Technical Note

WCDMA/HSPA Scenario Version (2)

MD8480C
W-CDMA Signalling Tester

Details of Sample C–Scenario Part 2/2

How to Use Scenario Library and Structure with "W_01_Packet_MD8480.c"

Version 1.0
Anritsu Corporation

Contents

Common Information

Transport Layer Structure Common information:

- **DTXPosition**

"MD8480C easy to" – A.2.2.5.2 Description of CPHY_TRCH_CONFIG_PAR structure
Specifies position of DTX after RateMatching for DownLink
Value in scenario description
DTX_FIXED_POSITION ———— Fixed Position
DTX_FLEXIBLE_POSITION —— Flexible Position

- **PuncLimit**

"MD8480C easy to" – A.2.2.5.2 Description of CPHY_TRCH_CONFIG_PAR structure
Specifies Puncturing Limit used in RateMatching for UpLink
Value in scenario description
50..100 (integer value) ——— 50% to 100%

Mac Layer Structure Common information:

The following parameters should be the same as CPHY_TRCH_CONFIG_PAR.
NumOfTrch
TFCS
TrchInfo
Set the same value at CMAC_CONFIG_PAR or copy the value from CPHY_TRCH_CONFIG_PAR.

```
macConfig–>NumOfTrch   = phyTrchConfig–>NumOfTrch;
macConfig–>TFCS        = phyTrchConfig–>TFCS;
memcpy (macConfig–>TrchInfo, phyTrchConfig–>TrchInfo, sizeof(macConfig–>TrchInfo));
```


Calculation for Rlc-Size in L3 Message (TrchInfo[n].Dynamic[n].TBSize)

```
TS25.331
BitModeRLC–SizeInfo ::= CHOICE {
        sizeType1               INTEGER (0..127),
        — Actual value sizeType2 = (part1 * 8) + 128 + part2
        sizeType2               SEQUENCE {
                part1           INTEGER (0..15),
                part2           INTEGER (1..7)  OPTIONAL
        },
        — Actual value sizeType3 = (part1 * 16) + 256 + part2
        sizeType3               SEQUENCE {
                part1           INTEGER (0..47),
                part2           INTEGER (1..15)         OPTIONAL
        },
        — Actual value sizeType4 = (part1 * 64) + 1024 + part2
        sizeType4               SEQUENCE {
                part1           INTEGER (0..62),
                part2           INTEGER (1..63)         OPTIONAL
        }
}

TS25.331
OctetModeRLC–SizeInfoType1 ::=          CHOICE {
                — Actual size = (8 * sizeType1) + 16
        sizeType1       INTEGER (0..31),
        sizeType2       SEQUENCE {
                — Actual size = (32 * part1) + 272 + (part2 * 8)
                part1   INTEGER (0..23),
                part2   INTEGER (1..3)  OPTIONAL
        },
        sizeType3       SEQUENCE {
                — Actual size = (64 * part1) + 1040 + (part2 * 8)
                part1   INTEGER (0..61),
                part2   INTEGER (1..7)  OPTIONAL
        }
}

TS25.331
OctetModeRLC–SizeInfoType2 ::=          CHOICE {
        — Actual size = (sizeType1 * 8) + 48
        sizeType1       INTEGER (0..31),
        — Actual size = (sizeType2 * 16) + 312
        sizeType2       INTEGER (0..63),
        — Actual size = (sizeType3 *64) + 1384
        sizeType3       INTEGER (0..56)
}
```

BCCH–BCH–P–CCPCH

TS25.302 Annex A

Table A1: Characterisation of Transport Format

		Attribute values	BCH	PCH	FACH	RACH
Dynamic part	Transport Block Size	0 to 5000 1 bit granularity	246	1 to 5000 1 bit granularity	0 to 5000 1 bit granularity	0 to 5000 1 bit granularity
	Transport Block Set Size	0 to 200 000 1 bit granularity	246	1 to 200 000 1 bit granularity	0 to 200 000 1 bit granularity	0 to 200 000 1 bit granularity
	Transmission Time Interval (option for TDD only)	10, 20 ms, 40 and 80 ms				
Semi-static part	Transmission Time Interval (FDD, option for TDD NRT bearers)	10, 20 ms, 40 and 80 ms	20 ms	10 ms for FDD, 20 ms for TDD	10, 20 ms, 40 and 80 ms	10 ms and 20 ms for FDD, 10 ms for 3.8477.68 Mcps TDD 5 ms, 10 ms and 20 ms for 1.28 Mcps TDD
	Type of channel coding	No Coding (TDD only) Turbo coding Convolutional coding	Convolutional coding	Convolutional coding	No coding (TDD only) Turbo coding Convolutional coding	Convolutional coding
	Code rates	1/2, 1/3	1/2 for FDD and 3.8477.68 Mcps TDD 1/3 for 1.28 Mcps TDD	1/2 for FDD and 3.8477.68 Mcps TDD 1/2, 1/3 for 1.28 Mcps TDD	1/2, 1/3	1/2
	CRC size	0, 8, 12, 16, 24	16	0, 8, 12, 16, 24	0, 8, 12, 16, 24	0, 8, 12, 16, 24
	Resulting ratio after static rate matching	0,5 to 4				

TR25.944 4.1.1.1 BCH

TS25.212 4.2.3 Channel coding

Table 1: Parameters for BCH

Transport block size	246
CRC	16 bits
Coding	CC, coding rate = 1/2
TTI	20 ms
The number of codes	1
SF	256

Table 1: Usage of channel coding scheme and coding rate

Type of TrCH	Coding scheme	Coding rate
BCH	Convolutional coding	1/2
PCH		
RACH		
DCH, FACH		1/3, 1/2
	Turbo coding	1/3

```
CphyRlSetupPar = &CphyRlSetup_P_CCPCH_BTS1;
memset( CphyRlSetupPar, 0, sizeof(CPHY_RL_SETUP_PAR) );

CphyRlSetupPar->Offset      = 7680;
CphyRlSetupPar->ScrCode     = 0x00000090;
CphyRlSetupPar->SlotFormat;                          // Don't care
CphyRlSetupPar->SymbolRate  = SYMRATE15K;            // *1
CphyRlSetupPar->ChCode      = 1;                     // *2
CphyRlSetupPar->Power       = POWER_STEP_01DB(-160);// Power = -16.0dB
CphyRlSetupPar->TxDiversity = DIVERSITY_OFF;


CphyTrchConfigPar = &CphyTrchConfig_P_CCPCH_BTS1;
memset( CphyTrchConfigPar, 0, sizeof(CPHY_TRCH_CONFIG_PAR) );

CphyTrchConfigPar->DTXPosition = DTX_FLEXIBLE_POSITION;
CphyTrchConfigPar->InterLv2nd  = INTERLEAVE_ON;
CphyTrchConfigPar->PunclLimit;  /* Don't care */

CphyTrchConfigPar->TFCS.NumOfTFC  = 2;
CphyTrchConfigPar->TFCS.TFC[0][0] = 0;
CphyTrchConfigPar->TFCS.TFC[1][0] = 1;

CphyTrchConfigPar->NumOfTrch    = 1;
CphyTrchConfigPar->TrchInfo[0].Trch              = D_BCH;
CphyTrchConfigPar->TrchInfo[0].TrchNo            = 07
CphyTrchConfigPar->TrchInfo[0].InterLvlst        = INTERLEAVE_ON;

CphyTrchConfigPar->TrchInfo[0].NumOfDynamic      = 2;
CphyTrchConfigPar->TrchInfo[0].Dynamic[0].NumOfTB = 0;
CphyTrchConfigPar->TrchInfo[0].Dynamic[0].TBSize  = 246;
CphyTrchConfigPar->TrchInfo[0].Dynamic[1].NumOfTB = 1;
CphyTrchConfigPar->TrchInfo[0].Dynamic[1].TBSize  = 246;

CphyTrchConfigPar->TrchInfo[0].Static.TTI        = 2;
CphyTrchConfigPar->TrchInfo[0].Static.EPType     = CODING CONV;
CphyTrchConfigPar->TrchInfo[0].Static.CodingRate  = CODINGRATE1_2;
CphyTrchConfigPar->TrchInfo[0].Static.CRC_Size    = 16;
CphyTrchConfigPar->TrchInfo[0].Static.RM_Attr     = 1;


CmacConfigPar = &CmacConfig_P_CCPCH_BTS1;
memset( CmacConfigPar, 0, sizeof(CMAC_CONFIG_PAR) );

CmacConfigPar->ActFlag   = MAC_ACTIVE;

CmacConfigPar->NumOfLoch[0] = 1;
CmacConfigPar->LochInfo[0][0].Loch       = D_BCCH;
CmacConfigPar->LochInfo[0][0].LochNo     = 07
CmacConfigPar->LochInfo[0][0].Priority   = 0;
CmacConfigPar->LochInfo[0][0].CTLength   = 0;
CmacConfigPar->LochInfo[0][0].CTValue    = 0;
CmacConfigPar->LochInfo[0][0].TCTFLength = 0;
CmacConfigPar->LochInfo[0][0].UEIDType   = UEID_NOTUSE;
CmacConfigPar->LochInfo[0][0].UEID       = 0;



CrlcConfigPar = &CrlcConfig_BCCH;
memset( CrlcConfigPar, 0, sizeof(CRLC_CONFIG_PAR) );

CrlcConfigPar->NumOfPUs          = 1;
CrlcConfigPar->PU_LengthTM       = 0;   // It will be ignored in this scenario. (TMD-PDU)
                                        // Refer to "Easy to… A.1.10 CrlcConfig()"
CrlcConfigPar->PU_LengthUM       = 238;
CrlcConfigPar->PU_LengthAM       = 230;
```


CCCH–RACH–PRACH

System Information Block–5

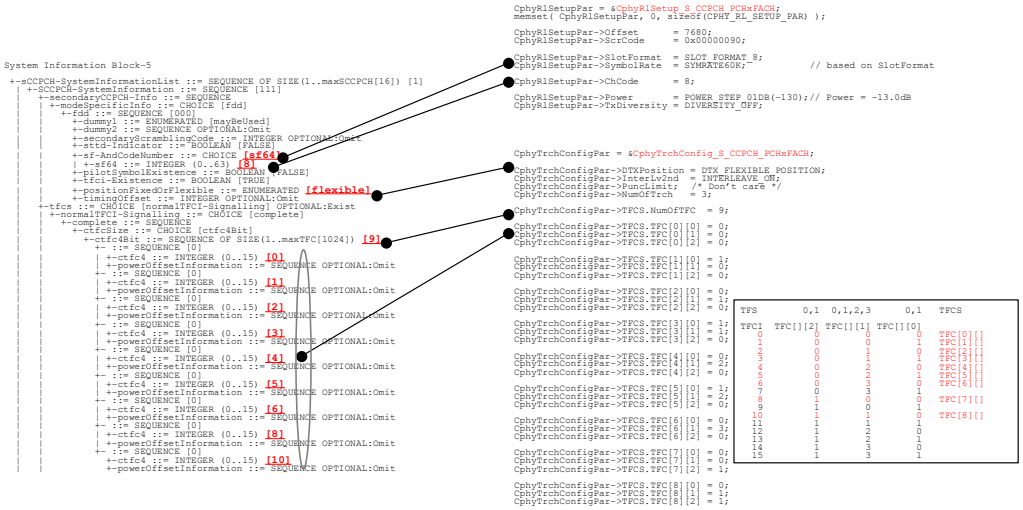
```
+-prach-SystemInformationList ::= SEQUENCE OF SIZE(1..maxPRACH[16]) [1]
| +-PRACH-SystemInformation ::= SEQUENCE [11101]
|   +-prach-RACH-Info ::= SEQUENCE
|   | +-modeSpecificInfo ::= CHOICE [fdd]
|   |   +-fdd ::= SEQUENCE
|   |     +-availableSignatures ::= BIT STRING
|   |     +-availableSF ::= ENUMERATED [sfpr64]
|   |     +-preambleScramblingCodeWordNumber ::= INTEGER (0..15) [0]
|   |     +-puncturingLimit ::= ENUMERATED [pl1]
|   |     +-availableSubChannelNumbers ::= BIT STRING
|   +-transportChannelIdentity ::= INTEGER (1..32) [3]
|   +-rach-TransportFormatSet ::= CHOICE [commonTransChTFS] OPTIONAL:Exist
|   | +-commonTransChTFS ::= SEQUENCE
|   |   +-tti ::= CHOICE [tti20]
|   |   | +-tti20 ::= SEQUENCE OF SIZE(1..maxTF[32]) [2]
|   |   |   +-CommonDynamicTF-Info ::= SEQUENCE
|   |   |   | +-rlc-Size ::= CHOICE [fdd]
|   |   |   |   +-fdd ::= SEQUENCE
|   |   |   |     +-octetModeRLC-SizeInfoType2 ::= CHOICE [sizeType1]
|   |   |   |     | +-sizeType1 ::= INTEGER (0..31) [15]
|   |   |   |     +-numberOfTbSizeList ::= SEQUENCE OF SIZE(1..maxTF[32]) [1]
|   |   |   |     | +-NumberOfTransportBlocks ::= CHOICE [one]
|   |   |   |     |   +-one ::= NULL
|   |   |   |     +-logicalChannelList ::= CHOICE [configured]
|   |   |   |       +-configured ::= NULL
|   |   |   +-CommonDynamicTF-Info ::= SEQUENCE
|   |   |     +-rlc-Size ::= CHOICE [fdd]
|   |   |       +-fdd ::= SEQUENCE
|   |   |         +-octetModeRLC-SizeInfoType2 ::= CHOICE [sizeType2]
|   |   |         | +-sizeType2 ::= INTEGER (0..63) [3]
|   |   |         +-numberOfTbSizeList ::= SEQUENCE OF SIZE(1..maxTF[32]) [1]
|   |   |         | +-NumberOfTransportBlocks ::= CHOICE [one]
|   |   |         |   +-one ::= NULL
|   |   |         +-logicalChannelList ::= CHOICE [configured]
|   |   |           +-configured ::= NULL
|   |   +-semiStaticTF-Information ::= SEQUENCE
|   |     +-channelCodingType ::= CHOICE [convolutional]
|   |     | +-convolutional ::= ENUMERATED [half]
|   |     +-rateMatchingAttribute ::= INTEGER (1..maxRM[256]) [1]
|   |     +-crc-Size ::= ENUMERATED [crc16]
|   +-rach-TFCS ::= CHOICE [normalTFCI-Signalling] OPTIONAL:Exist
|   | +-normalTFCI-Signalling ::= CHOICE [complete]
|   |   +-complete ::= SEQUENCE
|   |     +-ctfcSize ::= CHOICE [ctfc2Bit]
|   |       +-ctfc2Bit ::= SEQUENCE OF SIZE(1..maxTFC[1024]) [2]
|   |       | +- ::= SEQUENCE [1]
|   |       | | +-ctfc2 ::= INTEGER (0..3) [0]
|   |       | | +-powerOffsetInformation ::= SEQUENCE [1] OPTIONAL:Exist
|   |       | |   +-gainFactorInformation ::= CHOICE [signalledGainFactors]
|   |       | |   | +-signalledGainFactors ::= SEQUENCE [0]
|   |       | |   |   +-modeSpecificInfo ::= CHOICE [fdd]
|   |       | |   |   | +-fdd ::= SEQUENCE
|   |       | |   |   |   +-gainFactorBetaC ::= INTEGER (0..15) [15]
|   |       | |   |   +-gainFactorBetaD ::= INTEGER (0..15) [15]
|   |       | |   |   +-referenceTFC-ID ::= INTEGER OPTIONAL:Omit
|   |       | |   +-powerOffsetPp-m ::= INTEGER (-5..10) [2] OPTIONAL:Exist
|   |       | +- ::= SEQUENCE [1]
|   |       |   +-ctfc2 ::= INTEGER (0..3) [1]
|   |       |   +-powerOffsetInformation ::= SEQUENCE [1] OPTIONAL:Exist
|   |       |     +-gainFactorInformation ::= CHOICE [signalledGainFactors]
|   |       |     | +-signalledGainFactors ::= SEQUENCE [0]
|   |       |     |   +-modeSpecificInfo ::= CHOICE [fdd]
|   |       |     |   | +-fdd ::= SEQUENCE
|   |       |     |   |   +-gainFactorBetaC ::= INTEGER (0..15) [11]
|   |       |     |   +-gainFactorBetaD ::= INTEGER (0..15) [15]
|   |       |     |   +-referenceTFC-ID ::= INTEGER OPTIONAL:Omit
|   |       |     +-powerOffsetPp-m ::= INTEGER (-5..10) [2] OPTIONAL:Exist
```

TS25.321 Table 9.2.1.4:
Coding of the Target Channel Type Field on RACH for FDD

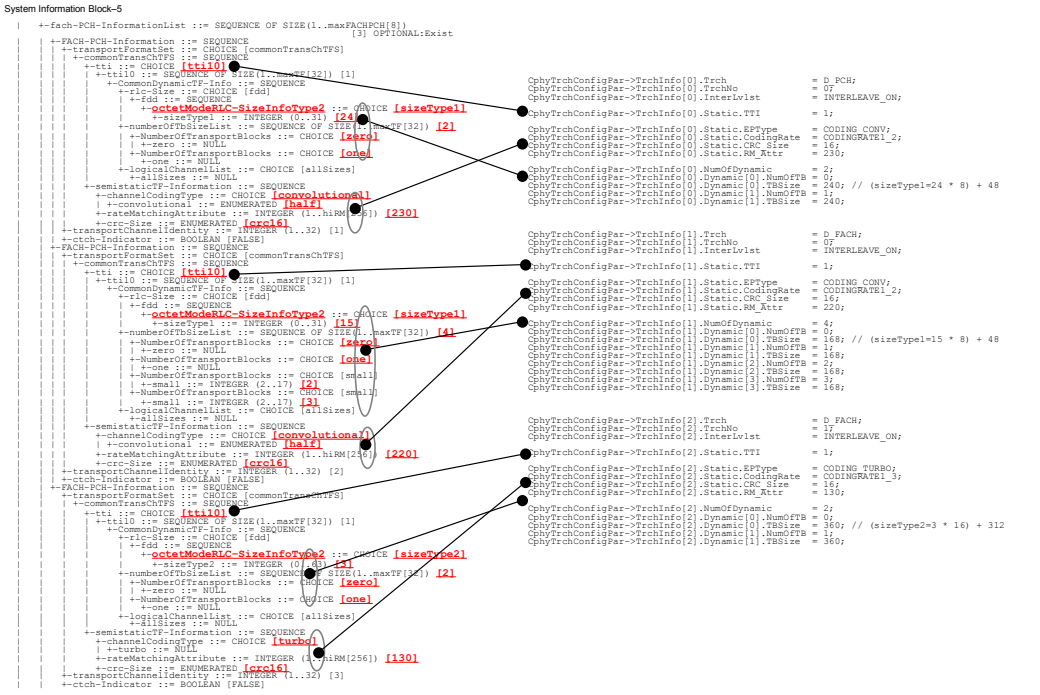
TCTF	Designation
00	CCCH
01	DCCH or DTCH over RACH
10–11	Reserved

```
CphyRlSetupPar = &CphyRlSetup_U_PRACH;
memset( CphyRlSetupPar, 0, sizeof(CPHY_RL_SETUP_PAR) );

CphyRlSetupPar->Offset      = 10240;
CphyRlSetupPar->ScrCode     = 0x01000090;

CphyRlSetupPar->SlotFormat;                          // Don't care
CphyRlSetupPar->SymbolRate  = SYMRATE60K;            // for message data field

CphyRlSetupPar->ChCode      = 128;      // It will be ignored in this scenario.
// Refer to "Easy to… A.1.10 A.2.2.6.2 Description of CPHY_RL_SETUP_PAR structure
//                                                  - PreambleRxMode"
// Refer to TS25.213 - 4.3.1.3

CphyRlSetupPar->Power;                               // Don't care
CphyRlSetupPar->AICHTiming  = AICH_3ACCESS_SLOT;
CphyRlSetupPar->TxDiversity = DIVERSITY_OFF;


CphyTrchConfigPar = &CphyTrchConfig_U_PRACH;
memset( CphyTrchConfigPar, 0, sizeof(CPHY_TRCH_CONFIG_PAR) );

CphyTrchConfigPar->DTXPosition;/* Don't care */
CphyTrchConfigPar->InterLv2nd  = INTERLEAVE_ON;
CphyTrchConfigPar->PunclLimit  = 100;
CphyTrchConfigPar->NumOfTrch   = 1;

CphyTrchConfigPar->TFCS.NumOfTFC  = 2;
CphyTrchConfigPar->TFCS.TFC[0][0] = 0;
CphyTrchConfigPar->TFCS.TFC[1][0] = 1;

CphyTrchConfigPar->TrchInfo[0].Trch              = U_RACH;
CphyTrchConfigPar->TrchInfo[0].TrchNo            = 07
CphyTrchConfigPar->TrchInfo[0].InterLvlst        = INTERLEAVE_ON;

CphyTrchConfigPar->TrchInfo[0].Static.TTI        = 2;

CphyTrchConfigPar->TrchInfo[0].Static.EPType     = CODING_CONV;
CphyTrchConfigPar->TrchInfo[0].Static.CodingRate  = CODINGRATE1_2;
CphyTrchConfigPar->TrchInfo[0].Static.CRC_Size    = 16;
CphyTrchConfigPar->TrchInfo[0].Static.RM_Attr     = 1;

CphyTrchConfigPar->TrchInfo[0].NumOfDynamic      = 2;
CphyTrchConfigPar->TrchInfo[0].Dynamic[0].NumOfTB = 1;
CphyTrchConfigPar->TrchInfo[0].Dynamic[0].TBSize  = 168;      // (sizeType1=15 * 8) +
CphyTrchConfigPar->TrchInfo[0].Dynamic[1].NumOfTB = 1;
CphyTrchConfigPar->TrchInfo[0].Dynamic[1].TBSize  = 360;      // (sizeType2=3 * 16) +


CmacConfigPar = &CmacConfig_U_PRACH;
memset( CmacConfigPar, 0, sizeof(CMAC_CONFIG_PAR) );

CmacConfigPar->ActFlag   = MAC_ACTIVE;

CmacConfigPar->NumOfLoch[0] = 1;
CmacConfigPar->LochInfo[0][0].Loch       = U_CCCH;
CmacConfigPar->LochInfo[0][0].LochNo     = 07
CmacConfigPar->LochInfo[0][0].Priority   = 0;
CmacConfigPar->LochInfo[0][0].CTLength   = 0;
CmacConfigPar->LochInfo[0][0].CTValue    = 0;
CmacConfigPar->LochInfo[0][0].TCTFLength = 2;
CmacConfigPar->LochInfo[0][0].UEIDType   = UEID_NOTUSE;
CmacConfigPar->LochInfo[0][0].UEID       = 0;



CrlcConfigPar = &CrlcConfig_CCCH;
memset( CrlcConfigPar, 0, sizeof(CRLC_CONFIG_PAR) );

CrlcConfigPar->NumOfPUs          = 1;
CrlcConfigPar->PU_LengthTM       = 0;   // It will be ignored in this scenario. (TMD-PDU)
                                        // Refer to "Easy to… A.1.10 CrlcConfig()"
CrlcConfigPar->PU_LengthUM       = 152;
CrlcConfigPar->PU_LengthAM       = 144;
```

PCCH, CCCH, DTCH–PCH, FACH#0, FACH#1–S–CCPCH



PCCH, CCCH, DTCH–PCH, FACH#0, FACH#1–S–CCPCH



PCCH, CCCH, DTCH–PCH, FACH#0, FACH#1–S–CCPCH

TS25.321 – Table 9.2.1.2: Coding of the Target Channel Type Field on FACH for FDD

TCTF	Designation
00	BCCH
01000000	CCCH
01000001–01001111	Reserved
01010000	MCCH
01010001–01011110	Reserved
01011111	MSCH
0110	MTCH
0111	Reserved
10000000	CTCH
10000001–10111111	Reserved
11	DCCH or DTCH over FACH

TS25.321 – Table 9.2.1.5a: Structure of the C/T field

C/T field	Designation
0000	Logical channel 1
0001	Logical channel 2
...	...
1110	Logical channel 15
1111	Reserved

TS25.3T21 – Table 9.2.1.6: Lengths of UE Id field

UE Id type	Length of UE Id field
U–RNTI	32 bits
C–RNTI	16 bits

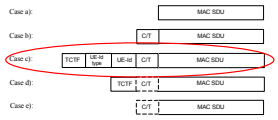
TS25.321 – Table 9.2.1.7: UE–Id Type field definition

UE–Id Type field 2 bits	UE–Id Type
00	U–RNTI
01	C–RNTI
10	Reserved
11	Reserved

```
CmacConfigPar = &CmacConfig_S_CCPCH_PCHxFACH;
CmacConfigPar->ActFlag    = MAC_ACTIVE;

CmacConfigPar->NumOfLoch[0] = 1;
CmacConfigPar->LochInfo[0][0].Loch        = D_PCCH;
CmacConfigPar->LochInfo[0][0].LochNo      = 07
CmacConfigPar->LochInfo[0][0].Priority    = 0;
CmacConfigPar->LochInfo[0][0].CTLength    = 0;
CmacConfigPar->LochInfo[0][0].TCTFLength  = 0;
CmacConfigPar->LochInfo[0][0].UEIDType    = UEID_NOTUSE;
CmacConfigPar->LochInfo[0][0].UEID        = 0;

CmacConfigPar->NumOfLoch[1] = 1;
CmacConfigPar->LochInfo[1][0].Loch        = D_CCCH;
CmacConfigPar->LochInfo[1][0].LochNo      = 07
CmacConfigPar->LochInfo[1][0].Priority    = 0;
CmacConfigPar->LochInfo[1][0].CTLength    = 0;
CmacConfigPar->LochInfo[1][0].CTValue     = 0;
CmacConfigPar->LochInfo[1][0].TCTFLength  = 8;
CmacConfigPar->LochInfo[1][0].UEIDType    = UEID_NOTUSE;
CmacConfigPar->LochInfo[1][0].UEID        = 0;

CmacConfigPar->NumOfLoch [2]= 1;
CmacConfigPar->LochInfo[2][0].Loch        = D_DTCH;
CmacConfigPar->LochInfo[2][0].LochNo      = 77
CmacConfigPar->LochInfo[2][0].Priority    = 0;
CmacConfigPar->LochInfo[2][0].CTLength    = 4;
CmacConfigPar->LochInfo[2][0].CTValue     = 0;
CmacConfigPar->LochInfo[2][0].TCTFLength  = 2;
CmacConfigPar->LochInfo[2][0].UEIDType    = UEID_C_RNTI;
CmacConfigPar->LochInfo[2][0].UEID        = 0;
```

```
CrlcConfigPar = &CrlcConfig_CCCH;
memset( CrlcConfigPar, 0, sizeof(CRLC_CONFIG_PAR) );

CrlcConfigPar->NumOfPUs             = 1;
CrlcConfigPar->PU_LengthTM          = 0;
CrlcConfigPar->PU_LengthUM          = 152; // 152+RLC Seq.No.(8)+Mac TCTF(8)
CrlcConfigPar->PU_LengthAM          = 144;
```

```
CrlcConfigPar = &CrlcConfig_PCCH;
memset( CrlcConfigPar, 0, sizeof(CRLC_CONFIG_PAR) );

CrlcConfigPar->NumOfPUs             = 1;
CrlcConfigPar->PU_LengthTM          = 0;
CrlcConfigPar->PU_LengthUM          = 232;
CrlcConfigPar->PU_LengthAM          = 224;
```



TS25.321 9.2.1.1 MAC header for DTCH and DCCH (not mapped on HS–DSCH or E–DCH)

c) DTCH or DCCH mapped to RACH/FACH:

–TCTF field, C/T field, UE-Id type field and UE-Id are included in the MAC header. For FACH, the UE-Id type field is C–RNTI or U–RNTI. For RACH, the UE-Id type field is C–RNTI.

TS34.108 6.10.2.4.1.3 Stand-alone UL: 13.6 DL: 13.6 kbps SRBs for DCCH

Stand-alone Uplink DPCH–Layer 1
Stand-alone Uplink DPCH–MAC, RLC
Stand-alone Downlink DPCH–Layer 1
Stand-alone Downlink DPCH–MAC, RLC

Stand-alone Uplink DPCH–Layer 1 (TS34.108 6.10.2.4.1.3 Stand-alone UL: 13.6 DL: 13.6 kbps SRBs for DCCH)

DPCH Uplink	Min spreading factor	64
	Max number of DPDCH data bits/radio frame	600
	Puncturing Limit	1

TFCS size	2
TFCS	SRBs for DCCH = TF0, TF1

			SRB#1	SRB#2	SRB#3	SRB#4
Higher layer	RAB/signalling RB					
	User of Radio Bearer		RRC	RRC	NAS_DT High prio	NAS_DT Low prio
RLC	Logical channel type		DCCH	DCCH	DCCH	DCCH
	RLC mode		UM	AM	AM	AM
	Payload sizes, bit		136	128	128	128
	Max data rate, bps		13 600	12 800	12 800	12 800
	AMD/UMD PDU header, bit		8	16	16	16
MAC	MAC header, bit		4	4	4	4
	MAC multiplexing		4 logical channel multiplexing			
Layer 1	TrCH type		DCH			
	TB sizes, bit		148 (alt 0, 148)			
	TFS	TF0, bits	0x148 (alt 1x0)			
		TF1, bits	1x148			
	TTI, ms		10			
	Coding type		CC 1/3			
	CRC, bit		16			
	Max number of bits/TTI before rate matching		516			
	Uplink: Max number of bits/radio frame before rate matching		516			

```
CphyRlSetupPar = &CphyRlSetup_U_DPCH_SDCCH;
memset( CphyRlSetupPar, 0, sizeof(CPHY_RL_SETUP_PAR) );
CphyRlSetupPar->Offset          = 8704;
CphyRlSetupPar->ScrCode         = 0x01000000;

CphyRlSetupPar->SymbolRate      = SYMRATE60K;   /* for DPDCH */
CphyRlSetupPar->SlotFormat      = SLOT_FORMAT_0;/* for DPCH */

CphyRlSetupPar->ChCode          = 0;
CphyRlSetupPar->NumOfDPDCH      = 1;
CphyRlSetupPar->Dpdch[0].ChCode = 16;
CphyRlSetupPar->TxDiversity     = DIVERSITY_OFF;
```

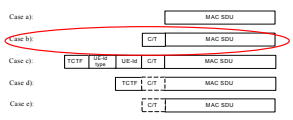
```
CphyTrchConfigPar = &CphyTrchConfig_U_DPCH_SDCCH;
memset( CphyTrchConfigPar, 0, sizeof(CPHY_TRCH_CONFIG_PAR) );
CphyTrchConfigPar->InterLv2nd  = INTERLEAVE_ON;

CphyTrchConfigPar->PuncLimit   = 100;

CphyTrchConfigPar->TFCS.NumOfTFC  = 2;
CphyTrchConfigPar->TFCS.TFC[0][0] = 0;
CphyTrchConfigPar->TFCS.TFC[1][0] = 1;

CphyTrchConfigPar->NumOfTrch   = 1;
CphyTrchConfigPar->TrchInfo[0].Trch              = U_DCH;
CphyTrchConfigPar->TrchInfo[0].TrchNo            = 0?
CphyTrchConfigPar->TrchInfo[0].InterLvlst        = INTERLEAVE_ON;

CphyTrchConfigPar->TrchInfo[0].NumOfDynamic      = 2;
CphyTrchConfigPar->TrchInfo[0].Dynamic[0].NumOfTB = 0;
CphyTrchConfigPar->TrchInfo[0].Dynamic[0].TBSize  = 148;
CphyTrchConfigPar->TrchInfo[0].Dynamic[1].NumOfTB = 1;
CphyTrchConfigPar->TrchInfo[0].Dynamic[1].TBSize  = 148;

CphyTrchConfigPar->TrchInfo[0].Static.TTI        = 1;
CphyTrchConfigPar->TrchInfo[0].Static.EPType     = CODING_CONV;
CphyTrchConfigPar->TrchInfo[0].Static.CodingRate = CODINGRATE1_3;
CphyTrchConfigPar->TrchInfo[0].Static.CRC_Size   = 16;
CphyTrchConfigPar->TrchInfo[0].Static.RM_Attr    = 160;
```


Stand-alone Uplink DPCH–MAC, RLC (TS34.108 6.10.2.4.1.3 Stand-alone UL: 13.6 DL: 13.6 kbps SRBs for DCCH)

			SRB#1	SRB#2	SRB#3	SRB#4
Higher layer	RAB/signalling RB					
	User of Radio Bearer		RRC	RRC	NAS_DT High prio	NAS_DT Low prio
RLC	Logical channel type		DCCH	DCCH	DCCH	DCCH
	RLC mode		UM	AM	AM	AM
	Payload sizes, bit		136	128	128	128
	Max data rate, bps		13 600	12 800	12 800	12 800
	AMD/UMD PDU header, bit		8	16	16	16
MAC	MAC header, bit		4	4	4	4
	MAC multiplexing		4 logical channel multiplexing			
Layer 1	TrCH type		DCH			
	TB sizes, bit		148 (alt 0, 148)			
	TFS	TF0, bits	0x148 (alt 1x0)			
		TF1, bits	1x148			
	TTI, ms		10			
	Coding type		CC 1/3			
	CRC, bit		16			
	Max number of bits/TTI before rate matching		516			
	Uplink: Max number of bits/radio frame before rate matching		516			

```
CrlcConfigPar = &CrlcConfig_DCCH;
memset( CrlcConfigPar, 0, sizeof(CRLC_CONFIG_PAR) );
CrlcConfigPar->NumOfPDUs        = 1;
CrlcConfigPar->PU_LengthTM      = 0;

CrlcConfigPar->PU_LengthUM      = 136;
CrlcConfigPar->PU_LengthAM      = 128;
```

```
CmacConfigPar = &CmacConfig_U_DPCH_SDCCH;
memset( CmacConfigPar, 0, sizeof(CMAC_CONFIG_PAR) );
CmacConfigPar->ActFlag   = MAC_ACTIVE?

CmacConfigPar->NumOfLoch[0] = 4;

CmacConfigPar->LochInfo[0][0].Loch       = U_DCCH;
CmacConfigPar->LochInfo[0][0].LochNo     = 0?
CmacConfigPar->LochInfo[0][0].Priority   = 0;
CmacConfigPar->LochInfo[0][0].CTLength   = 4;
CmacConfigPar->LochInfo[0][0].CTValue    = 0;
CmacConfigPar->LochInfo[0][0].TCTFLength = 0;
CmacConfigPar->LochInfo[0][0].UEIDType   = UEID_NOTUSE;
CmacConfigPar->LochInfo[0][0].UEID       = 0;

CmacConfigPar->LochInfo[0][1].Loch       = U_DCCH;
CmacConfigPar->LochInfo[0][1].LochNo     = 1?
CmacConfigPar->LochInfo[0][1].Priority   = 0;
CmacConfigPar->LochInfo[0][1].CTLength   = 4;
CmacConfigPar->LochInfo[0][1].CTValue    = 1;
CmacConfigPar->LochInfo[0][1].TCTFLength = 0;
CmacConfigPar->LochInfo[0][1].UEIDType   = UEID_NOTUSE;
CmacConfigPar->LochInfo[0][1].UEID       = 0;

CmacConfigPar->LochInfo[0][2].Loch       = U_DCCH;
CmacConfigPar->LochInfo[0][2].LochNo     = 2?
CmacConfigPar->LochInfo[0][2].Priority   = 0;
CmacConfigPar->LochInfo[0][2].CTLength   = 4;
CmacConfigPar->LochInfo[0][2].CTValue    = 2;
CmacConfigPar->LochInfo[0][2].TCTFLength = 0;
CmacConfigPar->LochInfo[0][2].UEIDType   = UEID_NOTUSE;
CmacConfigPar->LochInfo[0][2].UEID       = 0;

CmacConfigPar->LochInfo[0][3].Loch       = U_DCCH;
CmacConfigPar->LochInfo[0][3].LochNo     = 3?
CmacConfigPar->LochInfo[0][3].Priority   = 0;
CmacConfigPar->LochInfo[0][3].CTLength   = 4;
CmacConfigPar->LochInfo[0][3].CTValue    = 3;
CmacConfigPar->LochInfo[0][3].TCTFLength = 0;
CmacConfigPar->LochInfo[0][3].UEIDType   = UEID_NOTUSE;
CmacConfigPar->LochInfo[0][3].UEID       = 0;
```



TS25.321 – Table 9.2.1.5a: Structure of the C/T field

C/T field	Designation
0000	Logical channel 1
0001	Logical channel 2
...	...
1110	Logical channel 15
1111	Reserved

TS25.321 9.2.1.1 MAC header for DTCH and DCCH (not mapped on HS–DSCH or E–DCH)

b) DTCH or DCCH mapped to DCH, with multiplexing of dedicated channels on MAC:

–C/T field is included in MAC header.

Stand-alone Downlink DPCH–Layer 1 (TS34.108 6.10.2.4.1.3 Stand-alone UL: 13.6 DL: 13.6 kbps SRBs for DCCH)

DPCH Downlink	DTX position		Fixed
	Spreading factor		128
	DPCCH	Number of TFCI bits/slot	0
		Number of TPC bits/slot	2
		Number of Pilot bits/slot	4
	DPDCH	Number of data bits/slot	34
		Number of data bits/frame	510

TFCS size	2
TFCS	SRBs for DCCH = TF0, TF1

Higher layer	RAB/signalling RB		SRB#1	SRB#2	SRB#3	SRB#4
	User of Radio Bearer		RRC	RRC	NAS_DT High prio	NAS_DT Low prio
RLC	Logical channel type		DCCH	DCCH	DCCH	DCCH
	RLC mode		UM	AM	AM	AM
	Payload sizes, bit		136	128	128	128
	Max data rate, bps		13 600	12 800	12 800	12 800
	AMD/UMD PDU header, bit		8	16	16	16
MAC	MAC header, bit		4	4	4	4
	MAC multiplexing		4 logical channel multiplexing			
Layer 1	TrCH type		DCH			
	TB sizes, bit		148 (alt 0, 148) (note)			
	TFS	TF0, bits	0x148 (alt 1x0) (note)			
		TF1, bits	1x148			
	TTI, ms		10			
	Coding type		CC 1/3			
	CRC, bit		16			
	Max number of bits/TTI before rate matching		516			
NOTE:	alternative parameters enable the measurement "transport channel BLER" in the UE.					

```
CphyRlSetupPar = &CphyRlSetup_D_DPCH_SDCCH;
memset( CphyRlSetupPar, 0, sizeof(CPHY_RL_SETUP_PAR) );

CphyRlSetupPar->Offset          = 7680;
CphyRlSetupPar->ScrCode         = 0x00000090;

CphyRlSetupPar->SlotFormat      = SLOT_FORMAT_8;

CphyRlSetupPar->SymbolRate      = SYMRATE30K;

CphyRlSetupPar->ChCode          = 30;
CphyRlSetupPar->Power           = POWER_STEP_01DB(-160);
CphyRlSetupPar->NumOfDPDCH      = 1;
CphyRlSetupPar->Dpdch[0].Power  = POWER_STEP_01DB(-160);
CphyRlSetupPar->Dpdch[0].ChCode = 30;
CphyRlSetupPar->TxDiversity     = DIVERSITY_OFF;
CphyRlSetupPar->MaxDLPower      = -10;
CphyRlSetupPar->MinDLPower      = -99;
```

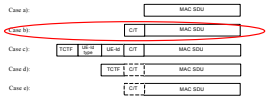
```
CphyTrchConfigPar = &CphyTrchConfig_D_DPCH_SDCCH;
memset( CphyTrchConfigPar, 0, sizeof(CPHY_TRCH_CONFIG_PAR) );

CphyTrchConfigPar->DTXPosition = DTX_FIXED_POSITION;

CphyTrchConfigPar->InterLv2nd  = INTERLEAVE_ON;
CphyTrchConfigPar->NumOfTrch   = 1;

CphyTrchConfigPar->TFCS.NumOfTFC  = 2;
CphyTrchConfigPar->TFCS.TFC[0][0] = 0;
CphyTrchConfigPar->TFCS.TFC[1][0] = 1;

CphyTrchConfigPar->TrchInfo[0].Trch              = D_DCH;
CphyTrchConfigPar->TrchInfo[0].TrchNo            = 0;
CphyTrchConfigPar->TrchInfo[0].InterLvlst        = INTERLEAVE_ON;

CphyTrchConfigPar->TrchInfo[0].NumOfDynamic      = 2;
CphyTrchConfigPar->TrchInfo[0].Dynamic[0].NumOfTB = 0;
CphyTrchConfigPar->TrchInfo[0].Dynamic[0].TBSize  = 148;
CphyTrchConfigPar->TrchInfo[0].Dynamic[1].NumOfTB = 1;
CphyTrchConfigPar->TrchInfo[0].Dynamic[1].TBSize  = 148;

CphyTrchConfigPar->TrchInfo[0].Static.TTI        = 1;
CphyTrchConfigPar->TrchInfo[0].Static.EPType     = CODING_CONV;
CphyTrchConfigPar->TrchInfo[0].Static.CodingRate = CODINGRATE1_3;
CphyTrchConfigPar->TrchInfo[0].Static.CRC_Size   = 16;
CphyTrchConfigPar->TrchInfo[0].Static.RM_Attr    = 160;
```


Stand-alone Downlink DPCH–MAC, RLC (TS34.108 6.10.2.4.1.3 Stand-alone UL: 13.6 DL: 13.6 kbps SRBs for DCCH)

Higher layer	RAB/signalling RB		SRB#1	SRB#2	SRB#3	SRB#4
	User of Radio Bearer		RRC	RRC	NAS_DT High prio	NAS_DT Low prio
RLC	Logical channel type		DCCH	DCCH	DCCH	DCCH
	RLC mode		UM	AM	AM	AM
	Payload sizes, bit		136	128	128	128
	Max data rate, bps		13 600	12 800	12 800	12 800
	AMD/UMD PDU header, bit		8	16	16	16
MAC	MAC header, bit		4	4	4	4
	MAC multiplexing		4 logical channel multiplexing			
Layer 1	TrCH type		DCH			
	TB sizes, bit		148 (alt 0, 148) (note)			
	TFS	TF0, bits	0x148 (alt 1x0) (note)			
		TF1, bits	1x148			
	TTI, ms		10			
	Coding type		CC 1/3			
	CRC, bit		16			
	Max number of bits/TTI before rate matching		516			
NOTE:	alternative parameters enable the measurement "transport channel BLER" in the UE.					

```
CrlcConfigPar = &CrlcConfig_DCCH;
memset( CrlcConfigPar, 0, sizeof(CRLC_CONFIG_PAR) );
CrlcConfigPar->NumOfPUs         = 1;

CrlcConfigPar->PU_LengthTM      = 0;
CrlcConfigPar->PU_LengthUM      = 136;
CrlcConfigPar->PU_LengthAM      = 128;
```

```
CmacConfigPar = &CmacConfig_D_DPCH_SDCCH;
memset( CmacConfigPar, 0, sizeof(CMAC_CONFIG_PAR) );
CmacConfigPar->ActFlag   = MAC_ACTIVE;

CmacConfigPar->NumOfLoch[0] = 4;

CmacConfigPar->LochInfo[0][0].Loch      = D_DCCH;
CmacConfigPar->LochInfo[0][0].LochNo    = 0;
CmacConfigPar->LochInfo[0][0].Priority  = 0;
CmacConfigPar->LochInfo[0][0].CTLength  = 4;
CmacConfigPar->LochInfo[0][0].CTValue   = 0;
CmacConfigPar->LochInfo[0][0].TCTFLength = 0;
CmacConfigPar->LochInfo[0][0].UEIDType  = UEID_NOTUSE;
CmacConfigPar->LochInfo[0][0].UEID      = 0;

CmacConfigPar->LochInfo[0][1].Loch      = D_DCCH;
CmacConfigPar->LochInfo[0][1].LochNo    = 17;
CmacConfigPar->LochInfo[0][1].Priority  = 0;
CmacConfigPar->LochInfo[0][1].CTLength  = 4;
CmacConfigPar->LochInfo[0][1].CTValue   = 1;
CmacConfigPar->LochInfo[0][1].TCTFLength = 0;
CmacConfigPar->LochInfo[0][1].UEIDType  = UEID_NOTUSE;
CmacConfigPar->LochInfo[0][1].UEID      = 0;

CmacConfigPar->LochInfo[0][2].Loch      = D_DCCH;
CmacConfigPar->LochInfo[0][2].LochNo    = 27;
CmacConfigPar->LochInfo[0][2].Priority  = 0;
CmacConfigPar->LochInfo[0][2].CTLength  = 4;
CmacConfigPar->LochInfo[0][2].CTValue   = 2;
CmacConfigPar->LochInfo[0][2].TCTFLength = 0;
CmacConfigPar->LochInfo[0][2].UEIDType  = UEID_NOTUSE;
CmacConfigPar->LochInfo[0][2].UEID      = 0;

CmacConfigPar->LochInfo[0][3].Loch      = D_DCCH;
CmacConfigPar->LochInfo[0][3].LochNo    = 37;
CmacConfigPar->LochInfo[0][3].Priority  = 0;
CmacConfigPar->LochInfo[0][3].CTLength  = 4;
CmacConfigPar->LochInfo[0][3].CTValue   = 3;
CmacConfigPar->LochInfo[0][3].TCTFLength = 0;
CmacConfigPar->LochInfo[0][3].UEIDType  = UEID_NOTUSE;
CmacConfigPar->LochInfo[0][3].UEID      = 0;
```



TS25.321 – Table 9.2.1.5a: Structure of the C/T field

C/T field	Designation
0000	Logical channel 1
0001	Logical channel 2
...	...
1110	Logical channel 15
1111	Reserved

TS25.321 9.2.1.1 MAC header for DTCH and DCCH (not mapped on HS–DSCH or E–DCH)

b) DTCH or DCCH mapped to DCH, with multiplexing of dedicated channels on MAC:

–C/T field is included in MAC header.

TS34.108 6.10.2.4.1.32 Interactive or background/UL: 64 DL: 384 kbps/PS RAB + UL: 3.4 DL: 3.4 kbps SRBs for DCCH

Uplink DPCH–Layer1
Uplink DPCH–MAC, RLC
Downlink DPCH–Layer1
Downlink DPCH–MAC, RLC

Uplink DPCH–Layer 1 (TS34.108 6.10.2.4.1.32 Interactive or background/UL: 64 DL: 384 kbps/PS RAB + UL: 3.4 DL: 3.4 kbps SRBs for DCCH)

DPCH Uplink	Min spreading factor	16
	Max number of DPDCH data bits/radio frame	2 400
	Puncturing Limit	0.96

TFCS size	10
TFCS	(64 kbps RAB, DCCH)= (TF0, TF0), (TF1, TF0), (TF2, TF0), (TF3, TF0), (TF4, TF0), (TF0, TF1), (TF1, TF1), (TF2, TF1), (TF3, TF1), (TF4, TF1)

Higher layer	RAB/Signalling RB		RAB
RLC	Logical channel type		DTCH
	RLC mode		AM
	Payload sizes, bit		320
	Max data rate, bps		64 000
	AMD PDU header, bit		16
MAC	MAC header, bit		0
	MAC multiplexing		N/A
Layer 1	TrCH type		DCH
	TB sizes, bit		336
	TFS	TF0, bits	0x336
		TF1, bits	1x336
		TF2, bits	2x336
		TF3, bits	3x336
		TF4, bits	4x336
	TTI, ms		20
	Coding type		TC
	CRC, bit		16
	Max number of bits/TTI after channel coding		4 236
	Uplink: Max number of bits/radio frame before rate matching		2 118
	RM attribute		130 to 170

DCH#1 is the same as stand-alone configuration for DCCH except TTI.

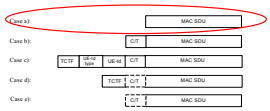
```
CphyRlSetupPar = &CphyRlSetup_U_DPCH_P64K;
memset( CphyRlSetupPar, 0, sizeof(CPHY_RL_SETUP_PAR) );
CphyRlSetupPar->OffSet          = 87047
CphyRlSetupPar->ScrCode         = 0x01000000;

CphyRlSetupPar->SymbolRate      = SYMRATE240K;  /* for DPDCH */

CphyRlSetupPar->SlotFormat      = SLOT_FORMAT_0;/* for DPCCH */

CphyRlSetupPar->ChCode          = 0; // TS25.213 4.3.1.2.1 Always 0
CphyRlSetupPar->NumOfDPDCH      = 1;
CphyRlSetupPar->DpdchPar[0].ChCode = 4; // TS25.213 4.3.1.2.1 SF=16/4
CphyRlSetupPar->TxDiversity     = DIVERSITY_OFF;


CphyTrchConfigPar = &CphyTrchConfig_U_DPCH_P64K;
memset( CphyTrchConfigPar, 0, sizeof(CPHY_TRCH_CONFIG_PAR) );
CphyTrchConfigPar->InterLv2nd   = INTERLEAVE_ON;
CphyTrchConfigPar->PuncLimit    = 96;

CphyTrchConfigPar->TFCS.NumOfTFC  = 10;
CphyTrchConfigPar->TFCS.TFC[0][0] = 0;
CphyTrchConfigPar->TFCS.TFC[0][1] = 0;
CphyTrchConfigPar->TFCS.TFC[1][0] = 1;
CphyTrchConfigPar->TFCS.TFC[1][1] = 0;
CphyTrchConfigPar->TFCS.TFC[2][0] = 2;
CphyTrchConfigPar->TFCS.TFC[2][1] = 0;
CphyTrchConfigPar->TFCS.TFC[3][0] = 3;
CphyTrchConfigPar->TFCS.TFC[3][1] = 0;
CphyTrchConfigPar->TFCS.TFC[4][0] = 4;
CphyTrchConfigPar->TFCS.TFC[4][1] = 0;
CphyTrchConfigPar->TFCS.TFC[5][0] = 0;
CphyTrchConfigPar->TFCS.TFC[5][1] = 1;
CphyTrchConfigPar->TFCS.TFC[6][0] = 1;
CphyTrchConfigPar->TFCS.TFC[6][1] = 1;
CphyTrchConfigPar->TFCS.TFC[7][0] = 2;
CphyTrchConfigPar->TFCS.TFC[7][1] = 1;
CphyTrchConfigPar->TFCS.TFC[8][0] = 3;
CphyTrchConfigPar->TFCS.TFC[8][1] = 1;
CphyTrchConfigPar->TFCS.TFC[9][0] = 4;
CphyTrchConfigPar->TFCS.TFC[9][1] = 1;

CphyTrchConfigPar->NumOfTrch    = 2;
CphyTrchConfigPar->TrchInfo[0].Trch             = U_DCH;
CphyTrchConfigPar->TrchInfo[0].TrchNo           = 07
CphyTrchConfigPar->TrchInfo[0].InterLvlst       = INTERLEAVE_ON;

CphyTrchConfigPar->TrchInfo[0].NumOfDynamic     = 5;
CphyTrchConfigPar->TrchInfo[0].Dynamic[0].NumOfTB = 0;
CphyTrchConfigPar->TrchInfo[0].Dynamic[0].TBSize  = 336;
CphyTrchConfigPar->TrchInfo[0].Dynamic[1].NumOfTB = 1;
CphyTrchConfigPar->TrchInfo[0].Dynamic[1].TBSize  = 336;
CphyTrchConfigPar->TrchInfo[0].Dynamic[2].NumOfTB = 2;
CphyTrchConfigPar->TrchInfo[0].Dynamic[2].TBSize  = 336;
CphyTrchConfigPar->TrchInfo[0].Dynamic[3].NumOfTB = 3;
CphyTrchConfigPar->TrchInfo[0].Dynamic[3].TBSize  = 336;
CphyTrchConfigPar->TrchInfo[0].Dynamic[4].NumOfTB = 4;
CphyTrchConfigPar->TrchInfo[0].Dynamic[4].TBSize  = 336;

CphyTrchConfigPar->TrchInfo[0].Static.TTI       = 2;
CphyTrchConfigPar->TrchInfo[0].Static.RPType    = CODING_TURBO;
CphyTrchConfigPar->TrchInfo[0].Static.CodingRate  = CODINGRATE1_3;
CphyTrchConfigPar->TrchInfo[0].Static.CRC_Size    = 16;
CphyTrchConfigPar->TrchInfo[0].Static.RM_Attr     = 143;

CphyTrchConfigPar->TrchInfo[1].Trch             = U_DCH;
CphyTrchConfigPar->TrchInfo[1].TrchNo           = 17
CphyTrchConfigPar->TrchInfo[1].InterLvlst       = INTERLEAVE_ON;

CphyTrchConfigPar->TrchInfo[1].NumOfDynamic     = 2;
CphyTrchConfigPar->TrchInfo[1].Dynamic[0].NumOfTB = 0;
CphyTrchConfigPar->TrchInfo[1].Dynamic[0].TBSize  = 148;
CphyTrchConfigPar->TrchInfo[1].Dynamic[1].NumOfTB = 1;
CphyTrchConfigPar->TrchInfo[1].Dynamic[1].TBSize  = 148;

CphyTrchConfigPar->TrchInfo[1].Static.TTI       = 4;
CphyTrchConfigPar->TrchInfo[1].Static.RPType    = CODING_CONV;
CphyTrchConfigPar->TrchInfo[1].Static.CodingRate  = CODINGRATE1_3;
CphyTrchConfigPar->TrchInfo[1].Static.CRC_Size    = 16;
CphyTrchConfigPar->TrchInfo[1].Static.RM_Attr     = 160;
```

Uplink DPCH–MAC, RLC (TS34.108 6.10.2.4.1.32 Interactive or background/UL: 64 DL: 384 kbps/PS RAB + UL: 3.4 DL: 3.4 kbps SRBs for DCCH)

Higher layer	RAB/Signalling RB		RAB
RLC	Logical channel type		DTCH
	RLC mode		AM
	Payload sizes, bit		320
	Max data rate, bps		64 000
	AMD PDU header, bit		16
MAC	MAC header, bit		0
	MAC multiplexing		N/A
Layer 1	TrCH type		DCH
	TB sizes, bit		336
	TFS	TF0, bits	0x336
		TF1, bits	1x336
		TF2, bits	2x336
		TF3, bits	3x336
		TF4, bits	4x336
	TTI, ms		20
	Coding type		TC
	CRC, bit		16
	Max number of bits/TTI after channel coding		4 236
	Uplink: Max number of bits/radio frame before rate matching		2 118
	RM attribute		130 to 170

```
CrlcConfigPar = &Packet_CrlcConfig_DTCH;
memset( CrlcConfigPar, 0, sizeof(CRLC_CONFIG_PAR) );
CrlcConfigPar->NumOfPUs          = 1;
CrlcConfigPar->PU_LengthTM       = 244;

CrlcConfigPar->PU_LengthUM       = 328;

CrlcConfigPar->PU_LengthAM       = 320;
```

```
CmacConfigPar = &CmacConfig_U_DPCH_P64K;
memset( CmacConfigPar, 0, sizeof(CMAC_CONFIG_PAR) );
CmacConfigPar->ActFlag   = MAC_ACTIVE;

CmacConfigPar->NumOfLoch[0] = 1;
CmacConfigPar->LochInfo[0][0].Loch       = U_DTCH;
CmacConfigPar->LochInfo[0][0].LochNo     = 07
CmacConfigPar->LochInfo[0][0].Priority   = 0;
CmacConfigPar->LochInfo[0][0].CTLength   = 0;
CmacConfigPar->LochInfo[0][0].CTValue    = 0;
CmacConfigPar->LochInfo[0][0].TCTFLength = 0;
CmacConfigPar->LochInfo[0][0].UEIDType   = UEID_NOTUSE;
CmacConfigPar->LochInfo[0][0].UEID       = 0;


CmacConfigPar->NumOfLoch[1] = 4;
CmacConfigPar->LochInfo[1][0].Loch       = U_DCCH;
CmacConfigPar->LochInfo[1][0].LochNo     = 07
CmacConfigPar->LochInfo[1][0].Priority   = 0;
CmacConfigPar->LochInfo[1][0].CTLength   = 4;
CmacConfigPar->LochInfo[1][0].CTValue    = 0;
CmacConfigPar->LochInfo[1][0].TCTFLength = 0;
CmacConfigPar->LochInfo[1][0].UEIDType   = UEID_NOTUSE;
CmacConfigPar->LochInfo[1][0].UEID       = 0;

CmacConfigPar->LochInfo[1][1].Loch       = U_DCCH;
CmacConfigPar->LochInfo[1][1].LochNo     = 17
CmacConfigPar->LochInfo[1][1].Priority   = 0;
CmacConfigPar->LochInfo[1][1].CTLength   = 4;
CmacConfigPar->LochInfo[1][1].CTValue    = 1;
CmacConfigPar->LochInfo[1][1].TCTFLength = 0;
CmacConfigPar->LochInfo[1][1].UEIDType   = UEID_NOTUSE;
CmacConfigPar->LochInfo[1][1].UEID       = 0;

CmacConfigPar->LochInfo[1][2].Loch       = U_DCCH;
CmacConfigPar->LochInfo[1][2].LochNo     = 27
CmacConfigPar->LochInfo[1][2].Priority   = 0;
CmacConfigPar->LochInfo[1][2].CTLength   = 4;
CmacConfigPar->LochInfo[1][2].CTValue    = 2;
CmacConfigPar->LochInfo[1][2].TCTFLength = 0;
CmacConfigPar->LochInfo[1][2].UEIDType   = UEID_NOTUSE;
CmacConfigPar->LochInfo[1][2].UEID       = 0;

CmacConfigPar->LochInfo[1][3].Loch       = U_DCCH;
CmacConfigPar->LochInfo[1][3].LochNo     = 37
CmacConfigPar->LochInfo[1][3].Priority   = 0;
CmacConfigPar->LochInfo[1][3].CTLength   = 4;
CmacConfigPar->LochInfo[1][3].CTValue    = 3;
CmacConfigPar->LochInfo[1][3].TCTFLength = 0;
CmacConfigPar->LochInfo[1][3].UEIDType   = UEID_NOTUSE;
CmacConfigPar->LochInfo[1][3].UEID       = 0;
```



TS25.321 9.2.1.1 MAC header for DTCH and DCCH (not mapped on HS–DSCH or E–DCH)

a) DTCH or DCCH mapped to DCH, no multiplexing of dedicated channels on MAC:

–no MAC header required

Downlink DPCH–Layer 1 (TS34.108 6.10.2.4.1.32 Interactive or background/UL: 64 DL: 384 kbps/PS RAB + UL: 3.4 DL: 3.4 kbps SRBs for DCCH)

DPCH Downlink	DTX position		Flexible
	Spreading factor		8
	Number of DPDCH		1
	DPCCH	Number of TFCI bits/slot	8
		Number of TPC bits/slot	8
		Number of Pilot bits/slot	16
	DPDCH	Number of data bits/slot	608
		Number of data bits/frame	9 120

TFCS size	12 (alt.18)
TFCS	(384 kbps RAB, DCCH)= (TF0, TF0), (TF1, TF0), (TF2, TF0), (TF3, TF0), (TF4, TF0), (TF5, TF0) (TF0, TF1), (TF1, TF1), (TF2, TF1), (TF3, TF1), (TF4, TF1), (TF5, TF1) (alt. (TF0, TF0), (TF1, TF0), (TF2, TF0), (TF3, TF0), (TF4, TF0), (TF5, TF0), (TF6, TF0), (TF7, TF0), (TF8, TF0), (TF0, TF1), (TF1, TF1), (TF2, TF1), (TF3, TF1), (TF4, TF1), (TF5, TF1), (TF6, TF1), (TF7, TF1), (TF8, TF1))

Higher layer	RAB/Signalling RB		RAB
RLC	Logical channel type		DTCH
	RLC mode		AM
	Payload sizes, bit		320
	Max data rate, bps		384 000
	AMD PDU header, bit		16
MAC	MAC header, bit		0
	MAC multiplexing		N/A
Layer 1	TrCH type		DCH
	TB sizes, bit		336
	TFS	TF0, bits	0x336
		TF1, bits	1x336
		TF2, bits	2x336
		TF3, bits	4x336
		TF4, bits	8x336
		TF5, bits	12x336
		TF6, bits	N/A (alt. 16x336)
		TF7, bits	N/A (alt. 20x336)
		TF8, bits	N/A (alt. 24x336)
	TTI, ms		10 (alt. 20)
	Coding type		TC
	CRC, bit		16
	Max number of bits/TTI after channel coding		12 684 (alt. 25 368)
	RM attribute		110 to 150

DCH#1 is the same as stand-alone configuration for DCCH except TTI.

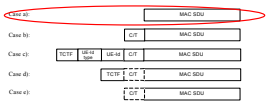
```
CphyRlSetupPar = &CphyRlSetup_D_DPCH_P384K;
memset( CphyRlSetupPar, 0, sizeof(CPHY_RL_SETUP_PAR) );
CphyRlSetupPar->dlOffset          = 76800;
CphyRlSetupPar->ScrCode           = 0x00000090;
CphyRlSetupPar->SymbolRate        = SYMBRATE480K;       /* for DPDCH */
CphyRlSetupPar->SlotFormat        = SLOT_FORMAT_15;     /* for DPDCH */
CphyRlSetupPar->CNCode            =  5;
CphyRlSetupPar->NumOfDPDCH        =  1;
CphyRlSetupPar->Dpdch[0].CNCode   =  5;
CphyRlSetupPar->Power             = POWER_STEP_01DB(-160);   /* Power = -16.0dB */
CphyRlSetupPar->Dpdch[0].Power    = POWER_STEP_01DB(-160);   /* Power = -16.0dB */
CphyRlSetupPar->TxDiversity       = DIVERSITY_OFF;
CphyRlSetupPar->MaxDLPower        = -10;
CphyRlSetupPar->MinDLPower        = -99;

CphyTrchConfigPar = &CphyTrchConfig_D_DPCH_P384K;
memset( CphyTrchConfigPar, 0, sizeof(CPHY_TRCH_CONFIG_PAR) );
CphyTrchConfigPar->DTXPosition = DTX_FLEXIBLE_POSITION;
CphyTrchConfigPar->InterLv2nd  = INTERLEAVE_ON;

CphyTrchConfigPar->TFCS.NumOfTFC  = 12;
CphyTrchConfigPar->TFCS.TFC[0][0] = 0;
CphyTrchConfigPar->TFCS.TFC[0][1] = 0;
CphyTrchConfigPar->TFCS.TFC[1][0] = 1;
CphyTrchConfigPar->TFCS.TFC[1][1] = 0;
CphyTrchConfigPar->TFCS.TFC[2][0] = 2;
CphyTrchConfigPar->TFCS.TFC[2][1] = 0;
CphyTrchConfigPar->TFCS.TFC[3][0] = 3;
CphyTrchConfigPar->TFCS.TFC[3][1] = 0;
CphyTrchConfigPar->TFCS.TFC[4][0] = 4;
CphyTrchConfigPar->TFCS.TFC[4][1] = 0;
CphyTrchConfigPar->TFCS.TFC[5][0] = 5;
CphyTrchConfigPar->TFCS.TFC[5][1] = 0;
CphyTrchConfigPar->TFCS.TFC[6][0] = 0;
CphyTrchConfigPar->TFCS.TFC[6][1] = 1;
CphyTrchConfigPar->TFCS.TFC[7][0] = 1;
CphyTrchConfigPar->TFCS.TFC[7][1] = 1;
CphyTrchConfigPar->TFCS.TFC[8][0] = 2;
CphyTrchConfigPar->TFCS.TFC[8][1] = 1;
CphyTrchConfigPar->TFCS.TFC[9][0] = 3;
CphyTrchConfigPar->TFCS.TFC[9][1] = 1;
CphyTrchConfigPar->TFCS.TFC[10][0] = 4;
CphyTrchConfigPar->TFCS.TFC[10][1] = 1;
CphyTrchConfigPar->TFCS.TFC[11][0] = 5;
CphyTrchConfigPar->TFCS.TFC[11][1] = 1;

CphyTrchConfigPar->NumOfTrch   = 2;
CphyTrchConfigPar->TrchInfo[0].Trch              = D_DCH;
CphyTrchConfigPar->TrchInfo[0].TrchNo            = 07
CphyTrchConfigPar->TrchInfo[0].InterLvlst        = INTERLEAVE_ON;

CphyTrchConfigPar->TrchInfo[0].NumOfDynamic      = 6;
CphyTrchConfigPar->TrchInfo[0].Dynamic[0].NumOfTB = 0;
CphyTrchConfigPar->TrchInfo[0].Dynamic[0].TBSize  = 336;
CphyTrchConfigPar->TrchInfo[0].Dynamic[1].NumOfTB = 1;
CphyTrchConfigPar->TrchInfo[0].Dynamic[1].TBSize  = 336;
CphyTrchConfigPar->TrchInfo[0].Dynamic[2].NumOfTB = 2;
CphyTrchConfigPar->TrchInfo[0].Dynamic[2].TBSize  = 336;
CphyTrchConfigPar->TrchInfo[0].Dynamic[3].NumOfTB = 4;
CphyTrchConfigPar->TrchInfo[0].Dynamic[3].TBSize  = 336;
CphyTrchConfigPar->TrchInfo[0].Dynamic[4].NumOfTB = 8;
CphyTrchConfigPar->TrchInfo[0].Dynamic[4].TBSize  = 336;
CphyTrchConfigPar->TrchInfo[0].Dynamic[5].NumOfTB = 12;
CphyTrchConfigPar->TrchInfo[0].Dynamic[5].TBSize  = 336;

CphyTrchConfigPar->TrchInfo[0].Static.TTI         = 1;
CphyTrchConfigPar->TrchInfo[0].Static.EPType      = CODING_TURBO;
CphyTrchConfigPar->TrchInfo[0].Static.CodingRate  = CODINGRATE1_3;
CphyTrchConfigPar->TrchInfo[0].Static.CRC_Size    = 16;
CphyTrchConfigPar->TrchInfo[0].Static.RM_Attr     = 110;

CphyTrchConfigPar->TrchInfo[1].Trch              = D_DCH;
CphyTrchConfigPar->TrchInfo[1].TrchNo            = 17
CphyTrchConfigPar->TrchInfo[1].InterLvlst        = INTERLEAVE_ON;

CphyTrchConfigPar->TrchInfo[1].NumOfDynamic      = 2;
CphyTrchConfigPar->TrchInfo[1].Dynamic[0].NumOfTB = 0;
CphyTrchConfigPar->TrchInfo[1].Dynamic[0].TBSize  = 148;
CphyTrchConfigPar->TrchInfo[1].Dynamic[1].NumOfTB = 1;
CphyTrchConfigPar->TrchInfo[1].Dynamic[1].TBSize  = 148;

CphyTrchConfigPar->TrchInfo[1].Static.TTI         = 4;
CphyTrchConfigPar->TrchInfo[1].Static.EPType      = CODING_CONV;
CphyTrchConfigPar->TrchInfo[1].Static.CodingRate  = CODINGRATE1_3;
CphyTrchConfigPar->TrchInfo[1].Static.CRC_Size    = 16;
CphyTrchConfigPar->TrchInfo[1].Static.RM_Attr     = 160;
```

Downlink DPCH–MAC, RLC (TS34.108 6.10.2.4.1.32 Interactive or background/UL: 64 DL: 384 kbps/PS RAB + UL: 3.4 DL: 3.4 kbps SRBs for DCCH)

Higher layer	RAB/Signalling RB		RAB
RLC	Logical channel type		DTCH
	RLC mode		AM
	Payload sizes, bit		320
	Max data rate, bps		384 000
	AMD PDU header, bit		16
MAC	MAC header, bit		0
	MAC multiplexing		N/A
Layer 1	TrCH type		DCH
	TB sizes, bit		336
	TFS	TF0, bits	0x336
		TF1, bits	1x336
		TF2, bits	2x336
		TF3, bits	4x336
		TF4, bits	8x336
		TF5, bits	12x336
		TF6, bits	N/A (alt. 16x336)
		TF7, bits	N/A (alt. 20x336)
		TF8, bits	N/A (alt. 24x336)
	TTI, ms		10 (alt. 20)
	Coding type		TC
	CRC, bit		16
	Max number of bits/TTI after channel coding		12 684 (alt. 25 368)
	RM attribute		110 to 150

```
CrlcConfigPar = &Packet_CrlcConfig_DTCH;
memset( CrlcConfigPar, 0, sizeof(CRLC_CONFIG_PAR) );
CrlcConfigPar->NumOfPUs          = 1;
CrlcConfigPar->PU_LengthTM       = 244;

CrlcConfigPar->PU_LengthUM       = 328;

CrlcConfigPar->PU_LengthAM       = 320;
```

```
CmacConfigPar = &CmacConfig_D_DPCH_P384K;
memset( CmacConfigPar, 0, sizeof(CMAC_CONFIG_PAR) );
CmacConfigPar->ActFlag   = MAC_ACTIVE;

CmacConfigPar->NumOfLoch[0] = 1;
CmacConfigPar->LochInfo[0][0].Loch       = D_DTCH;
CmacConfigPar->LochInfo[0][0].LochNo     = 07
CmacConfigPar->LochInfo[0][0].Priority   = 0;
CmacConfigPar->LochInfo[0][0].CTLength   = 0;
CmacConfigPar->LochInfo[0][0].CTValue    = 0;
CmacConfigPar->LochInfo[0][0].TCTFLength = 0;
CmacConfigPar->LochInfo[0][0].UEIDType   = UEID_NOTUSE;
CmacConfigPar->LochInfo[0][0].UEID       = 0;

CmacConfigPar->NumOfLoch[1] = 4;
CmacConfigPar->LochInfo[1][0].Loch       = D_DCCH;
CmacConfigPar->LochInfo[1][0].LochNo     = 07
CmacConfigPar->LochInfo[1][0].Priority   = 0;
CmacConfigPar->LochInfo[1][0].CTLength   = 4;
CmacConfigPar->LochInfo[1][0].CTValue    = 0;
CmacConfigPar->LochInfo[1][0].TCTFLength = 0;
CmacConfigPar->LochInfo[1][0].UEIDType   = UEID_NOTUSE;
CmacConfigPar->LochInfo[1][0].UEID       = 0;

CmacConfigPar->LochInfo[1][1].Loch       = D_DCCH;
CmacConfigPar->LochInfo[1][1].LochNo     = 17
CmacConfigPar->LochInfo[1][1].Priority   = 0;
CmacConfigPar->LochInfo[1][1].CTLength   = 4;
CmacConfigPar->LochInfo[1][1].CTValue    = 1;
CmacConfigPar->LochInfo[1][1].TCTFLength = 0;
CmacConfigPar->LochInfo[1][1].UEIDType   = UEID_NOTUSE;
CmacConfigPar->LochInfo[1][1].UEID       = 0;

CmacConfigPar->LochInfo[1][2].Loch       = D_DCCH;
CmacConfigPar->LochInfo[1][2].LochNo     = 27
CmacConfigPar->LochInfo[1][2].Priority   = 0;
CmacConfigPar->LochInfo[1][2].CTLength   = 4;
CmacConfigPar->LochInfo[1][2].CTValue    = 2;
CmacConfigPar->LochInfo[1][2].TCTFLength = 0;
CmacConfigPar->LochInfo[1][2].UEIDType   = UEID_NOTUSE;
CmacConfigPar->LochInfo[1][2].UEID       = 0;

CmacConfigPar->LochInfo[1][3].Loch       = D_DCCH;
CmacConfigPar->LochInfo[1][3].LochNo     = 37
CmacConfigPar->LochInfo[1][3].Priority   = 0;
CmacConfigPar->LochInfo[1][3].CTLength   = 4;
CmacConfigPar->LochInfo[1][3].CTValue    = 3;
CmacConfigPar->LochInfo[1][3].TCTFLength = 0;
CmacConfigPar->LochInfo[1][3].UEIDType   = UEID_NOTUSE;
CmacConfigPar->LochInfo[1][3].UEID       = 0;
```



TS25.321 9.2.1.1 MAC header for DTCH and DCCH (not mapped on HS–DSCH or E–DCH)

a) DTCH or DCCH mapped to DCH, no multiplexing of dedicated channels on MAC:

–no MAC header required

Anritsu

Specifications are subject to change without notice.

Anritsu Corporation
5-1-1 Onna, Atsugi-shi, Kanagawa, 243-8555 Japan
Phone: +81-46-223-1111
Fax: +81-46-296-1264

• **U.S.A.**
Anritsu Company
1155 East Collins Blvd., Suite 100, Richardson,
TX 75081, U.S.A.
Toll Free: 1-800-267-4878
Phone: +1-972-644-1777
Fax: +1-972-671-1877

• **Canada**
Anritsu Electronics Ltd.
700 Silver Seven Road, Suite 120, Kanata,
Ontario K2V 1C3, Canada
Phone: +1-613-591-2003
Fax: +1-613-591-1006

• **Brazil**
Anritsu Eletrônica Ltda.
Praca Amadeu Amaral, 27 - 1 Andar
01327-010-Paraiso-São Paulo-Brazil
Phone: +55-11-3283-2511
Fax: +55-11-3288-6940

• **Mexico**
Anritsu Company, S.A. de C.V.
Av. Ejército Nacional No. 579 Piso 9, Col. Granada
11520 México, D.F., México
Phone: +52-55-1101-2370
Fax: +52-55-5254-3147

• **U.K.**
Anritsu EMEA Ltd.
200 Capability Green, Luton, Bedfordshire, LU1 3LU, U.K.
Phone: +44-1582-433200
Fax: +44-1582-731303

• **France**
Anritsu S.A.
16/18 avenue du Québec-SILIC 720
91961 COURTABOEUF CEDEX, France
Phone: +33-1-60-92-15-50
Fax: +33-1-64-46-10-65

• **Germany**
Anritsu GmbH
Nemetschek Haus, Konrad-Zuse-Platz 1
81829 München, Germany
Phone: +49-89-442308-0
Fax: +49-89-442308-55

• **Italy**
Anritsu S.p.A.
Via Elio Vittorini 129, 00144 Roma, Italy
Phone: +39-6-509-9711
Fax: +39-6-502-2425

• **Sweden**
Anritsu AB
Borgafjordsgatan 13, 164 40 KISTA, Sweden
Phone: +46-8-534-707-00
Fax: +46-8-534-707-30

• **Finland**
Anritsu AB
Teknobulevardi 3-5, FI-01530 VANTAA, Finland
Phone: +358-20-741-8100
Fax: +358-20-741-8111

• **Denmark**
Anritsu A/S
Kirkebjerg Allé 90, DK-2605 Brøndby, Denmark
Phone: +45-72112200
Fax: +45-72112210

• **Spain**
Anritsu EMEA Ltd.
Oficina de Representación en España
Edificio Veganova
Avda de la Vega, n° 1 (edf 8, pl 1, of 8)
28108 ALCOBENDAS - Madrid, Spain
Phone: +34-914905761
Fax: +34-914905762

• **United Arab Emirates**
Anritsu EMEA Ltd.
Dubai Liaison Office
P O Box 500413 - Dubai Internet City
Al Thuraya Building, Tower 1, Suit 701, 7th Floor
Dubai, United Arab Emirates
Phone: +971-4-3670352
Fax: +971-4-3688460

• **Singapore**
Anritsu Pte. Ltd.
60 Alexandra Terrace, #02-08, The Comtech (Lobby A)
Singapore 118502
Phone: +65-6282-2400
Fax: +65-6282-2533

• **India**
Anritsu Pte. Ltd.
India Branch Office
Unit No. S-3, Second Floor, Esteem Red Cross Bhavan,
No. 26, Race Course Road, Bangalore 560 001, India
Phone: +91-80-32944707
Fax: +91-80-22356648

• **P.R. China (Hong Kong)**
Anritsu Company Ltd.
Units 4 & 5, 28th Floor, Greenfield Tower, Concordia Plaza,
No. 1 Science Museum Road, Tsim Sha Tsui East,
Kowloon, Hong Kong
Phone: +852-2301-4980
Fax: +852-2301-3545

• **P.R. China (Beijing)**
Anritsu Company Ltd.
Beijing Representative Office
Room 1515, Beijing Fortune Building,
No. 5, Dong-San-Huan Bei Road,
Chao-Yang District, Beijing 10004, P.R. China
Phone: +86-10-6590-9230
Fax: +86-10-6590-9235

• **Korea**
Anritsu Corporation, Ltd.
8F Hyunjuk Building, 832-41, Yeoksam Dong,
Kangnam-ku, Seoul, 135-080, Korea
Phone: +82-2-553-6603
Fax: +82-2-553-6604

• **Australia**
Anritsu Pty. Ltd.
Unit 21/270 Ferntree Gully Road, Notting Hill,
Victoria 3168, Australia
Phone: +61-3-9558-8177
Fax: +61-3-9558-8255

• **Taiwan**
Anritsu Company Inc.
7F, No. 316, Sec. 1, Neihu Rd., Taipei 114, Taiwan
Phone: +886-2-8751-1816
Fax: +886-2-8751-1817

Please Contact:

071001

Printed on Recycled Paper

No. MD8480C-E-E-3-(1.01) 公知 Printed in Japan 2008-3 AKD