      Reset the current White List info
      Save the current region in " White List" database
      Save the current serving cell arfcn in "White List" database
      Save the current the Location Area in "White List" database
      Convert the BA list received in SI2, SI2bis and SI2ter
messages into "White List" database
```

```
   }
}
```

4.1.2.2.3 cs_store_white_list

Prototype:

```
void cs_store_white_list (   )
```

Description:

This function is called during power off only when the mobile in Full service. It stores the white list information to FFS.

Input Parameters:

None

Return Value:

None

Pseudo-code :

```
void cs_store_white_list (   )
 {
     Store the "White List" from RAM to Flash File System
}
```

4.1.2.2.4 rr_csf_write_white_list

Prototype:

```
void rr_csf_write_white_list ( T_CS_WHITE_LIST *white_list  )
```

Description:

This function writes "White List" information to FFS. In case of windows simulation environment, "White List" is stored to simulated FFS area. This function is called during switch off.

Input Parameters:

white list   : pointer to T_CS_WHITE_LIST structure

Return Value:

None

Pseudo-code :

```
void rr_csf_write_white_list ( T_CS_WHITE_LIST *white_list  )
{
 If ( Windows simulation)
  Write "White List" information to simulated FFS for "White
list".
else
{
    Check if the directory is created/ create one if not
    Write the White List to Flash File System
    Handle the error
}
}
```

4.1.2.2.5 rr_csf_read_white_list

Prototype:

```
void rr_csf_read_white_list (   )
```

Description:

This function copies "White List" information from FFS to RR internal "White List" data structures. In case of windows simulation environment, "White List" is read from simulated FFS area. This function is called after power on.

Input Parameters:

None

Return Value:

None

Pseudo-code :

```
void rr_csf_read_white_list (   )
{
   If ( Windows simulation)
     Read "White List" information from simulated FFS for "White list".
  else
   {
      Check if the directory is created/ create one if not
      Read the White List from Flash File System
      Handle the error
    }
}
```

4.1.2.2.6 cs_clear_white_list

Prototype:

void cs_clear_white_list (U8 which)

Description:

This function is used to clear "White List" database. The function clears the "White List" database from RAM or FFS or SIM . This function is called in the following cases

1) In response to "ERASE_WL" dynamic configuration command

Input Parameters:

Which : RAM, SIM or FFS

Return Value:

None

Pseudo-code :

```
Void cs_clear_white_list (U8 which  )
{
   Clear the "White List" database from RAM/SIM/FFS
}
```

4.1.2.2.7 cs_is_in_white_list

Prototype:

BOOL cs_is_in_white_list (U8 region, U16 arfcn)

Description:

This function is used to check whether a GSM channel is in White List or not.. A white listed carrier is never added to Black List

Input Parameters:

Region	: European or American region of the carrier
arfcn	: Absolute Radio Frequency Channel Number of a GSM carrier

Return Value:

TRUE   - If the channel is present in the "White List"
 FALSE - If the channel is not present in the "White List"

Pseudo-code :

```
BOOL  cs_is_in_white_list(U8 region, U16 arfcn )
{
   Check its presence in the white list
   Return the presence status
}
```

4.1.2.2.8 cs_use_white_list_info

Prototype:

void cs_use_white_list_info (U8 num_of_chan)

Description:
This function is used to increase the priority of White List carriers present in MPH_POWER_CNF primitive to CS_HIGH_PRIORITY.
Input Parameters:
num_of_chan   : Number of white list carriers
Return Value:
void
Pseudo-code :

```
void  cs_use_white_list_info(U8 num_of_chan )
{
   Increase the priority of "White List" carriers to High priority
}
```

4.1.2.2.9 dat_store_neigh_cell_desc

Prototype:

```
void dat_store_neigh_cell_desc (UBYTE si, UBYTE index,

BUF_neigh_cell_desc *cd,
                                                      T_LIST
*new_neigh_list)
```

Description:
This function is used to store the neighbor cell information
Input Parameters:
si   : si2/si2bis or si2ter
index   : cell index (CR_INDEX or SC_INDEX)
cd   : neighbour cell information
new_neigh_list   : neighbor cell information in T_LIST format
Return Value:
void
Pseudo-code :
Only the modificationa are described here

```
void  cs_use_white_list_info(U8 num_of_chan )
{
   In case of change in system information on Serving cell, copy
the new neighbor cell
    information to White List database.
}
```

4.1.2.3 Storing of White List information on the FFS

* The "White List" is stored to FFS during switch off, if the MS is in "Full Service" state.
* cs_store_white_list( ) function shall be used for this purpose. Function cs_store_white_list( ) shall be called from function att_rr_deactivate_req( ) during switch off.

4.1.2.4 Reading of White List information from FFS

* After power ON the "White List" is read from the FFS and used as a White List.
* rr_csf_read_white_list (   ) function shall be used for this purpose.
* rr_csf_read_white_list (   ) function shall be called from function cs_init_process( ) after power on.

4.1.2.5 Usage of White List information during Cell Selection

* White List will be used in Non-Parallel cell selection. The BCCH information stored in "White List" database shall be passed to MPH_POWER_REQ primitive.

4.1.3 New search Modes

This section describes all the Global variables, Macros and Functions introduced or modified to implement the "New Search Modes".

4.1.3.1 Global Variables

New global variables are introduced both in RR and ALR entities as described below.

4.1.3.1.1 RR entity

```
{
  U8 previous_search_mode;
  U8 current_search_mode;
}T_CS_DATA
```

Structure Members
Variable
Type
Size
Description
previous_search_mode
U8
1
Identifies the previous search mode used
Current_search_mode
U8
1
Indentifies the current search mode

4.1.3.1.2 ALR entity

```
{
   T_MPH_POWER_REQ *p_power_req;
} T_CS_DATA
```

Structure Members
Variable
Type
Size
Description
p_power_req
T_MPH_POWER_REQ *
1
Stores the pointer to MPH_POWER_REQ primitive received from RR
power_scan_attempts
U8
4
This is a constant array indexed by the search mode. It contains the number of scan attempts for each search mode.
tim_powermeas_value
U16
4
This is a constant array indexed by the search mode. It contains the value for POWER_MEAS timer for each search mode.

4.1.3.2 Constants

4.1.3.2.1 RR Entity

The following constants are introduced in RR to handle New Search Modes.

Constant	Value	Description
TFAST_CS_VALUE	240000 ms	Default value for TFAST_CS timer
TNORMAL_CS_VALUE	240000 ms	Default value for TNORMAL_CS timer
UPPER_RXLEV_THRESHOLD	20	Identifies the upper threshold for the signal level of a channel. All carriers stronger than this threshold will be directly added to "Black List" following synchronization failure
MEDIUM_RXLEV_THRESHOLD	10	Identifies the medium threshold for the signal level of a channel. All carriers stronger than this threshold but weaker than upper threshold are considered as "Reasonably strong" carriers.

4.1.3.2.2 ALR entity

The following constants are introduced in ALR to handle New Search Modes.

Constant	Value	Description
FAST_SEARCH_MODE_ATTEMPTS	1	Number of search mode attempts for Fast search mode
BLACK_LIST_SEARCH_MODE_ATTEMPTS	1	Number of search mode attempts for Black List search mode
TIM_FAST_SEARCH_POWERMEAS_VAL	800ms	Value of POWERMEAS timer for Fast search mode
TIM_BLACK_LIST_SEARCH_POWERMEAS_VAL	800ms	Value of POWERMEAS timer for Black List search mode

4.1.3.3 Timers

Two new timers shall be introduced in RR to manage the New Search Modes.

Timer	Value	Description	Expiry Handler
T_FAST_CS	tfast_cs_val Default : 4min	Controls Fast Search	tim_tfast_cs The function traces the

white and black lists

T_NORMAL_CS
tnormal_cs_val
Default : 4min
Controls Normal Search

tim_tnormal_cs
The function traces white and Black lists

4.1.3.4 Function changes for New Search modes

New functions are introduced and some existing functions are modified both in RR and ALR entities to implement "New Search Modes" functionality.

4.1.3.4.1 RR Entity

The following functions are added/modified in RR to support "New Search Modes" functionality.

Function API
Description
Reference
cs_get_new_search_mode
Returns the new search mode
4.1.3.4.1.1
cs_handle_search_mode_timer
Handles the timers for the new Search Modes
4.1.3.4.1.2
att_start_cell_selection
This function is already existing, but will be modified to accommodate New Search Modes
4.1.3.4.1.3
cs_start_scan
This is an existing function. This will be modified to accommodate New Search Modes functionality
4.1.3.4.1.4
tim_treg
This is an existing function. This will be modified to accommodate New Search Modes functionality
4.1.3.4.1.5
att_start_cell_selection_gprs
This function is already existing, but will be modified to accommodate New Search Modes
4.1.3.4.1.6
att_full_service_found
New function. This function is called whenever FULL SERVICE is reached.
4.1.3.4.1.7
att_check_dynamic_search_mode_config
Updates the new search mode based on the current dynamic configuration of search modes
4.1.3.4.1.8
tim_tfast_cs
Expiry handler function for T_FAST_CS timer

4.1.3.4.1.9
tim_tnormal_cs
Expiry handler function for T_NORMAL_CS timer
4.1.3.4.1.10
cs_mph_power_cnf
This function is already existing, but will be modified to accommodate New Search Modes
4.1.3.4.1.11
4.1.3.4.1.1 cs_get_new_search_mode

Prototype:
U8 cs_get_new_seach_mode (   )

Description:
This function is used to obtain the new search mode based on the current search mode and the current state of search mode timers

Input Parameters:
None

Return Value:

```
Search Mode     :   FAST_SEARCH_MODE
                             NORMAL_SEARCH_MODE
                            FULL_SEARCH_MODE
                            BLACK_LIST_SEARCH
```

Pseudo-code :

```
U8 cs_get_new_search_mode (   )
{
    if ( Timer TFAST_CS is active )
       search type  =   FAST_SEARCH;

    else if ( Timer TNORMAL_CS is active )
     search type  =  NORMAL_SEARCH;

    else if ( previous search type is FULL_SEARCH or FAST_SEARCH )
     search type  =  NORMAL_SEARCH;

   else if ( previous search type is NORMAL_SEARCH )
     search type =   FULL_SEARCH;

    return search_type;
}
```

4.1.3.4.1.2 cs_handle_search_mode_timer

Prototype:
void cs_handle_search_mode_timer (U8 search_mode)

Description:
This function handles the search mode timers based on the new search mode. This function is called from cs_start_scan() function before sending MPH_POWER_REQ primitive to ALR

Input Parameters:

```
search Mode     :   FAST_SEARCH_MODE
                             NORMAL_SEARCH_MODE
                            FULL_SEARCH_MODE
                           BLACK_LIST_SEARCH
```

Return Value:
None

Pseudo-code :

```
void cs_handle_search_mode_timer   U8 search_mode )
```

```
{
      if  ( search mode is Fast Search  )
      {
         Start TFAST_CS timer if not running already.
      }
      if  ( Search mode is Normal Search )
     {
         Start TNORMAL_CS timer if not running already
     }
     if  ( Search mode is Full Search  )
     {
         if ( cell selection is originated by MM and the requested service is not equal to Net Search )
          {
            Stop TFAST_CS timer
            Stop  TNORMAL_CS timer
         }
     }
}
```

4.1.3.4.1.3 att_start_cell_selection

Prototype:

void att_start_cell_selection (BOOL originator, BOOL parallel, U8 search_mode)

Input Parameters:

```
originator            :   MM originated
                                RR originated
parallel               :    Parallel search
                               Non-Parallel search
search Mode     :   FAST_SEARCH_MODE
                              NORMAL_SEARCH_MODE
                             FULL_SEARCH_MODE
                             BLACK_LIST_SEARCH_MODE
```

Return Value:

None

Pseudo-code :

Only the changes with reference to New Search Modes are described here

```
void att_start_cell_selection (BOOL originator, BOOL parallel, U8 search_mode)
{
   Update search mode based on the dynamic search mode config
}
```

4.1.3.4.1.4 cs_start_scan

Prototype:

void cs_start_scan (   )

Input Parameters:

None

Return Value:

None

Pseudo-code :

Only the changes with reference to New Search Modes are described here

```
void cs_start_scan (BOOL originator, BOOL parallel, U8 search_mode)
```

```
{
    Set the new search mode in MPH_POWER_REQ primitive
    Copy Black List information to MPH_POWER_REQ primitive
   if (search_mode EQ BLACK_LIST_SEARCH)
      Copy "grey" carriers to the black list of the MPH_POWER_REQ
primitive
  else
      Copy White List information to MPH_POWER_REQ primitive
   Handle the Search Mode timers
}
```

4.1.3.4.1.5 tim_treg

Prototype:

```
void tim_treg (   )
```

Input Parameters:

None

Return Value:

None

Pseudo-code :

Only the changes with reference to New Search Modes are described here

```
void tim_treg (  )
{
   Obtain the new search mode
   Pass the new search mode to att_start_cell_selection
}
```

4.1.3.4.1.6 att_start_cell_selection_gprs

Prototype:

```
void att_start_cell_selection (BOOL originator,  U8 search_mode)
```

Input Parameters:

```
originator            :   MM originated
                                RR originated
search Mode     :   FAST_SEARCH_MODE
                              NORMAL_SEARCH_MODE
                             FULL_SEARCH_MODE
                             BLACK_LIST_SEARCH_MODE
```

Return Value:

None

Pseudo-code :

Only the changes with reference to New Search Modes are described here

```
void att_start_cell_selection (BOOL originator,  U8 search_mode)
{
   Update current search mode
 }
```

4.1.3.4.1.7 att_full_service_found

Prototype:

```
U8 att_full_service_found ( )
```

Description:

This function is called whenever RR reaches Full service following cell selection or reselection. All tasks that need to be performed after RR reaches full service are done here

Input Parameters:

None

Return Value:

None

Pseudo-code :

```
U8 att_full_service_found ( ))
{
  if (RR service is full service )
  {
    Call dat_copy_white_list( ) function to update white list.
   Send SI2 information to SIM through MM
   Stop TFAST_CS and TNORMAL_CS timers
  }
}
```

4.1.3.4.1.8 att_check_dynamic_search_mode_config

Prototype:

```
U8 att_check_dynamic_search_mode_config ( )
```

Description:

This function checks the current dynamic configuration of search modes and updates the new search mode accordingly. This function is called from att_start_cell_selection() before sending MPH_POWER_REQ primitive to ALR

Input Parameters:

None

Return Value:

Search_mode - new search mode to be used

Pseudo-code :

```
U8 att_check_dynamic_search_mode_config ( ))
{
   if( current search mode is Fast search and Fast search is
disabled)
      new search mode = Normal search;

  if( current search is normal search and Normal search is
disabled)
    new search mode = Full search

   return new search mode
 }
```

4.1.3.4.1.9 tim_tfast_cs

Prototype:

```
void tim_tfast_cs (   )
```

Description:

This is an expiry routine for T_FAST_CS timer. This function is currently used to trace all active Black List and White List carriers

Input Parameters:

None

Return Value:

None

Pseudo-code :

```
void tim_tfast_cs (  )
{
   Trace the white and Black list carriers
}
```

4.1.3.4.1.10 tim_tnormal_cs

Prototype:

```
void tim_tnormal_cs (   )
```

Description:

This is an expiry routine for T_NORMAL_CS timer. This function is currently used to trace all active Black List and White List carriers

Input Parameters:

None

Return Value:

None

Pseudo-code :

```
void tim_tnormal_cs (  )
{
   Trace the white and Black list carriers
}
```

4.1.3.4.1.11 cs_mph_power_cnf

Prototype:

```
void cs_mph_power_cnf (   )
```

Input Parameters:

None

Return Value:

None

Pseudo-code :

Only the modifications are described here

```
void cs_mph_power_cnf (  )
{
   In case of Fast search mode, copy only "White List" and
"Reasonably strong" carriers from
  MPH_POWER_CNF primitive.
}
```

4.1.3.4.2 ALR Entity

The following functions will be added/modified in ALR to support New Search Modes functionality

Function API	Description	Reference
cs_is_in_black_list	Check if the carrier is in Black List or not	4.1.3.4.2.1
ma_mph_power_req	Function is already existing, but will be modified to store the pointer to MPH_POWER_REQ primitive and doesn't free it anymore.	4.1.3.4.2.2
cs_power_req	This function is already existing, but will be modified to store region and serving cell info as well	4.1.3.4.2.3
cs_prepare_power_req	This function is already exists, but will be modified to fill the power array with black and grey carriers only in case of Black List Search.	

4.1.3.4.2.4
cs_add_and_sort_channels

This function already exists, but will be modified to copy SIM BCCH info to T_CS_WHITE_LIST structure.

4.1.3.4.2.5
cs_rxlev_ind

This function already exists, but will be modified to incorporate New search modes

4.1.3.4.2.6
cs_add_white_list_carriers

This is a new function. This function will add all carriers that are present in the white list and whose rxlev is greater than LOWER_RXLEV_THRESHOLD at the top of the MPH_POWER_CNF array in the descending order of their strength

4.1.3.4.2.7
4.1.3.4.2.1 cs_is_in_black_list

Prototype:

```
U8 cs_is_in_black_list ( U8 region, U16 arfcn )
```

Input Parameters:

```
region         :  European or American region of the carrier
arfcn           : Absolute Radio Frequency Channel Number of a
channel along with "Band" and "Region"
                      information
```

Return Value:

```
FALSE   - Not present in Black List
TRUE   - Present in Black List
```

Pseudo-code :

```
U8 cs_is_in_black_list(U8 region, U16 arfcn )
{
     if (  search Mode is Full Search )
         return  FALSE;
     Check its presence in the corresponding black list
    Return the presence status
}
```

4.1.3.4.2.2 ma_mph_power_req

Prototype:

```
U8 ma_mph_power_req ( T_MPH_POWER_REQ * mph_power_req )
```

Input Parameters:

```
mph_power_req :  Pointer to T_MPH_POWER_REQ primitive
```

Return Value:

None

Pseudo-code :

Only the changes with reference to New Search Modes are described here

```
U8 ma_mph_power_req( T_MPH_POWER_REQ * mph_power_req  )
{
   Store the pointer to MPH_POWER_REQ primitive in the global
variable p_power_req
   Do not FREE the MPH_POWER_REQ buffer (the primitive will be
freed short before the CNF will be
    sent)
}
```

4.1.3.4.2.3 cs_power_req

Prototype:

U8 cs_power_req (U8 pch_interrupt)

Input Parameters:

pch_interrupt :   with or without PCH interruption

Return Value:

None

Pseudo-code :

Only the changes with reference to New Search Modes are described here

```
U8 cs_power_req (U8 pch_interrupt)
{
   Set the number of RF scan attempts and TIM_POWER_MEAS timer value based on the search mode
}
```

4.1.3.4.2.4 cs_prepare_power_req

Prototype:

T_POWER_MEAS* cs_prepare_power_req (void)

Input Parameters:

None

Return Value:

Pointer to a MPHC_RXLEV_REQ structure

Pseudo-code :

Only the changes with reference to New Search Modes are described here

```
T_POWER_MEAS* cs_prepare_power_req (void)
{
   if (search_mode EQ BLACK_LIST_SEARCH)
      Fills the power_array with grey and black list carriers only depend on the actual region
      (derived from the black list of MPH_POWER_REQ)
   else
      Fills the power_array with all possible carriers depend on the actual region
}
```

4.1.3.4.2.5 cs_add_and_sort_channels

Prototype:

U8 cs_add_and_sort_channels ( )

Input Parameters:

None

Return Value:

None

Pseudo-code :

Only the changes with reference to New Search Modes are described here

```
U8 cs_add_and_sort_channels (   )
{

    if( EXT measurements are not running)
   {
        Fill all inactive carriers in the "inactive_carrier_list" of MPH_POWER_CNF array
        If(search mode is Black List search )
             Return;
        Fill all carriers from the White List whose Rxlev is more
than the LOWER_RXLEV_THRESHOLD in
        the   MPH_POWER_CNF array first. Sort them based on their
strength.
    }

      If ( search mode is not Full list search )
        Do not include Black list carriers in the MPH_POWER_CNF
primitive

Fill all remaining carriers. Only those carriers are added whose
Rxlev is greater than LOWER_RXLEV_THRESHOLD. Carriers are added in
descending order of field strengths, irrespective of which
frequency bands (and region) it belongs to.
A minimum of 40 carriers are added for each supported frequency
band provided they are available
If space is still available in MPH_POWER_CNF list more carriers
will be added. The maximum limit of carriers per individual band
that can be added to MPH_POWER_CNF list is 60.
Fill all remaining carriers (which are not  Black  List carriers)
whose RxLev is more than LOWER_RXLEV_THRESHOLD in the
MPH_POWER_CNF array until the maximal number is reachedThe order
of this remaing carriers is following the rules:
- from strongest to weakest carrier
- strive for even distribution between low and high frequencies
(GSM <----> DCS/PCS)
- strive for even distribution of carrier from both regions if
available
Due to the partly contradictory of this rules MS has to fulfill
first the requirement/recommendation of the spec 3GPP TS 03.22,
section 3.2.1: "The number of channels to be searched are 15 for
GSM 450, 15 for GSM 480, 30 for GSM 850 Band, 30 for GSM 900 and
40 for DCS 1800 and PCS 1900.".
Always in cases when no more carriers of one band or region are
available the sorting algorithm should fill up the place according
to the rules mentioned before.
   }
```

4.1.3.4.2.6 cs_rxlev_ind

Prototype:

```
U8 cs_rxlev_ind (T_MPHC_RXLEV_IND *rxlev_ind))
```

Input Parameters:

rxlev_ind   - pointer to T_MPHC_RXLEV_IND structure

Return Value:

None

Pseudo-code :

Only the changes w.r.t New Search Modes are described here

```
U8 cs_rxlev_ind (T_MPHC_RXLEV_IND *rxlev_ind)
{
   if (maximal attempts reached)
  {
      Allocate the MPH_POWER_CNF primitive
Call cs_add_and_sort_channels( )
      Free MPH_POWER_REQ primitive
      Send MPH_POWER_CNF primitive to RR
   }
}
```

4.1.3.4.2.7 cs_add_white_list_carriers

Prototype:

```
U8 cs_add_white_list_carriers (U16 max, U8 std, U8 attemps, SHORT
min_rxlev,

T_POWER_MEAS* presults  )
```

Description:

This functions all White list carriers at the top MPH_POWER_CNF primitive array. The White List are also sorted based on their strength.

Input Parameters:

max - maximum number of carriers measured per region
std - Number of bands supported in this region
attempts - Number of RF measurements done
min_rxlev - Minimum rxlev of the carrier
presults - pointer to RxLev measurement results done by Layer 1

Return Value:

No of white list carriers added to the MPH_POWER_CNF array

Pseudo-code :

```
U8 cs_power_req (T_MPHC_RXLEV_IND *rxlev_ind)
{
    Add all carriers that are present in White List and whose
RxLev is greter than
    LOWER_RXLEV_THRESHOLD to the top of the MPH_POWER_CNF array.
    Sort the added carriers in the descending order of their
strength
}
```

4.1.3.5 Fast Search

a) FAST SEARCH is only used if requested service is "Full Service".

b) A new timer, TFAST_CS, will be started when FAST SEARCH is activated. This will be done by the function cs_handle_search_timers() in RR called from cs_start_scan( ).

c) The Black List, the White List and the search mode are passed as parameters in MPH_POWER_REQ to ALR. This is done in function cs_start_scan( ) in RR.

d) ALR will make only one Power Measurement across all supported Frequency bands. This is handled in function cs_power_req() in ALR.

e) The MPH_POWER_CNF returned by the ALR must not contain any carriers from the Black List. This is handled in function cs_add_and_sort_channels( ) in ALR.

f) All White List carriers shall be filled at the top of MPH_POWER_CNF array. This is handled in function cs_add_and_sort_channels( ) in ALR.

g) The rest of the MPH_POWER_CNF array will hold carriers (not from the Black List) whose Rxlev is more than LOWER_RXLEV_THRESHOLD and which fulfills the order rules. This is handled in function cs_add_and_sort_channels( ) in ALR.

h) First Scan is made on White List Carriers and "Reasonably Strong Carriers" for "Full Service". This is handled in function cs_sync_next_bsic( ) function in RR.
i) If no suitable carrier was found, the Second Scan is made on all with EMERGENCY_CELL marked carriers in MPH_POWER_CNF for "Limited Service". This is handled in function cs_sync_next_bsic ( ) which have to adapt to this new restriction.
j) CQ 27675 is not applicable here as we are searching "Reasonably strong" carriers also in addition to "White List" carriers. If the MS crosses borders, "White list" becomes useless, but the "reasonably strong carriers" would still contain carriers from the new roaming environment.
k) During the Second Scan, if a carrier is found where "Full Service" is possible, it is selected. However the scan should stop at the first available carrier where either "Limited Service" or "Full Service" is possible. This is already implemented as part of CQ 24416
l) FAST SEARCH timer is stopped after a carrier is found where "Full Service" is possible. This can be done in dat_convert_white_list( ) function.
m) If no suitable carrier is found where "Full Service" is available, FAST SEARCH will be used as long as the timer TFAST_CS is active. Handled by the function cs_get_new_search_mode ( ).
n) FAST SEARCH has also to be stopped in case of a new RR_ACTIVATE_REQ (both "limited" or "full plmn" or "net search").

4.1.3.6 Normal Search
a) A new timer, TNORMAL_CS, will be started when NORMAL SEARCH is started.. This will be done by the function cs_handle_search_timers( ) in RR called from cs_start_scan( ).
b) The Black List, the White List and the search mode are passed as parameters in MPH_POWER_REQ to ALR. This is done in function cs_start_scan( ) in RR.
c) 5 power measurements are made across each carrier in all supported bands spread over 3-5 seconds. This is handled in function cs_power_req() in ALR.
d) steps e) to g) of Fast Search
e) First Scan is made on White List Carriers and "Reasonably Strong Carriers" for "Full Service". This is handled in function cs_sync_next_bsic( ) function in RR.
f) NORMAL SEARCH is stopped after a carrier is found where "Full Service" is possible. This can be done in dat_copy_white_list( ) function.
g) If no suitable carrier is found where "Full Service" is available, NORMAL SEARCH will be used as long as the timer TNORMAL_CS is active. Handled by the function cs_get_new_search_mode( ).
h) After the expiry of TNORMAL_CS, the next Cell Selection would be a FULL SEARCH, if the MS has still not reached "Full Service". Handled by the function cs_get_new_search_mode( ).
4.1.3.7 Full Search
a) The Black List, the White List and the search mode are passed as parameters in MPH_POWER_REQ to ALR. This is done in function cs_start_scan( ) in RR.

b) 5 power measurements are made across each carrier in all supported bands spread over 3-5 seconds. This is handled in function cs_power_req() in ALR.

c) steps e) to j) of Normal Search

4.1.3.8 Black List Search

MS uses "Black List Search" to look for inactive Black list carriers after a cell reselection to a different Location Area or a Routing Area. In phase 1 Blacklist search will be initiated only when Location Area changes. Blacklist search following change in Routing area will be implemented in Phase 2, as this requires some more study.

4.1.3.8.1 Global Variables

The following global variables will be added in RR entity to support Black list search.

```
{
  U8 blacklist_search_pending;
  } T_CS_DATA;
```

Structure Members

Variable

Type

Size

Description

blacklist_search_pending

U8

1

Indicates that there was a change in location area and black list search is pending.

Black list search cannot be started immediately after change in Location area. It should be started after going back from dedicated state to idle state following the completion of Location area update procedure by MM. The establishment cause in received RR_ESTABLISH_REQ primitive can be used to check whether black list search should be started or not. However, we cannot rely completely on the establishment cause, as the establishment cause can be used for Detach also.

Hence this flag has been added. This flag will be set in function att_code_rr_act_ind( ) whenever a change in location area is detected.

4.1.3.8.2 Function API for Managing Black List Search

4.1.3.8.2.1 RR entity

The following functions will be implemented/modified in RR to manage Black List Search

Function API

Description

Reference

att_code_rr_act_ind

This function already exists. The Black List search pending flag will be set inside this function

4.1.4.2.1.2

4.1.3.8.2.1.1 att_code_rr_act_ind

Prototype:

void att_code_rr_act_ind ( )

Input Parameters:

None

Return Value:
None
Pseudo-code :

```
void att_code_rr_act_ind (   )
{
  Set the black list search pending flag whenever the location
area changes
}
```

4.1.3.8.2.2 ALR Entity

The following Functions are modified in ALR to implement Black List Search

Function
Description
Reference
cs_find_inactive_carriers
.Find all inactive carriers from the current search results and move them to "inactive carrier" list. This function is called by cs_add_and_sort_channels() for all search modes.
4.1.4.2.2.2

4.1.3.8.2.2.1 cs_find_inactive_carriers

Prototype:

```
void cs_find_inactive_carriers (T_POWER_MEAS **p_results, U16
*p_results_size
                                                     U8 *std,  U8
no_of_attempts, SHORT min_rxlev )
```

Description:
This functions detects all inactive carriers and adds them to MPH_POWER_CNF primitive. It also sets the RxLev of all Black List carriers to less than MIN RxLev , so that these carriers further in sorting

Input Parameters:
p_results   - pointer to a pointer pointing to T_POWER_MEAS structure
p_results_size   - pointer to size for European and American regions
std   - pointer to std value for European and American regions
no_of_attempts - number of attempts for the current search mode
min_rxlev   - minimum RxLev

Return Value:
None
Pseudo-code :

```
void cs_find_inactive_carriers (T_POWER_MEAS **p_results, U16
*p_results_size
                                                     U8 *std,  U8
no_of_attempts, SHORT min_rxlev)
{
    Add all the carriers who's RXLEV is less than
LOWER_RXLEV_THRESHOLD to "inactive carrier list"
    of MPH_POWER_CNF  primitive
    Set the Rxlev of all blacklisted carriers to MIN_RX_LEV -1
}
```

4.1.3.8.3 RR in "Full Service", after Location Area Update / Routing Area Update

* After a Location Area Update (or Routing Area Update), and RR reaches idle state (or Packet Idle), RR shall initiate a parallel BLACK LIST SEARCH to look for inactive carriers in the Black List.

4.1.3.8.4 RR in ""Limited Service"", requested service is ""Full Service"" and it has done a cell reselection to a carrier to another Location Area

* When RR is in ""Limited Service"" and a cell reselection has been done to a carrier to another Location Area (or Routing Area), then in theory, this is a good point to do a BLACK LIST SEARCH. However, the expiry of TREG timer could be used to look at Rxlev values of Black List carriers. After the cell reselection completes, the TREG timer shall be restarted with duration of one second (the reg_counter value that decides the duration of TREG timer is preserved). At its expiry RR would do a NORMAL SEARCH or FAST SEARCH or FULL SEARCH". This gives the MS a chance to look for inactive Black List carriers.

4.1.3.8.5 RR in ""Limited Service"", requested service is ""Limited Service"" and it has done a cell reselection to a carrier to another Location Area

* After the cell reselection completes, RR shall initiate a parallel BLACK LIST SEARCH.

4.1.3.8.6 MM sends RR_ESTABLISH_REQ when BLACK List Search is active

* Black List Search is stopped and the call establishment is carried on

* Functions att_dat_con_est( ) and att_notify_stop_plmn_search( ) will be modified to handle this requirement

4.1.3.9 Management of New Search Modes

The following table lists different scenarios and details which type of Cell Selection that should be used.

Sr No.

Scenario

Search Type

Search Mode

Function called

Called from

1

Power ON

Non Parallel

Full Search

att_start_cell_selection

att_handle_rr_act_req

2

After Dedicated Mode for more than 30 seconds, Cell Reselection started, and fails.

Non Parallel

Normal Search

att_start_cell_selection

att_select_cell_dedicated

3

After Dedicated Mode less than 30 seconds but not for a Location Area Update or Routing area update.

MS continues to camp on the same cell

None
None
None
4
After Dedicated Mode less than 30 seconds for a Location Area
Update or Routing area update.

For phase 1, only LU is being be used a trigger for Black List
search. However for the future, we need to combine LU/RU as a
trigger to start Black list search.

Parallel
Black List Search
att_start_cell_selection()
att_leave_dedicated

5
In "Full Service", cell reselecttion started and fails.

Cell Reselection was started for any of the following reasons
- "Downlink Failure"
- "BCCH Read Failure"
- C1 / C2 criterion
Non-Parallel
Fast Search is started if Full Service is requested by MM.
Otherwise Normal Search is used
att_start_cell_selection
att_try_old_cell
6
In Dedicated Mode and "Radio Link Failure" or "Data Link Failure".
A Cell Reselection is started and fails.
Non-Parallel
Fast Search is started if Full Service is requested by MM.
Otherwise Normal Search is used
att_start_cell_selection
att_select_cell_dedicated
7,8,9,10
In "Limited Service", cell reselection started and fails.

Cell Reselection was started for any of the following reasons
- "Downlink Failure"
- "BCCH Read Failure"
- C1 / C2 criterion
Non-Parallel
Function
cs_get_new_search_mode is called to obtain the new search mode
att_start_cell_selection
att_try_old_cell
11,
12,
13,
14
"Limited Service / No Service" and TREG timer expiry.
Parallel

Function
cs_get_new_search_mode is called to obtain the new search mode
att_start_cell_selection
tim_treg

15,
16,
17,
18,
18a
Net Search by MM
Non-Parallel or Parallel based on ATT state
Full Search
att_start_cell_selection
att_handle_rr_act_req
19
MM originated "Limited Service" search.
Non-Parallel
Full Search
att_start_cell_selection
att_handle_rr_act_req
20
Request from GRR, after a failure of Cell Change Order.

Non-Parallel
Normal Search
att_start_cell_selection
att_rrgrr_cr_req
21,
22
MM originated FUNC_PLMN_SEARCH.

Non-Parallel
Full Search
att_start_cell_selection
att_handle_rr_act_req
23
Cell reselection on GPRS activation fails due to TRESELECT timer expiry
Non-Parallel
Fast Search
att_start_cell_selection_gprs
tim_treselect

4.2 Multiple Frequency Bands in a Region

4.2.1 Increasing the size of carrier list in MPH_POWER_CNF

According to 3GPP TS 03.22, section 3.2.1 MS should scan a certain minimum number of carriers on each frequency band. The numbers of carriers to be searched are 30 for GSM 850 Band, 30 for GSM 900 and 40 for DCS 1800 and PCS 1900.

The size of the carrier list in MPH_POWER_CNF will be increased to 160. This is realized by changing the constant MAX_CHANNELS in MPH SAP file. This makes it possible to include a minimum of 40 carriers for each band mentioned above.

	Current Value	New Value
MAX_CHANNELS	80	160

4.2.2 Strategy in filling the carrier list

4.2.2.1 Rules

The following rules will be observed in filling the carrier list in MPH_POWER_CNF.

* Only those carriers are added whose Rxlev is greater than LOWER_RXLEV_THRESHOLD. Carriers are added in descending order of field strengths, irrespective of which frequency bands (and region) it belongs to.
* Carriers from the White List are added first. (The maximum number of carriers in a White List is 32.)
* Carriers listed as "Black" are not included except for "Full search mode".
* There should be a minimum of 40 carriers for each supported frequency band. If there are not enough carriers available in a particular band to fill 40 elements, then an exception to 40 carriers per band rule is made for that band.
* If space is still available in MPH_POWER_CNF list more carriers can be added. The maximum limit of carriers per individual band that can be added to MPH_POWER_CNF list is 60. This is done to keep the carrier list in MPH_POWER_CNF small.

4.2.2.2 Design Approach

The algorithm for filling carrier list in MPH_POWER_CNF shall be changed to improve its run time efficiency. Currently, addition of each carrier to the list, requires MAX_CARRIERS_DUAL_EGSM + MAX_CARRIERS_DUAL_US iterations through the power array list reported by Layer 1 for a quad band MS. The information collected while adding the first carrier to the list is not used in the subsequent additions. For example, all carriers whose RxLev is less than LOWER_RXLEV_THRESHOLD, can be excluded from power array list after the first addition. The new cell selection algorithm incorporates all such changes to improve the run time efficiency. The same is described below.

The following strategy shall be used in filling/sorting the carrier list in MPH_POWER_CNF.

* All the inactive carriers (Carriers whose RxLev is less than LOWER_RXLEV_THRESHOLD) and "Black List" carriers (except for Full search mode) shall be excluded from power array list for that region. All such carriers shall be moved to the end of power array list. The active carriers at the end of power array list shall occupy the place of inactive carriers. The size of the power array list shall be decremented by the number of inactive carriers. This exclusion of Inactive carriers right at the beginning, greatly improves the run time efficiency.
* The first 40 carriers belonging to each band shall be added from the top of MPH_POWER_CNF array in descending order of their strength.
* Carriers beyond 40 (41st to 60th) for each band, whose RxLev is greater than LOWER_RXLEV_THRESHOLD shall be added to the MPH_POWER_CNF list from the bottom. When the number of carriers

placed in the MPH_POWER_CNF list for any band reaches 60, all the remaining carriers for that band shall be set as Inactive carriers, so that they are excluded from further sorting.

* First 40 carriers for each band added from top, can overwrite the 41st to 60th carriers added from the bottom. The other way around is not allowed. Addition of 41st to 60th carriers from bottom is stopped once the crossover occurs.

MPH_POWER_CNF

* After the completion of carrier inclusion in MPH_POWER_CNF list, the 41st to 60th carriers for all bands present at the bottom of the list, shall be rearranged in the descending order of strength based on their RxLev and moved up the MPH_POWER_CNF list , if required.

4.2.3 Constants

The following constants a MIN_CHANNELS_PER_BAND, MAX_CHANNELS_PER_BAND are used to represent the minimum and maximum number of channels can be accommodated for each band (GSM_900, DCS_1800, PCS_1900, GSM_850).

Value
Description
MIN_CHANNELS_PER_BAND
40
Minimum number of carriers per individual band that can be added to MPH_POWER_CNF list
MAX_CHANNELS_PER_BAND
60
Maximum number of carriers per individual band that can be added to MPH_POWER_CNF list

4.2.3.1.1 Removal of existing global variables

The following existing global variables shall be removed from the T_CS_DATA structure in alr.h file. Local variables shall be used instead.

```
{
  UBYTE           c_channels_gsm;
  UBYTE           c_channels_dpcs;
} T_CS_DATA
```

4.2.4 Functional changes

The following functions in ALR shall be modified/added to implement this requirement

Function API
Description
Reference
cs_restrict_max_carriers_per_band
This is a new function. This function shall restrict the maximum number of channels per band.
4.2.4.1
cs_add_and_sort_channels

This function already exists. The existing function cs_increment_c_channels will be replaced by the new function cs_restrict_max_carriers_per_band.
4.2.4.2

4.2.4.1 cs_restrict_max_carriers_per_band

Prototype:

```
BOOL  cs_restrict_max_carriers_per_band (U16 arfcn, U8 std, U16 no_of_carriers_per_band[4])
```

Input Parameters:

```
Arfcn              :  L3 ARFCN  number as per GSM spec.
Std                 : Std value of the ARFCN.
no_of_carriers_per_band : Pointer to array of counters for the four bands( P_GSM and EGSM,
                                             DCS1800,PCS1900,850).
```

Return Value:

```
BOOL    - Tells where to add this carrier to MPH_POWER_CNF list.
ADD_AT_THE_TOP              - From the top (first 40 carrier)
ADD_AT_THE_BOTTOM      - From the bottom (41st to 60th carrier)
REACHED_THE_MAXIMUM -  All 60 carriers for a band have been added
DO_NOT_ADD                       - Do not add this carrier
```

Pseudo-code:

```
U8  cs_restrict_max_carriers_per_band (U16 arfcn, U8 std,  U16 no_of_carriers_per_band [4])
{
    Obtain the band index  based on the ARFCN and std value.
    Increment the counter for the corresponding band.
    If ( minimum number of channels(40)  for that band are added to MPH_POWER_CNF list)
    {
        if  (maximum number of carriers (60) for that band are added to MPH_POWER_CNF list)
        {
            Set all the remaining carriers from this band as Inactive carriers
        }
        return 1 i.e add from bottom to MPH_POWER_CNF list
    }
   else
    {
       return 0 i.e add from top to MPH_POWER_CNF list
    }
}
```

4.2.4.2 cs_add_and_sort_channels

Prototype:

```
void cs_add_and_sort_channels (void)
```

Input Parameters:

None

Return Value:

None

Pseudo-code:

```
void cs_add_and_sort_channels (void)
{
  U16 extra_cnf = MAX_CHANNELS;
  U8 no_of_carriers_per_band [4] = {0,   /* P-GSM and E-GSM band */
                                    0,  /* DCS 1800 band */
                                    0,  /* PCS 1900 band */
                                    0  /* 850 band  */
                                   };
  While (total number of channels added to MPH_POWER_CNF list < MAX_CHANNELS)
  {
     Obtain the strongest carrier from the power array list

     where_to_add = cs_restrict_max_carriers_per_band(arfcn, std, no_of_carriers_per_band)

    if(where_to_add EQ AT_THE_TOP)
    {
        This is first 40 carrier. Add the carrier in the I_cnf position of MPH_POWER_CNF primitive
       from the top

       Increment the I_cnf counter to move down from top
    }
    else
    {
      This is 41st  to 60th carrier. This has to be added from the bottom of MPH_POWER_CNF list.
      if(cross over has not occurred )
      {
        Add the carrier/rxlevel in the extra_cnf position of MPH_POWER_CNF primitive from the bottom
        Decrement the extra_cnf counter  to move up from bottom
      }
    }
  } // end while

   sort the extra carriers(41st to 60th carriers) and move them up if required

}
```

4.3 Region Selection

Searching for Full Service, when MS is camped on in Limited Service (in an area where multiple frequency bands are present)

In the current implementation of Cell Selection Algorithm, if the MS has found "Limited Service" in an area where multiple regions are present and the TREG timer expires, MS would search for "Full Service" only in the region where it has found "Limited Service". Consider the following scenario; requested PLMN is on the PCS 1900 Band, in an area where there is strong coverage of DCS 1800 Band carriers. If the MS cannot find the requested PLMN (on PCS 1900 Band), it enters "Limited Service" on a DCS 1800 Band carrier.

Thereafter MS will look for "Full Service" on DCS 1800 Band and GSM 900 Band carriers only. In such a case, MS will never find "Full Service" until it is able to scan across all supported Frequency Bands.

One way to solve the above problem is MS could do a non-parallel search (FAST SEARCH or NORMAL SEARCH or FULL SEARCH as described in scenarios 11, 12, 13 and 14 in Section 4.1.3.9 "Management of New Search Modes") across all supported frequency bands after a TREG timer expires. Any assumption made of the Selected Region (according to the first found suitable cell; see above), can also be cleared at the expiry of TREG timer, and the MS can start looking for "Full Service" across carriers from all regions. (Non-parallel search shall be used only if the MS is operating in an area which contains multiple frequency bands belonging to different regions.)

The disadvantage of the above solution is that MS cannot make emergency calls while searching for "Full Service".

4.3.1.1 Global Variables

The following global variables will be introduced to support this requirement.

```
{
 U8 all_freq_area;
} T_CS_DATA
```

Structure Members

Variable	Type	Size	Description
all_freq_area	U8	1	Indicates whether both American and European bands are detected in the current region.

```
{
 U8 reg_time_gap;
} T_MS_DATA
```

Structure Members

Variable	Type	Size	Description
reg_time_gap	U8	1	Indicates the time gap between Non-Parallel searches in Limited Service

Default Value : DELAY_NON_PAR_SEARCH_LIM_SER
(2 minutes)

4.3.1.2 Functionality

* The variable all_freq_area in T_CS_DATA will be set only if RR can sync to carriers from 2 different regions.
* Following TREG timer expiry in Limited Service state, non-parallel search will be issued, in case the variable all_freq_area is set to one and the MM requested service is Full Service.

4.4 Searching of carriers during 2 Scans

4.4.1 FIRST SCAN and FIRST ATTEMPT

* When requested service is ""Full Service"", RR searches all carriers from MPH_POWER_CNF for ""Full Service"". During this search RR will mark carriers as ""Emergency cell"" and ""Low Priority cell"" as it finds one. The scanning stops when RR finds a ""Full Service"" on a carrier with Normal Priority, or if the entire list is scanned.
* When the requested service is ""Limited Service"", RR searches all carriers from MPH_POWER_CNF for ""Limited Service"".

4.4.2 FIRST SCAN and SECOND ATTEMPT

* This is only applicable if requested service is ""Full Service"". RR shall search only those carriers that are marked as Low Priority.
* Function cs_def_list( ) and cs_start_sync( ) shall be modified to cater to this requirement

4.4.3 SECOND SCAN

* This is only applicable if requested service is ""Full Service"". RR shall try to reach ""Full Service"" or ""Limited Service"", but will stop searching if it finds a carrier where either service mode is possible. This is already implemented as part of CQ 24416
* RR shall first search carriers that are marked as ""Emergency cell"", and then all other carriers from the MPH_POWER_CNF list. Searching carriers that are not marked as "Emergency cell" may seem unnecessary, but on the field, it works well in areas of weak coverage.
* Function cs_def_list ( ) and cs_start_sync( ) shall be modified to meet the above requirement.

4.4.4 Global Variables

The following new global variables will be introduced to support this requirement.

```
{
 U8 scan_mode;
} T_CS_DATA
```

Structure Members

Variable

Type

Size

Description

scan_mode

U8 enum

1

Identifies the current scan mode. Can take four enum values

CS_NO_SCAN

CS_FIRST_SCAN_FIRST_ATTEMPT

CS_FIRST_SCAN_SECOND_ATTEMPT
CS_SECOND_SCAN
This variable shall be set to CS_NO_SCAN during initialization and following completion of cell selection.

4.4.5 Functional changes

The following functions in RR shall be modified to implement this requirement

Function
Description
Reference
cs_start_sync
This function currently resets the CHECK BIT for all the channels. This shall be modified as described below.

Cs_def_list
This function currently checks whether cells belonging to a particular Attribute are present or not.

4.4.5.1 cs_start_sync

Prototype:
```
void cs_start_sync (   )
```
Input Parameters:
None
Return Value:
None
Pseudo-code :
Only the modifications are described here
```
void cs_start_sync(   )
{
   Resets the CHECK BIT for all the channels reported in
mph_power_cnf  only for Second Scan and if the search mode is not
Fast search mode
 }
```

4.4.5.2 cs_def_list

Prototype:
```
void cs_def_list (U8 attribute
```
Input Parameters:
Attribute - Indicates the Attribute flag
Return Value:
TRUE   - Cells with the passed Attribute are present
FALSE   - Cells with the passed attribute are not present
Pseudo-code :
Only the modifications are described here
```
void cs_def_list(U8 attribute   )
{
   Reset the CHECK BIT for all  Channels found  with the given
Attribute set
 }
```

5 Interface Changes

This section describes the changes required to the MPH SAP between RR and ALR for Cell Selection Improvements feature implementation.

5.1 Introduction of new constants

The following new constants are introduced in MPH SAP between RR and ALR to support Cell Selection Improvements feature.

Constants

Value

Description

LOWER_RXLEV_THRESHOLD

4

Identifies the Lower Threshold for the signal level of a carrier. All carriers weaker than this are not included in the mph_power_cnf list

FAST_SEARCH_MODE

0x01

Identifies "Fast Search"

NORMAL_SEARCH_MODE

0x02

Identifies "Normal Search"

FULL_SEARCH_MODE

0x03

Identifies "Full Search"

BLACKLIST_SEARCH_MODE

0x04

Identifies "Black List Search"

FULL_SEARCH_MODE_ATTEMPTS

5

The number of search mode attempts for Full Search. This is defined by the standard

NORMAL_SEARCH_MODE_ATTEMPTS

5

The number of search mode attempts for Normal search. This is defined by the standard

TIM_FULL_SEARCH_POWERMEAS_VAL

4000ms

Power measurements spreading time for Full Search mode. This value is defined by the standard

TIM_NORMAL_SEARCH_POWERMEAS_VAL

4000ms

Power measurements spreading time for Normal search mode. This value is defined by the standard

5.2 Primitive changes

The interface between RR and ALR requires changes to support new requirements for Cell Selection Improvements feature. The following primitives are modified/added in MPH SAP between RR and ALR.

Primitive

Direction

Type

Reference

MPH_POWER_REQ

RR --> ALR

T_MPH_POWER_REQ

5.2.1

MPH_POWER_CNF
ALR --> RR
T_MPH_POWER_CNF
5.2.2

5.2.1 MPH_POWER_REQ

RR uses this primitive to request ALR for "Parallel" or "Not-Parallel" search for GSM channels.

```
{
   U8   pch_interrupt;
   U8   freq_bands;
   U8   search_mode;
   U8   lower_rxlev_threshold;
   T_BLACK_LIST  black_list;
   T_WHITE_LIST  white_list;
}  T_MPH_POWER_REQ
```

Where :

```
{
  T_LIST list[2];
} T_BLACK_LIST

{
  U8 white_list_valid;
  U8 region;
  T_LIST list;
} T_WHITE_LIST
```

T_MPH_POWER_REQ members

Type
Size
Description
pch_interrupt
U8
1
Takes two values.
0x00 - Power measurements with PCH listening
0x01 - Power measurements without PCH listening
freq_bands
U8
1
Identifies the list of GSM frequency bands over which search for channel is requested
search_mode
U8
1
Identifies the search mode. Can take four values
0x01   - FAST_SEARCH_MODE
0x02   - NORMAL_SEARCH_MODE
0x03   - FULL_SEARCH_MODE
0x04   - BLACKLIST_SEARCH_MODE
Lower_rxlev_threshold
U8

1
Contains the lower threshold of RxLev
black_list
T_BLACK_LIST
1
Contains a separate bit map of Black Listed carries for Euro / American regions. In case of "Black List search, it contains "Grey" carriers also
white_list
T_WHITE_LIST
1
Contains the bitmap of carriers present in the "White List". This list is empty in case of "Black List search".

T_BLACK_LIST members

Type
Size
Description
list
T_LIST
2 * T_LIST
Each bit represents one carrier in the range 0 - 1024 (1024/8 = 128).
Bit value
2 - Carrier is part of "Black List"
0   - Carrier is not part of "Black List"

T_WHITE_LIST members

Type
Size
Description
white_list_valid
U8
1
Indicates whether "White List" is valid or not
0x00 - Not valid
0x01 - Valid
region
U8
1
Indicates whether "White List" belongs to European or American region
0x00 - European region
0x01 - American region
list
T_LIST
1
Bitmap for "White List" carriers

5.2.2 MPH_POWER_CNF

ALR uses this primitive to provide the list of carriers scanned by Layer 1. The list contains the ARFCN and RXLEV values. It also contains the list of inactive carriers.

```
{
  U8  num_of_chan;
  U8  num_of_white_list_chan;
  U16 arfcn[ MAX_CHANNELS] ;
  U8  rxlev [ MAX_CHANNELS] ;
  T_BLACK_LIST inactive_carrier_list;
} T_MPH_POWER_CNF
```

T_MPH_POWER_CNF members

Type

Size

Description

num_of_chan

U8

1

Total number of detected channels

num_of_white_list_chan

U8

1

The number of "White List" carriers included in the list. These carriers are put at the top of the list.

Arfcn

U16

MAX_CHANNELS

channel number

rxlev

U8

MAX_CHANNELS

received field strength

inactive_carrier_list

T_BLACK_LIST

1

Contains a separate bit map of carriers which are not grey or black anymore for Euro / American regions

6 Configurable Parameters

This section describes all the configuration commands that are added as part of this feature.

6.1 Configuration Commands

The following Dynamic configuration commands will be introduced in RR to support Cell Selection Improvements feature.

Command

Format

Range

Description

TIM_FAST

TIM_FAST <val>

0

Value in minutes

Configures the value of TFAST_CS timer used during Fast Search.
Fast Search is disabled when the value = 0.
TIM_NORMAL
TIM_NORMAL <val>
1
Value in minutes
Configures the value of TNORMAL_CS timer used during Normal
Search. Normal Search is disabled when the value = 0.
ERASE_BL
ERASE_BL
N.A.
Erases the Black List both in RAM as well as FFS
ERASE_WL
ERASE_WL
N.A
Erases the White List both in RAM as well as FFS
SET_BL
SET_BL < region, upto 5 arfcns >

This command is used to add GSM channels to "Black List".
This can be of immense use during windows simulation testing
SET_WL
SET_WL < region, upto 5 arfcns >

This command is used to add GSM channels to "White List"
This can be of immense use during windows simulation testing
SET_WL_REG
SET_WL_REG <region>
0,1
This command is used to set region information in white list.
This can be of immense use during windows simulation testing
SET_WL_PLMN
Set_WL_PLMN <mcc, mnc)

This command is used to set the PLMN ID of the white list stored
on Flash.
White list shall be used only when its PLMN ID matches with the
requested PLMN
BL_CS
BL_CS <val>
0, 1
Controls Black List Search
0 - Black List Search is disabled
1 - Black List Search is enabled
U_RXT
U_RXT <val>
0 to 63
Configures the Upper rxlev threshold
M_RXT
M_RXT <val>
0 to 63
Configures the Medium rxlev threshold
L_RXT
L_RXT <val>

0 to 63
Configures the Lower RxLev threshold
FBLS
FBLS

Forces Black List search. Can be used during testing.
SET_NPS_DELAY
SET_NPS_DELAY <delay in seconds>
>= 0
Used to set the time delay between Non-Parallel searches in Limited service when reg_counter is less than 20

The following configuration commands will be removed.
1. ID_CLEAR_BCCH_INFO - This is now replaced by ERASE_WL command
2. ID_PCM - PCM is no longer used for storing BCCH(White List) information.

6.2 Global Variables

A new variable will be introduced in T_RR_DATA structure in rr.h file as shown below for Dynamic Configuration Commands.

```
{
  T_DYNAMIC_CONFIG  dyn_config;
}  T_RR_DATA
```

Structure Members

Type
Size
Description
dyn_config
T_DYNAMIC_CONFIG
1
Used to store all the dynamic configuration Command variables

6.3 Funtional changes

The following functions in RR will be modified to support the new dynamic configuration commands.

Function
Description
Reference
cs_init_process
This is an existing function. This will be modified to initialize the dynamic configuration variables to default values
4.2.3.1

6.3.1 cs_init_process

Prototype:
void cs_init_process ( )
Input Parameters:
None
Return Value:
None
Pseudo-code :
Only the modifications are described here

```
void cs_init_process (  )
{
   Enable Black list search mode
   Set TFAST_CS and TNORMAL_CS timer values to 4 min
  Set upper and medium level threshold to default values
}
```

7 Common Library for List Processing Functions

The Channel List processing functionality is currently used only by RR module and hence is implemented in the RR file rr_srv.c. Since a part of this functionality is now required in ALR also for Cell Selection Improvements feature, these functions shall be moved to a common library to avoid duplication of code.

The following two new files shall be added to the common library.

File Name
Location
Functionality
cl_list.h
/g23m/condat/com/include
Contains declarations of List processing functions. Any source file using List processing functionality shall include this header file.
cl_list.c
/g23m/condat/com/src/comlib
Contains definitions for all List processing functions.

7.1 Functions

The following new functions have been added to List processing library

The following functions in RR shall be modified to implement this requirement

Function
Description
Reference
srv_unmask_list
Resets all the bits in the "target" that are set in the "source"
7.1.1
Srv_count_list
Returns the count of the number of channels set in the list
7.1.2
Srv_is_list_set
Checks whether at least one channel is set in the list or not
7.1.3
srv_trace_freq_in_list
Traces all the channels that are set in the list
7.1.4
srv_get_region_from_std
Derives the "region" from "std"
7.1.5

7.1.1 srv_unmask_list

Prototype:

```
void srv_unmask_list ( T_LIST *target, T_LIST *source  )
```

Description:

This function resets all the bits in the "target" that are set in the "source". This function is used to update "Black List" with BA and MA lists.

Input Parameters:

target : destination list

source : source list

Return Value:

None

Pseudo-code :

```
void srv_unmask_list (   )
{
   Reset all those bits in target list, which are also set in source list ( INVERT and AND )
}
```

7.1.2 srv_count_list

Prototype:

```
void U16 srv_count_list ( T_LIST *list  )
```

Description:

This function returns the number of GSM channels set in the list.

Input Parameters:

List : pointer to T_LIST structure

Return Value:

Returns the number of channels set in the list

Pseudo-code :

```
void U16 srv_count_list (   )
{
   Return the number of channels set in the list
}
```

7.1.3 srv_is_list_set

Prototype:

```
void BOOL  srv_is_list_set ( T_LIST *list  )
```

Description:

This function checks whether any GSM channel is set in the list or not.

Input Parameters:

List : pointer to T_LIST structure

Return Value:

TRUE : channel is set

FALSE : list is empty

Pseudo-code :

```
void BOOL srv_is_list_set (   )
{
   Return whether any GSM channel is set in the list or not
}
```

7.1.4 srv_trace_freq_in_list

Prototype:

```
void srv_trace_freq_in_list ( T_LIST *list)
```

Description:

This function traces all the GSM channels set in the list

Input Parameters:

list : pointer to T_LIST structure

Return Value:

None

Pseudo-code :

```
void srv_trace_freq_in_list (   )
{
   Trace all the GSM channels set in the list
}
```

7.1.5 srv_get_region_from_std

Prototype:

```
U8 srv_get_region_from_std ( U8 std  )
```

Description:

This function derives the "Region" information from "Band" information.

Input Parameters:

std   :   band information

Return Value:

region   :   European / American region

Pseudo-code :

```
U8 srv_get_region_from_std ( U8 std  )
{
   Return the region information derived from the std
}
```

The following List processing functions have been moved to cl_list.c file from rr_srv.c file.

* srv_set_channel
* srv_unset_channel
* srv_get_channel
* scr_channel_bit
* srv_create_list
* srv_clear_list
* srv_copy_list
* srv_compare_list
* srv_merge_list
* setBit
* getBit
* resetBit

8 Approach to reduce the number of search

The existing sorting implementation in function cs_add_and_sort_channels does the following number of searches,

(MAX_CHANNELS) * max1 * max2 times.

MAX_CHANNELS = Maximum carriers can be added in to the MPH_POWER_CNF
Max1 = Number of carriers measured by L1 in European Region
Max2 = Number of carriers measured by L1 in American Region

Note: In Cell Selection Improvement Feature the size of MAX_CHANNELS has been increased from 80 to 160 carriers.

In existing code we are going through the entire lists (European / American) of power_result array to find a carrier, which has the maximum rxlevel. After finding the biggest carrier the carrier will be added in to the MPH_POWER_CNF and the rxlevel will be set to min_rxlevel -1. The rxlevel will be set to low because we want to exclude the carrier during the next search. By doing this we could make the next highest rxlevel carrier to become first during the next search.

Here we are just excluding the carrier by setting the rxlevel to low. But actually the carrier will be searched every time during sorting. This searching on the Low rxlevel carriers can be avoided by following the below approach.
Code Snippet:

```
=============================================
    if (max1)
    {
      parray = p_results1->power_array;
      for (i1=0; i1 < max1; i1++, parray++)        <== Searching
is always on all the carriers (including inactive)
      {
        if (parray->accum_power_result > rxlev)   <== Always
comparing the carriers (including inactive[x1])
        {
          pbig = parray;
          rxlev = parray->accum_power_result;
          radio_band_config = std1;
        }
      }
    }

    if (max2)
    {
      parray = p_results2->power_array;
      for (i2=0; i2 < max2; i2++, parray++)   <== Searching is
always on all the carriers (including inactive)
      {
        if (parray->accum_power_result > rxlev)  <== Always
comparing the carriers (including inactive[x2])
        {
          pbig = parray;
          rxlev = parray->accum_power_result;
          radio_band_config = std2;
        }
      }
    }
====================================
```

When the carrier will be set to less than LOWER_RXLEVEL_THRESHOLD:
1. Carriers reported by L1 may have LOWER_RXLEVEL_THRESHOLD
2. Every time after adding the biggest rxlevel carrier to MPH_POWER_CNF we set the rxlevel to less than min rxlevel.

How can we avoid searches on the inactive (low rxlevel) carriers?
1. Carriers reported by L1 has LOWER_RXLEVEL_THRESHOLD
During the first search we can find the inactive carriers reported by L1. After identifying those carriers it can be swapped with the last carrier of the p_results array. The maximum number of channels in the p_results array can be decremented by 1.

2. After adding the carrier to MPH_POWER_CNF
After adding the carrier to POWER_CNF instead of setting to min_rxlevel -1. We can swap the particular carrier with the last

carrier of the p_results array. The maximum number of channels in the p_result array can be decremented by 1.

Example:
Original List:
1, 2, 3, 4, 5   count (max) = 5
Just consider that the 2nd carrier is inactive we can swap that carrier with the last carrier 5 and reduce the count to 4.

After removing the inactive carrier:
      1, 5, 3, 4 count (max) = 4

Example - how this approach would reduces the number of searches
Just consider L1 measured 124 carriers (European/American), on that there were around 50 carriers (on each region) are less than Rxlevel threshold.

During the first search the power_result will be searched 124 times (on both region), but from the next search onwards the array will be searched only 104 times (if we use this approach).

In the above scenario we could save the following number of searches
        ((MAX_CHANNELS - 1) * 50 * 50)

Overhead:
1) But here the overhead will be the swapping. After finding the inactive carrier the particular carrier needs to be swapped with the last carrier in the p_results array.
2) In function cs_increment_c_channels (as part of CSImp feature the function name changed to cs_restrict_max_carriers_per_band) after reaching the maximum channel for a band the rxlevel for the remaining carriers will be set to less than the min rxlevel. Here we assume that the order of the carriers was not changed (sequential order used in L1). This may not be possible if we use this approach. We may have to search through the whole p_results array to find a particular carrier. Because the order of the carriers were changed by swapping the inactive carrier with the last carrier in p_results array.

Assumption
      There is an assumption that nowhere we are going to use the p_results array other than the Initial power request. Because here we are changing the order of the p_results array which was sent by L1.

9 Deviations from HLD

* Black List shall be stored in FFS during switch off irrespective of the current RR service state. Unlike White List, the validity of "Black List" has no dependency on the RR service state. It contains up to date information about the Black List carriers in a given area at a given time and must always be stored during switch off.

* White list shall be used only when its PLMN ID matches with the requested PLMN

Appendix A. Open Questions

1. Can the black list be stored independent of the service at switch off? This is useful in cases where the device does not have a SIM inserted, and after power on the device has to find limited service as soon as possible?

Maybe it makes sense to store the Black List during switch OFF if the TFAST_CS timer is active. This might be useful when a user in bad signal conditions try to "reset" the mobile.

2. What are the other possible candidates for the White List?

o Cells contained in the IE BA List Pref from the message Channel Release

o a cell, which is used very often as a serving cell in the past.

3. Is the FAST SEARCH also useful if requested service is "Limited Service"? In these scenarios a quick re-finding of at least limited service is also important.

4. MS cannot do a BLACK LIST SEARCH during Dedicated mode (voice call or GPRS Packet Transfer). If the MS has moved several Location Areas (or Routing Areas) during Dedicated mode, then the Black List may become inconsistent.

How do we handle this?

Can the Black list be erased if the MS has moved several location areas in PTM or dedicated mode?

[x1]

Here we know most of the carriers are inactive (By L1 or added to POWER_CNF) even though we are doing searching on the whole list reported by L1 always.

[x2]

Here we know most of the carriers are inactive (By L1 or added to POWER_CNF) even though we are doing searching on the whole list reported by L1 always.

Deviations from HLD