Technical Specification Group Services and System Aspects **TSGS#20(03)0214**
Meeting #20, Hämeenlinna, Finland
09-12 June 2003

Source: **TSG-SA WG4**

Title: CRs to TS 26.104 - Correction on codec mode handling during DTX (Release 4 and Release 5)

Document for: **Approval**

Agenda Item: **7.4.3**

The following CRs, agreed at the TSG-SA WG4 meeting #26, are presented to TSG SA #20 for approval.

Spec	CR	Rev	Phase	Subject	Cat	Vers	WG	Meeting	S4 doc
26.104	025		Rel-4	Correction on codec mode handling during DTX	F	4.4.0	S4	TSG-SA WG4#26	S4-030340
26.104	026		Rel-5	Correction on codec mode handling during DTX	A	5.1.0	S4	TSG-SA WG4#26	S4-030341

3GPP TSG-SA4 Meeting #26
Paris, France, 5-9 May 2003

Tdoc ⌘S4-030340

CR-Form-v7

CHANGE REQUEST

⌘	**26.104**	**CR**	**025**	⌘**rev**	**-**	⌘	Current version:	**4.4.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects:	UICC apps⌘		ME	**X**	Radio Access Network		Core Network **X**

Title:	⌘	Correction on codec mode handling during DTX
Source:	⌘	TSG SA WG4
Work item code:	⌘	AMR
Date:	⌘	10/06/2003
Category:	⌘	**F**
Release:	⌘	Rel-4

Use one of the following categories:
- ***F*** *(correction)*
- ***A*** *(corresponds to a correction in an earlier release)*
- ***B*** *(addition of feature),*
- ***C*** *(functional modification of feature)*
- ***D*** *(editorial modification)*

Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*

Reason for change:	⌘	Codec mode not handled correctly during DTX. The codec mode handling deviates from the fixed point AMR decoder in 26.073.
Summary of change:	⌘	Do not use the invalid speech mode `MRDTX` during DTX; instead use the mode encoded in the SID frames.
Consequences if not approved:	⌘	SID frames not handled as in the fixed point AMR codec. Worse comfort noise performance than in the fixed point AMR codec may result.

Clauses affected:	⌘	C code file interf_dec.c			
		Y	**N**		
Other specs Affected:	⌘			Other core specifications	⌘
				Test specifications	
				O&M Specifications	
Other comments:	⌘				

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

1. How the code was changed

1.1 File `interf_dec.c`

1.1.1 Before the change (lines 698 – 750)

```
   /*
    * if no mode information
    * guess one from the previous frame
    */
   if ( frame_type == RX_SPEECH_BAD ) {
      if ( s->prev_ft > 3 ) {
         frame_type = RX_SID_BAD;
         mode = MRDTX;
      }
      else {
         mode = s->prev_mode;
      }
   }
   else if ( frame_type == RX_NO_DATA ) {
      mode = s->prev_mode;
   }

   if ( bfi == 1 ) {
      if ( mode < 8 ) {
         frame_type = RX_SPEECH_BAD;
      }
      else if ( mode != 15 ) {
         frame_type = RX_SID_BAD;
      }
   }

#else
   bfi = 0;
   frame_type = bits[0];

   switch ( frame_type ) {
      case 0:
         frame_type = RX_SPEECH_GOOD;
         mode = bits[245];
         Bits2Prm( mode, &bits[1], prm );
         break;

      case 1:
         frame_type = RX_SID_FIRST;
         mode = s->prev_mode;
         break;

      case 2:
         frame_type = RX_SID_UPDATE;
         mode = s->prev_mode;
         Bits2Prm( MRDTX, &bits[1], prm );
         break;

      case 3:
         frame_type = RX_NO_DATA;
         mode = s->prev_mode;
         break;
   }
```

1.1.2 After the change

```
~~   /*~~
~~    * if no mode information~~
~~    * guess one from the previous frame~~
~~    */~~
~~   if ( frame_type == RX_SPEECH_BAD ) {~~
~~      if ( s->prev_ft > 3 ) {~~
~~         frame_type = RX_SID_BAD;~~
~~         mode = MRDTX;~~
~~      }~~
~~      else {~~
~~         mode = s->prev_mode;~~
~~      }~~
~~   }~~
~~   else if ( frame_type == RX_NO_DATA ) {~~
~~      mode = s->prev_mode;~~
~~   }~~

~~   if ( bfi == 1 ) {~~
~~      if ( mode < 8 ) {~~
~~         frame_type = RX_SPEECH_BAD;~~
~~      }~~
~~      else if ( mode != 15 ) {~~
```

```
         ~~frame_type = RX_SID_BAD;~~
      ~~}~~
   ~~}~~
   if ( bfi == 1 ) {
      if ( mode <= MR122 ) {
         frame_type = RX_SPEECH_BAD;
      }
      else if ( frame_type != RX_NO_DATA ) {
         frame_type = RX_SID_BAD;
         mode = s->prev_mode;
      }
   } else {
       if ( frame_type == RX_SID_FIRST || frame_type == RX_SID_UPDATE)
{
           mode = speech_mode;
       }
       else if ( frame_type == RX_NO_DATA ) {
           mode = s->prev_mode;
       }
       /*
        * if no mode information
        * guess one from the previous frame
        */
       if ( frame_type == RX_SPEECH_BAD ) {
          mode = s->prev_mode;
          if ( s->prev_ft >= RX_SID_FIRST ) {
             frame_type = RX_SID_BAD;
          }
       }
   }
#else
   bfi = 0;
   frame_type = bits[0];

   switch ( frame_type ) {
      case 0:
         frame_type = RX_SPEECH_GOOD;
         mode = bits[245];
         Bits2Prm( mode, &bits[1], prm );
         break;

      case 1:
         frame_type = RX_SID_FIRST;
```

```
            mode = bits[245]~~s->prev_mode~~;
            break;

         case 2:
            frame_type = RX_SID_UPDATE;
            mode = bits[245]~~s->prev_mode~~;
            Bits2Prm( MRDTX, &bits[1], prm );
            break;

         case 3:
            frame_type = RX_NO_DATA;
            mode = s->prev_mode;
            break;
      }
```

3GPP TSG-SA4 Meeting #26
Paris, France, 5-9 May 2003

Tdoc ⌘S4-030341

CR-Form-v7

CHANGE REQUEST

⌘	**26.104**	**CR**	**026**	⌘**rev**	**-**	⌘	Current version:	**5.1.0**	⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ [ ] ME [X] Radio Access Network [ ] Core Network [X]

Title:	⌘	Correction on codec mode handling during DTX		
Source:	⌘	TSG SA WG4		
Work item code:	⌘	AMR	***Date:*** ⌘	10/06/2003
Category:	⌘	**A**	***Release:*** ⌘	Rel-5
		Use one of the following categories: ***F*** *(correction)* ***A*** *(corresponds to a correction in an earlier release)* ***B*** *(addition of feature),* ***C*** *(functional modification of feature)* ***D*** *(editorial modification)* Detailed explanations of the above categories can be found in 3GPP TR 21.900.		*Use one of the following releases:* *2 (GSM Phase 2)* *R96 (Release 1996)* *R97 (Release 1997)* *R98 (Release 1998)* *R99 (Release 1999)* *Rel-4 (Release 4)* *Rel-5 (Release 5)* *Rel-6 (Release 6)*

Reason for change:	⌘	Codec mode not handled correctly during DTX. The codec mode handling deviates from the fixed point AMR decoder in 26.073.
Summary of change:	⌘	Do not use the invalid speech mode `MRDTX` during DTX; instead use the mode encoded in the SID frames.
Consequences if not approved:	⌘	SID frames not handled as in the fixed point AMR codec. Worse comfort noise performance than in the fixed point AMR codec may result.

Clauses affected:	⌘	C code file interf_dec.c				
		Y	**N**			
Other specs Affected:	⌘			Other core specifications	⌘	
				Test specifications		
				O&M Specifications		
Other comments:	⌘					

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

1. How the code was changed

1.1 File `interf_dec.c`

1.1.1 Before the change (lines 698 – 750)

```
      /*
       * if no mode information
       * guess one from the previous frame
       */
      if ( frame_type == RX_SPEECH_BAD ) {
         if ( s->prev_ft > 3 ) {
            frame_type = RX_SID_BAD;
            mode = MRDTX;
         }
         else {
            mode = s->prev_mode;
         }
      }
      else if ( frame_type == RX_NO_DATA ) {
         mode = s->prev_mode;
      }

      if ( bfi == 1 ) {
         if ( mode < 8 ) {
            frame_type = RX_SPEECH_BAD;
         }
         else if ( mode != 15 ) {
            frame_type = RX_SID_BAD;
         }
      }

   #else
      bfi = 0;
      frame_type = bits[0];

      switch ( frame_type ) {
         case 0:
            frame_type = RX_SPEECH_GOOD;
            mode = bits[245];
            Bits2Prm( mode, &bits[1], prm );
            break;

         case 1:
            frame_type = RX_SID_FIRST;
            mode = s->prev_mode;
            break;

         case 2:
            frame_type = RX_SID_UPDATE;
            mode = s->prev_mode;
            Bits2Prm( MRDTX, &bits[1], prm );
            break;

         case 3:
            frame_type = RX_NO_DATA;
            mode = s->prev_mode;
            break;
      }
```

1.1.2 After the change

```
~~   /*~~
~~    * if no mode information~~
~~    * guess one from the previous frame~~
~~    */~~
~~   if ( frame_type == RX_SPEECH_BAD ) {~~
~~      if ( s->prev_ft > 3 ) {~~
~~         frame_type = RX_SID_BAD;~~
~~         mode = MRDTX;~~
~~      }~~
~~      else {~~
~~         mode = s->prev_mode;~~
~~      }~~
~~   }~~
~~   else if ( frame_type == RX_NO_DATA ) {~~
~~      mode = s->prev_mode;~~
~~   }~~

~~   if ( bfi == 1 ) {~~
~~      if ( mode < 8 ) {~~
~~         frame_type = RX_SPEECH_BAD;~~
~~      }~~
~~      else if ( mode != 15 ) {~~
```

```
         ~~frame_type = RX_SID_BAD;~~
      ~~}~~
   ~~}~~
   if ( bfi == 1 ) {
      if ( mode <= MR122 ) {
         frame_type = RX_SPEECH_BAD;
      }
      else if ( frame_type != RX_NO_DATA ) {
         frame_type = RX_SID_BAD;
         mode = s->prev_mode;
      }
   } else {
       if ( frame_type == RX_SID_FIRST || frame_type == RX_SID_UPDATE)
{
           mode = speech_mode;
       }
       else if ( frame_type == RX_NO_DATA ) {
           mode = s->prev_mode;
       }
       /*
        * if no mode information
        * guess one from the previous frame
        */
       if ( frame_type == RX_SPEECH_BAD ) {
          mode = s->prev_mode;
          if ( s->prev_ft >= RX_SID_FIRST ) {
             frame_type = RX_SID_BAD;
          }
       }
   }
#else
   bfi = 0;
   frame_type = bits[0];

   switch ( frame_type ) {
      case 0:
         frame_type = RX_SPEECH_GOOD;
         mode = bits[245];
         Bits2Prm( mode, &bits[1], prm );
         break;

      case 1:
         frame_type = RX_SID_FIRST;
```

```
            mode = bits[245]~~s->prev_mode~~;
            break;

         case 2:
            frame_type = RX_SID_UPDATE;
            mode = bits[245]~~s->prev_mode~~;
            Bits2Prm( MRDTX, &bits[1], prm );
            break;

         case 3:
            frame_type = RX_NO_DATA;
            mode = s->prev_mode;
            break;
      }
```