

3GPP2 A.S0004- 0

Version 1.0.0

Date: December 21, 1999 V&V Draft



**3RD GENERATION
PARTNERSHIP
PROJECT 2
"3GPP2"**

3GPP2 Tandem Free Operation Specification

Release A

COPYRIGHT

3GPP2 and its Organizational Partners claim copyright in this document and individual Organizational Partners may copyright and issue documents or standards publications in individual Organizational Partner's name based on this document. Requests for reproduction of this document should be directed to the 3GPP2 Secretariat at secretariat@3gpp2.org. Requests to reproduce individual Organizational Partner's documents should be directed to that Organizational Partner. See www.3gpp2.org for more information.

1 Contents

| | | |
|---------|--|----|
| 2 | Foreword..... | 5 |
| 3 | Scope | 6 |
| 4 | References..... | 6 |
| 5 | Definitions and Abbreviations..... | 6 |
| 5.1 | Definitions..... | 6 |
| 5.2 | Abbreviations | 7 |
| 6 | General Approach..... | 8 |
| 6.1 | Background | 8 |
| 6.2 | Principle of Tandem Free Operation | 9 |
| 7 | TFO Frame Structure | 10 |
| 7.1 | TFO Frame Structure for 16 kBit/s Sub-Multiplexing | 10 |
| 7.1.1 | Requirements/Assumptions for TFO Frame Structure Definitions..... | 10 |
| 7.1.2 | Coding of TFO Frames for 16 kBit/s Sub-Multiplexing | 10 |
| 7.1.2.1 | Coding of System Identification Bits (S1..S8)..... | 10 |
| 7.1.2.2 | Coding of Speech Codec Type (C1..C4) | 11 |
| 7.1.2.3 | Embedded TFO Message Indicator Bit (C5)..... | 11 |
| 7.1.2.4 | Packet Type Information for CDMA Variable Rate Codecs (D1..D9) | 11 |
| 7.1.2.5 | Coding of Time Alignment Bits (T1..T4)..... | 13 |
| 7.1.2.6 | Coding of Data Bits for VSELP, EFRC, US1, Q8 and EVRC (D-Bits) | 13 |
| 7.1.2.7 | Coding of Data Bits for Q13 (D-Bits)..... | 13 |
| 7.1.2.8 | Rate Reduction Parameters for CDMA Variable Rate Codecs..... | 14 |
| 7.1.2.9 | Dim-And-Burst with CDMA Variable Rate Codecs..... | 14 |
| 7.1.3 | TFO Frames for 16 kBit/s Sub-Multiplexing..... | 14 |
| 7.1.3.1 | TFO Frame Structure for TDMA VSELP (TIA/EIA-136-420)..... | 15 |
| 7.1.3.2 | TFO Frame Structure for TDMA EFRC (TIA/EIA-136-410)..... | 17 |
| 7.1.3.3 | TFO Frame Structure for TDMA US1 (TIA/EIA-136-430) | 19 |
| 7.1.3.4 | TFO Frame Structure for CDMA Q8 (IS-96-C)..... | 20 |
| 7.1.3.5 | TFO Frame Structure for CDMA EVRC (IS-127) | 23 |
| 7.1.3.6 | TFO Frame Structure for CDMA Q13 (IS-733)..... | 25 |
| 8 | TFO Message Structure..... | 28 |
| 8.1 | Definition of the TFO_REQ Messages | 29 |
| 8.1.1 | Definition of the TFO_Req_Extension_Block..... | 30 |
| 8.1.2 | Cyclic Redundancy Check | 30 |
| 8.1.3 | Definition of the Codec_List_Extension_Block..... | 30 |
| 8.2 | Definition of the TFO_ACK Messages..... | 31 |
| 8.2.1 | Definition of the TFO_Ack_Extension_Block..... | 31 |
| 8.3 | Definition of the TFO_TRANS Messages | 32 |
| 8.4 | Definition of the TFO_NORMAL Message..... | 32 |
| 8.5 | Definition of the TFO_FILL Message | 32 |
| 8.6 | Definition of the TFO_DUP Message..... | 32 |
| 8.7 | Definition of the TFO_SYL Message..... | 32 |

| | | |
|--------|---|----|
| 9 | Time Alignment of TFO Frames and TFO Messages..... | 32 |
| 9.1 | Time Alignment of TFO Messages | 33 |
| 9.2 | Time Alignment of TFO Frames..... | 33 |
| 9.3 | Time Alignment of TFO Frames to Downlink TRAU Frames | 34 |
| 10 | Processes for Tandem Free Operation | 35 |
| 10.1 | Rx_TRAU Process | 36 |
| 10.2 | Tx_TRAU Process | 36 |
| 10.2.1 | Downlink Speech Transmission if TFO is ON..... | 36 |
| 10.2.2 | DTX handling for ANSI-136..... | 37 |
| 10.2.3 | Synchronisation and Bit Errors in Received TFO Frames | 37 |
| 10.3 | Tx_TFO Process | 37 |
| 10.4 | Rx_TFO Process | 38 |
| 10.4.1 | Search for and Monitoring of TFO Synchronization | 38 |
| 10.4.2 | Errors in TFO Messages and TFO Frames..... | 39 |
| 10.5 | TFO_Protocol Process..... | 40 |
| 10.5.1 | Messages from Rx_TRAU or local MSC | 40 |
| 10.5.2 | Messages to Tx_TRAU..... | 40 |
| 10.5.3 | Optional Messages to the local MSC | 40 |
| 10.5.4 | Messages to and from Tx_TFO..... | 40 |
| 10.5.5 | Messages from Rx_TFO..... | 40 |
| 10.5.6 | Messages from higher system level (CDMA only) | 41 |
| 11 | State Machine of the TFO_Protocol Process for CDMA..... | 42 |
| 11.1 | Initialisation | 43 |
| 11.1.1 | Not_Active State | 43 |
| 11.1.2 | Wakeup State | 43 |
| 11.2 | Establishment | 43 |
| 11.2.1 | First_Try State | 43 |
| 11.2.2 | Continuous_Retry State..... | 43 |
| 11.2.3 | Periodic_Retry State | 43 |
| 11.2.4 | Monitor State | 44 |
| 11.2.5 | Mismatch State | 44 |
| 11.3 | Contact State | 44 |
| 11.4 | Konnect State | 44 |
| 11.5 | Operation State..... | 44 |
| 11.6 | Tandem | 44 |
| 12 | Detailed Description of TFO_Protocol for CDMA | 45 |
| 13 | State Machine of the TFO_Protocol Process for TDMA..... | 57 |
| 13.1 | Initialisation | 58 |
| 13.1.1 | Not_Active State | 58 |
| 13.1.2 | Wakeup State | 58 |
| 13.2 | Establishment | 58 |
| 13.2.1 | First_Try State | 58 |
| 13.2.2 | Continuous_Retry State..... | 58 |
| 13.2.3 | Periodic_Retry State | 59 |
| 13.2.4 | Monitor State | 59 |
| 13.2.5 | Mismatch State | 59 |
| 13.3 | Contact State | 59 |
| 13.4 | Konnect State | 59 |
| 13.5 | Operation State..... | 59 |
| 13.6 | Local HO..... | 59 |
| 13.6.1 | Fast_Try State..... | 59 |

| | | |
|----------|--|----|
| 13.6.2 | Fast_Contact State..... | 60 |
| 13.7 | Distant HO, TFO Interruption..... | 60 |
| 13.7.1 | Sync_Lost State..... | 60 |
| 13.7.2 | Re_Konnect State..... | 60 |
| 13.8 | Failure State | 60 |
| 14 | Detailed Description of TFO_Protocol for CDMA..... | 60 |
| 15 | Codec Mismatch Resolution and Codec Optimization..... | 72 |
| 16 | Annex A (Normative): Inband Signalling Protocol: Generic Structure | 74 |
| 16.1 | Generic Structure of Inband Signalling Messages | 74 |
| 16.1.1 | Frequency and Order of Bit Transmission | 74 |
| 16.1.2 | IS_Header | 75 |
| 16.1.3 | IS_Command_Block | 75 |
| 16.1.4 | IS_Extension_Block(s)..... | 76 |
| 16.2 | Detailed Specification of IS Messages..... | 76 |
| 16.2.1 | IS_REQ Message | 76 |
| 16.2.2 | IS_ACK Message..... | 76 |
| 16.2.3 | IS_IPE, IS_TRANS and IS_NORMAL Messages | 77 |
| 16.2.4 | IS_FILL Message..... | 78 |
| 16.2.5 | IS_DUP Message..... | 78 |
| 16.2.6 | IS_SYL Message..... | 78 |
| 16.3 | Keep_Open_Indication | 79 |
| 16.4 | Rules for Sending of IS Messages | 79 |
| 16.5 | IS_System_Identification_Block | 80 |
| 17.1 | Types of In Path Equipment..... | 82 |
| 17.2 | IS_Compliant IPEs | 83 |
| 17.2.1 | Typical IPEs are IS_Passive | 83 |
| 17.2.2 | IS_Message_Transparency | 83 |
| 17.2.2.1 | First IS Message | 83 |
| 17.2.2.2 | IS Messages within a Sequence..... | 83 |
| 17.2.2.3 | Isolated IS Message..... | 84 |
| 17.2.2.4 | Check if IS Message is following | 84 |
| 17.2.3 | IPE State Representation | 84 |
| 17.2.4 | IPE in Sync_Not_Found..... | 85 |
| 17.2.5 | IPE in Sync_Found..... | 86 |
| 17.2.6 | IPE in Sync_Lost..... | 86 |
| 17.2.7 | IPE in Keep_Open_Sync | 86 |
| 17.2.8 | IPE in Keep_Open_Lost..... | 86 |
| 17.3 | IPE Error Handling | 87 |
| 17.4 | IPE Transmission Delay | 87 |
| 17.4.1 | IPE Transmission Delay in Normal_Mode..... | 87 |
| 17.4.2 | IPE Transmission Delay in Transparent_Mode..... | 87 |
| 17.5 | Compliance to IS Messages | 87 |
| 17.5.1 | Compliance to IS_REQ and IS_ACK Messages..... | 88 |
| 17.5.2 | Compliance to IS_NORMAL Message | 88 |
| 17.5.3 | Compliance to IS_TRANS_x Messages | 88 |
| 17.5.4 | Compliance to IS_TRANS_x_u Messages | 88 |
| 17.5.5 | Compliance to IS_FILL Message..... | 88 |
| 17.5.6 | Compliance to IS_DUP Messages..... | 89 |
| 17.5.7 | Compliance to IS_SYL Messages | 89 |
| 18 | History..... | 90 |

2 Foreword

This Technical Specification is based on ETS GSM 08.62, Version 7.0.0, released Feb-1999. It is modified in order to adapt the Tandem Free Operation to the North American Standards ANSI-136 and ANSI-95.

The present document introduces the Inband Signalling Protocol between Transcoder/Rate Adaptor Units for speech traffic channels for the Tandem Free Operation (TFO) of Speech Codecs within the digital cellular telecommunications system. The contents of this document are subject to continuing work within the Tandem Free Operation Group and may change following standard TIA/EIA procedures.

3 Scope

This service description document details the Inband Signalling Protocol between Transcoder/Rate Adaptor Units (TRAU) for speech traffic channels for the Tandem Free Operation (TFO) of Speech Codecs, sometimes also termed "Vocoder Bypass". It is applied to the North American standards ANSI-136 and ANSI-95. The term TFO is used throughout the document to demonstrate the interoperability to TFO in GSM.

Annex A is mandatory and describes the general Inband Signalling (IS) Principle. It is identical for all systems (e.g. ANSI-136, ANSI-96, GSM) and only updated to include the System_Identifications for ANSI-136 and ANSI-95.

Annex B is informative and gives the rules for In Path Equipment (IPE). It is identical for all systems (e.g ANSI-136, ANSI-95, GSM).

The service description does - up to now - not include requirements and specifications for Inter_System operation. This will follow at a later stage.

4 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.
- A non-specific reference to an ETS shall also be taken to refer to later versions published as an EN with the same number.

- [1] TIA/EIA - 136 - 410 "VSELP"
- [2] TIA/EIA - 136 - 420 "ACELP"
- [3] TIA/EIA - 136 - 430 "US1"
- [4] ANSI/TIA/EIA-95-B-99 "Mobile Station – Base Station Compatibility Standard for Wideband Spread Spectrum Cellular Systems"
- [5] ANSI/TIA/EIA-96C-98 "Speech Service Option Standard for Wideband Spread Spectrum Systems"
- [6] IS-127 EVRC "Speech Service Option Standard for Wideband Spread Spectrum Digital Cellular System"
- [7] IS-733 "High Rate Speech Service Option for Wideband Spread Spectrum Communications System"
- [8] GSM 08.62: "Digital cellular telecommunication system (Phase 2+); Inband Tandem Free Operation(TFO) of Speech Codecs; Version 7.0.0".
- [9] ITU-T Recommendation I.130: "Method for the characterization of telecommunication services supported by an ISDN and network capabilities of an ISDN".

5 Definitions and Abbreviations

5.1 Definitions

For the purposes of the present document, the following definitions apply.

TRAU Frame is used equivalent to "TRAU Speech Frame". TRAU Frames are transmitted between TRAU and TRX.

TFO Frame is used equivalent to "TFO TRAU Frame". TFO Frames are transmitted between TFO Partners (e.g. TRAU's).

TTL is used to denominate the link between the TRAU and the TRX, independent of the application (ANSI-136 or ANSI-95).

Silence Descriptor (SID) is the replacement for the speech parameters in TFO frames containing the comfort noise parameters, which are calculated in the encoder and used to generate comfort noise in the decoder during speech pauses.

Error Concealment Indicator (ECI) is used to signal that a TFO frame does not contain valid speech parameters.

PSTN Interface is used to denominate the 64 kbit/s PCM link from/to the TRAU. Please note, that PSTN Interface does not necessarily mean a connection to a PSTN, but is a general description for a digital 64 kbit/s PCM line.

5.2 Abbreviations

For the purposes of the present document, the following abbreviations apply.

| | |
|-------------|--|
| ACELP | Algebraic Code Exited Linear Prediction (Codec) |
| BFI | Bad Frame Indicator |
| ECI | Error Concealment Indicator |
| EVRC | Enhanced Variable Rate Codec (IS-127) |
| HO | Hard Hand Off in the case of ANSI-95; Handover with TRAU change involved in the case of ANSI-136 |
| IPE | In Path Equipment |
| MS | Mobile Station |
| MSC | Mobile Switching Centre |
| PCM | Pulse_Coded_Modulation |
| PCM sample | 8-bit value representing the A_Law or μ _Law coded sample of a speech or audio signal; sometimes used to indicate the time interval between two PCM samples (125 μ s). |
| PCM_Silence | either PCM_Alaw_Silence, or PCM_ μ Law_Silence, dependent on application |
| Q8 | Speech Codec Service Option for ANSI-95 at 8 kbit/s (ANSI-96-C) |
| Q13 | Speech Codec Service Option for ANSI-95 at 13.3 kbit/s (IS-733) |
| SID | Silence Descriptor |
| SDU | Signal ... |
| TCME | TFO Circuit Multiplication Equipment |
| TFO_ACK | TFO Acknowledgement Message |
| T_Bits | Time Alignment Bits |
| TFO_FILL | TFO Fill Message |
| TFO_TRANS | TFO Transparent Mode Message |
| TFO_NORMAL | TFO Normal Mode Message |
| TFO_DUP | TFO (Half) Duplex Mode Message |
| TFO_REQ | TFO Request Message |
| TFO_SYL | TFO Sync Lost Message |
| TFO | Tandem Free Operation |
| TRAU | Transcoder and Rate Adaptor Unit; In the case of ANSI-95, the TRAU functionality is located in the SDU; the terminology TRAU is not to be confused with GSM architecture |
| TRX | Radio transceiver station |
| TTL | TRAU-TRX-Link |
| US1 | US 1 Codec (12.2 kbit/s) |
| VSELP | Vector Sum Code Exited Linear Prediction (Codec) |

6 General Approach

6.1 Background

In case of mobile-to-mobile calls (MS-MS calls) in mobile networks without TFO, the speech signal is encoded within the first mobile station for transmission on the air interface, and decoded within the associated first TRAU. The PCM samples are then transported within the fixed part of the network to the second TRAU using 64kBit/s traffic links. This second TRAU encodes the speech signal a second time for the transmission on the second air interface, and the associated mobile station decodes it again. The two Codecs (Encoder-Decoder pair) of the connection are in "Tandem Operation".

This **Tandem Operation** has several **disadvantages**:

- The two consecutive encoding/decoding processes degrade the speech quality more than necessary;
- The links between the TRAUs within the fixed network need 64 kBit/s, where 16 or 8 kBit/s would be sufficient;
- The unnecessary encoding/decoding within the TRAUs allocates DSP power.

Tandem Free Operation requires two (back and forth) "transparent" digital channels or paths between the TRAUs. Devices within these paths need to be transparent or to be switched off for the **TFO Messages** and the **TFO Frames**. To guarantee this digital transparency with **out_of_band signalling** is not trivial. Out_of_band signalling is especially not fast enough for fall back to normal operation in case of sudden interruption of the transparency of the links.

This TFO recommendation defines therefore an **inband signalling protocol**, which **tests**, if:

- an MS-MS call is given;
- the paths between the TRAUs are digitally transparent;
- both TRAUs support TFO;
- the speech Codecs on both radio legs are identical.

establishes the TFO connection by:

- commanding the paths to go transparent;
- bypassing the decoder/encoder functions within the TRAUs.

guarantees a fast fall back procedure for sudden TFO interruption and

supports:

- the resolution of Codec mismatch situations; and
- the cost efficient transmission within the fixed part of the network.

The Tandem Free Operation is **fully compatible** with existing equipment. In its basic operation **it affects only TRAUs**. The additional computational complexity is small compared to the encoding/decoding functions of the TRAUs. Mobile Station, MSC and other network elements are not at all affected in this basic operation.

In an optional mode, the TFO supports the resolution of Codec mismatch situations, i.e. the situation where the Speech-Codecs at both radio-legs are different. For this, an additional communication channel between TRAU and MSC is necessary and the MSC has to take appropriate steps to change the Codec. The procedure how to change a Codec is considered as manufacturer proprietary and not handled within this recommendation.

Once TFO functionality is implemented in TFO compatible TRAU equipment, it can be employed also for TFO connections to other systems, like GSM, ISDN phones, speech servers, Internet connections or connections to other systems, like UMTS.

6.2 Principle of Tandem Free Operation

The TRAU shall be controlled by the MSC when it is positioned remote from the MSC. In this case, the speech/data information and TRAU control signals shall be transferred between the MSC and the TRAU in frames denoted "TRAU Frames", not described further (manufacturer proprietary).

In Tandem Free Operation similar frames, denoted "**TFO Frames**", are transferred between the two TRAUs on the **PSTN-interface** (decoded speech at 64 kBit/s) by inband signalling, i.e. inserting them into the PCM sample bit stream.

In the case of Half Rate speech traffic, these TFO Frames shall be carried by 8 kBit/s traffic channels mapped onto the least significant bit (LSB) of the PCM samples.

In the case of Full Rate speech traffic, these TFO Frames shall be carried by 16 kBit/s traffic channels mapped onto the two least significant bits of the PCM samples.

TFO Frames have a fixed size (and length) of 320 bits (20 ms).

Prior and parallel to these TFO Frames also other **TFO Messages** are transferred on the PSTN interface. TFO Messages conform to the IS_Message Principles described in Annexes A and B.

The TFO protocol between the TRAUs is independent of the position of the TRAUs within the mobile networks.

A possible configuration of two TRAUs is shown in Figure 1, which is intended as a reference model.

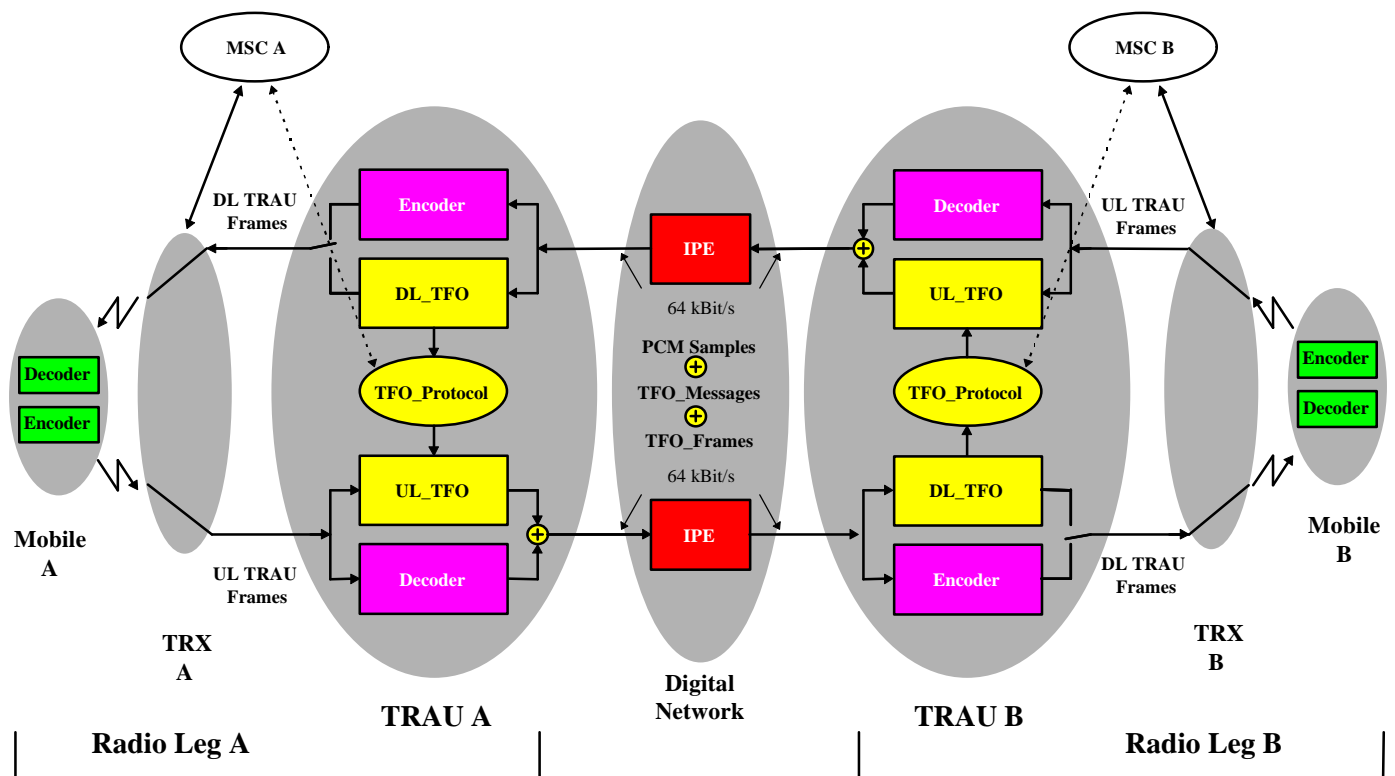


Figure 1: Functional Entities for Handling of Tandem Free Operation in MS-MS calls

TFO shall provide a virtually transparent digital channel from Encoder of Mobile A to Decoder of Mobile B and vice versa.

7 TFO Frame Structure

7.1 TFO Frame Structure for 16 kBit/s Sub-Multiplexing

This section defines TFO frame structures for various communication systems. These definitions include structures for Q8 (IS-96-C), EVRC (IS-127), Q13 (IS-733) for CDMA system and EFRC (TIA/EIA-136-410), VSELP (TIA/EIA-136-420), and US1 (TIA/EIA-136-430) for TDMA system. All TFO frame structures are defined to ensure interoperability with the existing GSM TFO standard (GSM 08.62).

7.1.1 Requirements/Assumptions for TFO Frame Structure Definitions

This section lists a set of requirements and assumptions used to define the TFO frame structures for different systems and speech codecs. Most of the requirements and assumptions are adopted from the existing GSM TFO standard (GSM 08.62) to ensure CDMA, TDMA, and GSM TFO interoperability when compatible speech codecs are in use.

- The two base station entities in a tandem call are inter-connected by a 64 kbps link with an 8000 Hz sampling frequency.
- TFO frame information is placed into the two least significant bits of each 8-bit octet (PCM sample) of the 64 kbps link between the local and remote base station units. The remaining 6 bits of each octet are filled with the corresponding decoded PCM bits.
- A TFO frame covers 160 8-bit octets and spans a period of 20 ms.
- The least significant bit (LSB) of every other 16th PCM octet is used for embedded TFO message bits.
- Eight bits are allocated for system identification. This 8-bit information is also used as a TFO frame identifier.
- Synchronization bits are defined in TFO frames to ensure TFO synchronization capability. These bits consist of 19 “1”s and eight “0”s in predefined TFO frame positions. This synchronization pattern is compatible to the GSM TFO standard (GSM 08.62)
- Each set of traffic data in a TFO frame is marked with a 4-bit speech codec type field. It is used to confirm the codec type in systems where codec change during TFO is allowed.
- Each set of traffic data within a TFO frame is protected by Cyclic Redundancy Checks (CRC) against data corruption during transmission over the inter-base station link(s).
- One bit is reserved to denote the presence of absence of TFO embedded IS_messages (see section 5.1.2.3).
- Time Alignment bits are allocated for potential phase adjustment after TFO establishment.
- All unused bit positions in a TFO frame structure are undefined and are not to be used for frame synchronization. They are marked as “Reserved”.

7.1.2 Coding of TFO Frames for 16 kBit/s Sub-Multiplexing

This section presents general information applicable to some of the fields in a TFO frame structure. Section 5.1.3 details the TFO frame structure definition information for each individual codec.

7.1.2.1 Coding of System Identification Bits (S1..S8)

The eight bits S1..S8 are used for System_Identification. They identify the sending system of the TFO_Frames.

Table 1: System Identification Assignment

| S1..S8 | System |
|-----------|--|
| 0000.0000 | GSM (for information, see ETS GSM 08.62 and 08.60) |
| 0000.0001 | TDMA |

| | |
|------------------|----------|
| 0000.0010 | CDMA |
| All others (253) | Reserved |

Note: Coding of these synchronisation bits is the same as coding of System Identification in TFO_REQ_Messages and TFO_ACK_Messages. See Annex A and chapter 6.

7.1.2.2 Coding of Speech Codec Type (C1..C4)

A 4-bit field is used to identify the codec type in the TFO frame construction. TFO frames for different systems may have the same codec type. This field, together with the 8-bit system identifier, uniquely identifies the system and codec types characterizing the TFO frame sending party.

Table 2: Speech Codec Type Assignment

| S1..S8 | System | C1..C4 |
|-----------|--------|--------------------------------------|
| 0000.0000 | GSM | (for information, see ETS GSM 08.62) |
| 0000.0001 | TDMA | 0000 (VSELP, TIA/EIA-136-420) |
| | | 0001 (EFRC, TIA/EIA-136-410) |
| | | 0010 (US1, TIA/TIA-136-430) |
| | | All others (13) reserved |
| 0000.0010 | CDMA | 0000 (Q8, IS-96-C) |
| | | 0001 (EVRC, IS-127) |
| | | 0010 (Q13, IS-733) |
| | | All others (13) reserved |

7.1.2.3 Embedded TFO Message Indicator Bit (C5)

This bit is clear (“0”) if the TFO frame has no embedded IS-message. It is set (“1”) to denote the presence of an embedded IS-message in the TFO frame.

7.1.2.4 Packet Type Information for CDMA Variable Rate Codecs (D1..D9)

Variable bit rate speech packet decoding in CDMA requires information on the packet encoding rate and the packet status. For Q8, EVRC, or Q13, the information is represented by a 3-bit field attached to each packet received from the BTS. During CDMA TFO operation, proper handling of the encoded packet received by the local BSC from the remote TFO partner requires successful transmission of the 3-bit packet type information for each encoded packet. The information is used to determine if frame erasure recovery is necessary and to ensure proper processing by the local BSC and BTS for downlink transmission to the local mobile user.

A 9-bit field, D1 to D9, is defined to provide error detection and correction capability for the 3-bit packet type information transmission. Table 3 lists the 9-bit non-linear block codes.

Table 3: Block codes for CDMA variable rate codec packet type coding

| Code index | D1..D9 |
|------------|-------------|
| 0 | 0,0000,0000 |
| 1 | 0,0001,1111 |
| 2 | 0,1111,1000 |
| 3 | 1,0110,0110 |
| 4 | 1,1101,0101 |
| 5 | 1,1010,1011 |

| | |
|---|-------------|
| 6 | 1,1000,1100 |
| 7 | 0,0111,0011 |

Table 4, 5 and 6 present a set of mappings of the Q8, EVRC, and Q13 packet type information to the code words. During a TFO frame construction, each 3-bit packet type is mapped by the Tx_TFO to a 9-bit code, except for Q13 Rate 1 packets. When a TFO frame is received by the Rx_TFO, the reverse mapping is performed to retrieve the packet type information. Additional packet type information is indirectly provided by other rate specific bit fields such as CRC bits. Information in these fields could be used in the packet type identification process.

When only the first six code words from Table 3 are used, double error correction capability is provided. For EVRC, a blank frame and a Rate 1/8 frame with all bits set to “1” are easily identified and validated with the data content. They are mapped to the last two code words, which have slightly less protection capability. Similarly, the blank frame packet type for Q8 is also mapped to the 6th code word.

Table 4: Q8 packet type information to 9-bit code word mapping. For more information on Q8 packet type information, see Table 2.3.2.2-1 IS-96-C

| Q8 packet type | Code index | D1 D2 ... D9 |
|--------------------------------------|------------|--------------|
| Rate 1 | 0 | 0,0000,0000 |
| Rate 1/2 | 1 | 0,0001,1111 |
| Rate 1/4 | 2 | 0,1111,1000 |
| Rate 1/8 | 3 | 1,0110,0110 |
| Blank | 6 | 1,1000,1100 |
| Rate 1 with bit errors | 4 | 1,1101,0101 |
| Insufficient frame quality (erasure) | 5 | 1,1010,1011 |

Table 5: EVRC packet type information to 9-bit code word mapping. For more information on EVRC packet type information, see Table 5.1.1-1 IS-127

| EVRC packet type | Code index | D1 D2 ... D9 |
|--------------------------------------|------------|--------------|
| Rate 1 | 0 | 0,0000,0000 |
| Rate 1/2 | 1 | 0,0001,1111 |
| Rate 1/4 | 2 | 0,1111,1000 |
| Rate 1/8 | 3 | 1,0110,0110 |
| Blank | 6 | 1,1000,1100 |
| Rate 1 with bit errors | 4 | 1,1101,0101 |
| Insufficient frame quality (erasure) | 5 | 1,1010,1011 |
| Rate 1/8 with all bits set to '1' | 7 | 0,0111,0011 |

Table 6: Q13 packet type information to 9-bit code word mapping. For more information on Q13 packet type information, see Table 2.3.2.2-1

| Q13 packet type | Code index | 9-bit code |
|--------------------------------------|------------|-------------|
| Rate 1 | 0 | 0,000 |
| Blank | 1 | 0,0001,1111 |
| Rate 1/4 | 2 | 0,1111,1000 |
| Rate 1/8 | 3 | 1,0110,0110 |
| Rate 1/2 | 4 | 1,1101,0101 |
| Insufficient frame quality (erasure) | 5 | 1,1010,1011 |

7.1.2.5 Coding of Time Alignment Bits (T1..T4)

Table 7: Coding of Time Alignment bits

| T1..T4 | Notes |
|--------|--|
| 1111 | Bits positioned at the end of a TFO Frame. No other codes allowed. |

If the timing of the frame is to be advanced by 250 μ s, then these four bits (T1..T4) are not transferred in order to reduce the length of the frame accordingly. If the timing is to be advanced by 125 μ s, then T3 and T4 are not transferred. If the timing of the frame is to be delayed by 125 μ s (or a multiple of it), then another pair of T-Bits (1-bits) is transferred (or a multiple of it), in order to increase the length of the frame accordingly.

Timing for the IS-733 can only be advanced by 125 μ s, as only T-Bits T1, T2 are available.

7.1.2.6 Coding of Data Bits for VSELP, EFRC, US1, Q8 and EVRC (D-Bits)

Each set of traffic data in a TFO frame is partitioned at the subframe boundaries. Each partitioned sub-set is protected by 3 parity bits at the end of the sub-set. The order within a given subset is the same as presented in section 7.1.3.

These parity bits are added to the bits of the subset, according to a degenerate (shortened) cyclic code using the generator polynomial:

$$g(D) = D^3 + D + 1$$

The encoding of the cyclic code is performed in a systematic form which means that, in $GF(2)$, the polynomial:

$$d(m)D^n + d(m+1)D^{n-1} + \dots + d(m+n-3)D^3 + p(0)D^2 + p(1)D + p(2)$$

where $p(0)$, $p(1)$, $p(2)$ are the parity bits, when divided by $g(D)$, yields a remainder equal to:

$$1 + D + D^2$$

and where $d(m)$ corresponds to a lower order D bit, e.g. D1, and $d(m+n-3)$ corresponds to a higher order D bit, e.g. D29.

For every CRC, the transmission order is $p(0)$ first followed by $p(1)$ and $p(2)$ successively.

7.1.2.7 Coding of Data Bits for Q13 (D-Bits)

Each set of traffic data in a Q13 TFO frame is protected by 7 parity bits at the end of the set. The order within a given set is the same as presented in section 7.1.3.

The CRC generator polynomial $g(X)$ is defined as

$$g(X) = 1 + X + X^2 + X^4 + X^5 + X^7$$

The input polynomial $a(X)$ consists of the N data bits to be protected, where $d(1)$ is the lower order D bit, e.g. D10 and $d(N)$ is the higher order D bit, e.g. D25.

$$a(X) = d(1)X^{N-1} + d(2)X^{N-2} + \dots + d(N-1)X + d(N)X^0$$

The parity polynomial $b(X)$ is the remainder of the division of the input polynomial and the generator polynomial, i.e.

$$\frac{a(X)X^7}{g(X)} = q(X) + \frac{b(X)}{g(X)}$$

$$b(X) = C(1)X^6 + C(2)X^5 + C(3)X^4 + C(4)X^3 + C(5)X^2 + C(6)X + C(7)X^0$$

The CRC bits are written into the TFO frame in ascending order, i.e. $C(1)$ is first and $C(7)$ is last.

The quotient polynomial $q(X)$ is discarded.

7.1.2.8 Rate Reduction Parameters for CDMA Variable Rate Codecs

The TFO frame structures for CDMA variable rate codecs Q8, EVRC, and Q13 typically contain the local system downlink and uplink rate reduction parameters. Upon receiving a remote downlink rate reduction parameter from the remote TFO partner for CDMA variable rate coding, the local system may command the local mobile station to perform speech encoding for local uplink transmission at the matching rate reduction mode. This measure avoids unnecessary rate conversions by the remote TFO partner and thus avoids any possible artifacts associated with the rate conversion process in the remote mobile.

Uplink rate reduction parameter is transmitted to the remote TFO partner for information only.

7.1.2.9 Dim-And-Burst with CDMA Variable Rate Codecs

The Dim-And-Burst feature of ANSI-95 requires the TRAU to lower the coding rate upon request. During TFO, however, the coding rate is not determined by the TRAU itself, but by the remote mobile. For this situation, it is required to implement a functionality, which is capable of reducing a rate 1 frame to a frame of a lower rate. The design of a suitable algorithm is manufacturer dependent and not scope of this document.

7.1.3 TFO Frames for 16 kBit/s Sub-Multiplexing

This section presents TFO frame structures in detail for the speech codecs presented above.

Bit m of octet n , shall be transmitted in the **Least Significant Bit** of the PCM sample $k = n*4 + (m+1)/2$ for $m = (1, 3, 5, 7)$ (grey shaded columns) and $n = (0..39)$.

Bit m of octet n shall be transmitted in the **second Least Significant Bit** of the PCM sample $k = n*4 + m/2$ for $m = (2, 4, 6, 8)$ (not shaded columns) and $n = (0..39)$.

7.1.3.1 TFO Frame Structure for TDMA VSELP (TIA/EIA-136-420)

TFO frame structure for TDMA VSELP speech codec is as defined in Table 8. Each frame consists of 320 bits. For the purpose of this description the 320 bits of one TFO frame are arranged in 40 rows (0..39), with 8 bits (1..8: one octet) each.

Table 8: TFO Frame Structure for TDMA VSELP speech codec

| Octet no. (n) | Bit number (m) | | | | | | | |
|------------------|----------------|------|-------------|--------------|-------------|------|-------------|------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 0 | <u>0</u> | S1 | 0 | S2 | 0 | S3 | 0 | S4 |
| 1 | 0 | S5 | 0 | S6 | 0 | S7 | 0 | S8 |
| 2 | 1 | C1 | C2 | C3 | C4 | C5 | C6 | C7 |
| 3 | C8 | C9 | C10 | C11 | C12 | C13 | C14 | C15 |
| 4 | <u>1</u> | D1 | D2 | D3 | D4 | D5 | D6 | D7 |
| 5 | | | | | | | | |
| 6 | <u>1</u> | | | | | | | |
| 7 | | | | | | | | |
| 8 | <u>1</u> | | | | | | | |
| 9 | | | | | | | | |
| 10 | <u>1</u> | | | | | | | |
| 11 | | | | | | | | |
| 12 | <u>1</u> | | | | | | | |
| 13 | | | | | | | | |
| 14 | <u>1</u> | | | Other D bits | | | | |
| 15 | | | | | | | | |
| 16 | <u>1</u> | | | | | | | |
| 17 | | | | | | | | |
| 18 | <u>1</u> | | | | | | | |
| 19 | | | | | | | | |
| 20 | <u>1</u> | | | | | | | |
| 21 | | | | | | | | |
| 22 | <u>1</u> | | | | | | | |
| 23 | | | | | | | | |
| 24 | <u>1</u> | | | | | | | |
| 25 | D158 | D159 | D160 | D161 | D162 | D163 | D164 | D165 |
| 26 | 1 | D166 | D167 | D168 | D169 | D170 | D171 | C16 |
| 27 | C17 | C18 | C19 | C20 | C21 | C22 | C23 | C24 |
| 28 | <u>1</u> | | | | | | | |
| 29 | | | | | | | | |
| 30 | <u>1</u> | | | | | | | |
| 31 | | | | | | | | |
| 32 | <u>1</u> | | | Other C bits | | | | |
| 33 | | | | | | | | |
| 34 | <u>1</u> | | | | | | | |
| 35 | | | | | | | | |
| 36 | <u>1</u> | | | | | | | |
| 37 | | | | | | | | |
| 38 | 1 | C100 | C101 | C102 | C103 | C104 | C105 | C106 |
| 39 | C107 | C108 | C109 | C110 | T1 | T2 | T3 | T4 |

S1..S8: System identifier

C1..C4: Codec type

C5: Embedded TFO message indicator bit

C6..C11: Reserved.

C12: Bad Frame Indicator (BFI). Set if the speech frame has a CRC failure during the over the air uplink transmission. Cleared for valid frame.

| | |
|-------------|--|
| C13..C15: | Reserved. |
| C16: | Error concealment indicator. Set if the frame underwent error concealment treatment in the local decoder. Cleared otherwise. |
| C17..C110: | Reserved. |
| D1..D72: | R0, LPC and subframe 1 information bits |
| D73..D75: | CRC over R0, LPC and subframe 1 information bits |
| D76..D104: | Subframe 2 information bits |
| D105..D107: | CRC over subframe 2 information bits |
| D108..D136: | Subframe 3 information bits |
| D137..D139: | CRC over subframe 3 information bits |
| D140..D168: | Subframe 4 information bits |
| D169..D171: | CRC over subframe 4 information bits |

Notes: VSELP encoded parameters are transmitted in the order specified in Table 4, TIA/EIA-136-420, and with the most significant bits first.

7.1.3.2 TFO Frame Structure for TDMA EFRC (TIA/EIA-136-410)

TFO frame structure for TDMA EFRC speech codec is as defined in Table 9. Each frame consists of 320 bits. For the purpose of this description the 320 bits of one TFO frame are arranged in 40 rows (0..39), with 8 bits (1..8: one octet) each.

Table 9: TFO Frame Structure for TDMA EFRC speech codec

| Octet no. (n) | Bit number (m) | | | | | | | |
|------------------|----------------|------|--------------|------|------|------|------|------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 0 | 0 | S1 | 0 | S2 | 0 | S3 | 0 | S4 |
| 1 | 0 | S5 | 0 | S6 | 0 | S7 | 0 | S8 |
| 2 | 1 | C1 | C2 | C3 | C4 | C5 | C6 | C7 |
| 3 | C8 | C9 | C10 | C11 | C12 | C13 | C14 | C15 |
| 4 | 1 | D1 | D2 | D3 | D4 | D5 | D6 | D7 |
| 5 | | | | | | | | |
| 6 | 1 | | | | | | | |
| 7 | | | | | | | | |
| 8 | 1 | | | | | | | |
| 9 | | | | | | | | |
| 10 | 1 | | | | | | | |
| 11 | | | | | | | | |
| 12 | 1 | | | | | | | |
| 13 | | | | | | | | |
| 14 | 1 | | Other D bits | | | | | |
| 15 | | | | | | | | |
| 16 | 1 | | | | | | | |
| 17 | | | | | | | | |
| 18 | 1 | | | | | | | |
| 19 | | | | | | | | |
| 20 | 1 | | | | | | | |
| 21 | | | | | | | | |
| 22 | 1 | | | | | | | |
| 23 | | | | | | | | |
| 24 | 1 | D151 | D152 | D153 | D154 | D155 | D156 | D157 |
| 25 | D158 | D159 | D160 | C16 | C17 | C18 | C19 | C20 |
| 26 | 1 | C21 | C22 | C23 | C24 | C25 | C26 | C27 |
| 27 | | | | | | | | |
| 28 | 1 | | | | | | | |
| 29 | | | | | | | | |
| 30 | 1 | | | | | | | |
| 31 | | | Other C bits | | | | | |
| 32 | 1 | | | | | | | |
| 33 | | | | | | | | |
| 34 | 1 | | | | | | | |
| 35 | | | | | | | | |
| 36 | 1 | | | | | | | |
| 37 | | | | | | | | |
| 38 | 1 | C111 | C112 | C113 | C114 | C115 | C116 | C117 |
| 39 | C118 | C119 | C120 | C121 | T1 | T2 | T3 | T4 |

S1..S8: System identifier

C1..C4: Codec type

C5: Embedded TFO message indicator bit

C6..C11: Reserved.

C12: Bad Frame Indicator (BFI). Set if the speech frame has a CRC failure during the over the air uplink transmission or if the frame is not a speech frame. Cleared for valid frame.

- C13: Comfort Noise Bad Frame Indicator (BFI_CN): Set if the comfort noise parameter frame has a CRC failure during the over the air uplink transmission or if the frame is not a comfort noise parameter frame. Cleared otherwise.
- C14..C15: Reserved.
- C16: Error concealment indicator. Set if the frame underwent error concealment treatment in the local decoder. Cleared otherwise.
- C17: DTX high/low state indicator. Set if DTX is applied on the frame in the local uplink transmission. Cleared otherwise.
- C18..C121: Reserved.

Speech TFO frame:

- D1..D58: LPC and subframe 1 information bits s0 to s57
- D59..D61: CRC over bits s0 to s57
- D62..D90: Subframe 2 information bits s58 to s86
- D91..D93: CRC over bits s58 to s86
- D94..D125: Subframe 3 information bits s87 to s118
- D126..D128: CRC over bits s87 to s118
- D129..D157: Subframe 4 information bits s119 to s147
- D158..D160: CRC over bits s119 to s147

Comfort noise parameter TFO frame:

- D1..D38: LPC, excitation gain and RESC parameter bits cn0 to cn37
- D39..D41: CRC over bits cn0 to cn37
- D42..D160: Reserved.

Blank TFO frame:

- D1..D160: Reserved.

- T1..T4: Time alignment bits

Notes:

- For speech parameter bit order information, see TIA/EIA/136-410 Table 4.1.
- During DTX-low, local Tx_TFO sends comfort noise parameter TFO frames to the remote TFO partner. The remote TFO partner is to convert these TFO frames into normal speech frame format for the remote mobile station. This approach is taken by the GSM TFO standard and is adopted here to ensure interoperability with the GSM TFO standard.
- During DTX-low and in the absence of comfort noise parameters, local Tx_TFO sends blank TFO frames to the remote TFO partner. The remote TFO partner is to convert these blank TFO frames into normal speech frame format for the remote mobile station.
- Speech, comfort noise parameter or blank TFO frames are marked by control bits C12, C13 and C17.

7.1.3.3 TFO Frame Structure for TDMA US1 (TIA/EIA-136-430)

TFO frame structure for TDMA US1 speech codec is as defined in Table 10. Each frame consists of 320 bits. For the purpose of this description the 320 bits of one TFO Frame are arranged in 40 rows (0..39), with 8 bits (1..8: one octet) each.

Table 10: TFO Frame Structure for TDMA US1 speech codec

| Octet no. (n) | Bit number (m) | | | | | | | |
|------------------|----------------|------|------|--------------|------|------|-----|-----|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 0 | 0 | S1 | 0 | S2 | 0 | S3 | 0 | S4 |
| 1 | 0 | S5 | 0 | S6 | 0 | S7 | 0 | S8 |
| 2 | 1 | C1 | C2 | C3 | C4 | C5 | C6 | C7 |
| 3 | C8 | C9 | C10 | C11 | C12 | C13 | C14 | C15 |
| 4 | 1 | D1 | D2 | D3 | D4 | D5 | D6 | D7 |
| 5 | | | | | | | | |
| 6 | 1 | | | | | | | |
| 7 | | | | | | | | |
| 8 | 1 | | | | | | | |
| 9 | | | | | | | | |
| 10 | 1 | | | | | | | |
| 11 | | | | | | | | |
| 12 | 1 | | | | | | | |
| 13 | | | | | | | | |
| 14 | 1 | | | | | | | |
| 15 | | | | | | | | |
| 16 | 1 | | | | | | | |
| 17 | | | | | | | | |
| 18 | 1 | | | | | | | |
| 19 | | | | Other D bits | | | | |
| 20 | 1 | | | | | | | |
| 21 | | | | | | | | |
| 22 | 1 | | | | | | | |
| 23 | | | | | | | | |
| 24 | 1 | | | | | | | |
| 25 | | | | | | | | |
| 26 | 1 | | | | | | | |
| 27 | | | | | | | | |
| 28 | 1 | | | | | | | |
| 29 | | | | | | | | |
| 30 | 1 | | | | | | | |
| 31 | | | | | | | | |
| 32 | 1 | | | | | | | |
| 33 | | | | | | | | |
| 34 | 1 | | | | | | | |
| 35 | | | | | | | | |
| 36 | 1 | | | | | | | |
| 37 | | | | | | | | |
| 38 | 1 | D256 | D257 | D258 | D259 | D260 | C16 | C17 |
| 39 | C18 | C19 | C20 | C21 | T1 | T2 | T3 | T4 |

S1..S8: System identifier

C1..C4: Codec type

C5: Embedded TFO message indicator bit

C6..C11: Reserved.

C12: Bad Frame Indicator (BFI). Set if the frame has a CRC failure during the over the air uplink transmission. Cleared for valid frame.

- C13..C14: Silence Descriptor (SID). (0,0) for valid encoded speech frame and (1,0) for valid comfort noise parameter frame in DTX.
- C15: Time Alignment Flag (TAF). Set if information in the frame should be used for comfort noise parameter updates in DTX. Cleared otherwise.
- C16: Error concealment indicator. Set if the frame underwent frame erasure treatment in the local decoder. Cleared otherwise.
- C17: DTX high/low state indicator. Set if DTX is applied on the frame in the local uplink transmission. Cleared otherwise.
- C18..C21: Reserved.

Speech TFO frame:

- D1: Reserved (set to "1").
- D2..D39: LPC information bits s1 to s38
- D40..D42: CRC over bits D1 to D22, D25 to D27 and D29
- D43..D95: Subframe 1 information bits s39 to s91
- D96..D98: CRC over bits D43 to D52, D91 and D92
- D99..D148: Subframe 2 information bits s92 to s141
- D149..D151: CRC over bits D99 to D103, D105, D144 and D145
- D152..D204: Subframe 3 information bits s142 to s194
- D205..D207: CRC over bits D152 to D161, D200 and D201
- D208..D257: Subframe 4 information bits s195 to s244
- D258..D260: CRC over bits D208 to D212, D214, D253 and D254

Comfort noise parameter TFO frame:

- D1: Reserved (set to "1")
- D2..D92; LPC and codebook gain bits s1 to s91
- D93..D95: CRC over bits D2 to D92
- D96..D260: Reserved.

Blank TFO frame:

- D1: Reserved (set to "1")
- D2..D260: Reserved.

- T1..T4: Time alignment bits

Notes:

- This frame structure is compatible with the GSM Enhanced Full Rate frame structure. See GSM 08.60 Version 8.0.
- For information on US1 speech parameter bit order, see GSM 06.60 Table 6.
- During DTX-low, local Tx_TFO sends comfort noise parameter TFO frames to the remote TFO partner. The remote TFO partner is to convert these TFO frames into normal speech frame format for the remote mobile station. This approach is taken by the GSM TFO standard and is adopted here to ensure interoperability with the GSM TFO standard.
- During DTX-low and in the absence of comfort noise parameters, local Tx_TFO sends blank TFO frames to the remote TFO partner. The remote TFO partner is to convert these blank TFO frames into normal speech frame format for the remote mobile station.
- Speech, comfort noise parameter or blank TFO frames are marked by control bits C12, C13, C14, C15 and C17.

7.1.3.4 TFO Frame Structure for CDMA Q8 (IS-96-C)

TFO frame structure for CDMA Q8 speech codec is as defined in Table 11. Each frame consists of 320 bits. For the purpose of this description the 320 bits of one TFO Frame are arranged in 40 rows (0..39), with 8 bits (1..8: one octet) each.

Table 11: TFO Frame Structure for CDMA Q8 speech codec

| Octet no. (n) | Bit number (m) | | | | | | | |
|------------------|----------------|------|--------------|------|------|------|------|------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 0 | 0 | S1 | 0 | S2 | 0 | S3 | 0 | S4 |
| 1 | 0 | S5 | 0 | S6 | 0 | S7 | 0 | S8 |
| 2 | 1 | C1 | C2 | C3 | C4 | C5 | C6 | C7 |
| 3 | C8 | C9 | C10 | C11 | C12 | C13 | C14 | C15 |
| 4 | 1 | D1 | D2 | D3 | D4 | D5 | D6 | D7 |
| 5 | | | | | | | | |
| 6 | 1 | | | | | | | |
| 7 | | | | | | | | |
| 8 | 1 | | | | | | | |
| 9 | | | | | | | | |
| 10 | 1 | | | | | | | |
| 11 | | | | | | | | |
| 12 | 1 | | | | | | | |
| 13 | | | | | | | | |
| 14 | 1 | | | | | | | |
| 15 | | | Other D bits | | | | | |
| 16 | 1 | | | | | | | |
| 17 | | | | | | | | |
| 18 | 1 | | | | | | | |
| 19 | | | | | | | | |
| 20 | 1 | | | | | | | |
| 21 | | | | | | | | |
| 22 | 1 | | | | | | | |
| 23 | | | | | | | | |
| 24 | 1 | | | | | | | |
| 25 | | | | | | | | |
| 26 | 1 | | | | | | | |
| 27 | | | | | | | | |
| 28 | 1 | D181 | D182 | D183 | D184 | D185 | D186 | D187 |
| 29 | D188 | D189 | D190 | D191 | D192 | C16 | C17 | C18 |
| 30 | 1 | C19 | C20 | C21 | C22 | C23 | C24 | C25 |
| 31 | | | | | | | | |
| 32 | 1 | | | | | | | |
| 33 | | | | | | | | |
| 34 | 1 | | | | | | | |
| 35 | | | Other C bits | | | | | |
| 36 | 1 | | | | | | | |
| 37 | | | | | | | | |
| 38 | 1 | C79 | C80 | C81 | C82 | C83 | C84 | C85 |
| 39 | C86 | C87 | C88 | C89 | T1 | T2 | T3 | T4 |

S1..S8: System identifier

C1..C4: Codec type

C5: Embedded TFO message indicator bit

C6..C15: Reserved.

C16..C18: Rate reduction parameter, specifying the fraction of packets of Rate 1 and Rate 1/2 in the local uplink

C19..C21: Rate reduction parameter, specifying the fraction of packets of Rate 1 and Rate 1/2 in the local downlink

C22: Error concealment indicator. Set if the frame underwent frame erasure treatment in the local decoder. Cleared otherwise.

C23..C89: Reserved.

D1..D9: Packet type from local multiplex sublayer on received packet from local mobile (See IS-96-C Table 2.3.2.2-1).

Rate 1 packet:

D10..D79: LPC and subframe 1 information bits s170 to s128, s126 to s120, s118 to s112, s110 to s104, s102 to s97
D80..D82: CRC over bits s170 to s128, s126 to s120, s118 to s112, s110 to s104, s102 to s97
D83..D114: Subframe 2 information bits s127, s119, s96, s94 to s88, s86 to s65
D115..D117: CRC over bits s127, s119, s96, s94 to s88, s86 to s65
D118..D151: Subframe 3 information bits s111, s103, s64 to s33
D152..D154: CRC over bits s111, s103, s64 to s33
D155..D189: Subframe 4 information bits s95, s87, s32 to s0
D190..D192: CRC over bits s95, s87, s32 to s0

Rate 1/2 packet:

D10..D59: LPC and subframe 1 information bits s79 to s30
D60..D62: CRC over bits s79 to s30
D63..D92: Subframe 2 information bits s29 to s0
D93..D95: CRC over bits s29 to s0
D96..D192: Reserved.

Rate 1/4 packet:

D10..D49: Information bits s39 to s0
D50..D52: CRC over bits s39 to s0
D53..D192: Reserved.

Rate 1/8 packet:

D10..D25: Information bits s15 to s0
D26..D28: CRC over bits s15 to s0
D29..D192: Reserved.

T1..T4: Time Alignment Bits

Note:

- See IS-96-C Tables 2.4.7.1.2-1, 2.4.7.2-1, 2.4.7.3-1 and 2.4.7.4-1 for information on speech parameter bit order.
- See IS-96-C Table 2.3.3.1.2-1 for rate reduction parameter values.

7.1.3.5 TFO Frame Structure for CDMA EVRC (IS-127)

TFO frame structure for CDMA EVRC speech codec is as defined in Table 12. Each frame consists of 320 bits. For the purpose of this description the 320 bits of one TFO frame are arranged in 40 rows (0..39), with 8 bits (1..8: one octet) each.

Table 12: TFO Frame Structure for CDMA EVRC speech codec

| Octet no. (n) | Bit number (m) | | | | | | | |
|------------------|----------------|------|--------------|------|------|------|------|------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 0 | 0 | S1 | 0 | S2 | 0 | S3 | 0 | S4 |
| 1 | 0 | S5 | 0 | S6 | 0 | S7 | 0 | S8 |
| 2 | 1 | C1 | C2 | C3 | C4 | C5 | C6 | C7 |
| 3 | C8 | C9 | C10 | C11 | C12 | C13 | C14 | C15 |
| 4 | 1 | D1 | D2 | D3 | D4 | D5 | D6 | D7 |
| 5 | | | | | | | | |
| 6 | 1 | | | | | | | |
| 7 | | | | | | | | |
| 8 | 1 | | | | | | | |
| 9 | | | | | | | | |
| 10 | 1 | | | | | | | |
| 11 | | | | | | | | |
| 12 | 1 | | | | | | | |
| 13 | | | | | | | | |
| 14 | 1 | | | | | | | |
| 15 | | | Other D bits | | | | | |
| 16 | 1 | | | | | | | |
| 17 | | | | | | | | |
| 18 | 1 | | | | | | | |
| 19 | | | | | | | | |
| 20 | 1 | | | | | | | |
| 21 | | | | | | | | |
| 22 | 1 | | | | | | | |
| 23 | | | | | | | | |
| 24 | 1 | | | | | | | |
| 25 | | | | | | | | |
| 26 | 1 | | | | | | | |
| 27 | | | | | | | | |
| 28 | 1 | D181 | D182 | D183 | D184 | D185 | D186 | D187 |
| 29 | D188 | D189 | D190 | D191 | D192 | C16 | C17 | C18 |
| 30 | 1 | C19 | C20 | C21 | C22 | C23 | C24 | C25 |
| 31 | | | | | | | | |
| 32 | 1 | | | | | | | |
| 33 | | | | | | | | |
| 34 | 1 | | Other C bits | | | | | |
| 35 | | | | | | | | |
| 36 | 1 | | | | | | | |
| 37 | | | | | | | | |
| 38 | 1 | C79 | C80 | C81 | C82 | C83 | C84 | C85 |
| 39 | C86 | C87 | C88 | C89 | T1 | T2 | T3 | T4 |

S1..S8: System identifier

C1..C4: Codec type

C5: Embedded TFO message indicator bit

C6..C15: Reserved.

C16..C18: Rate reduction parameter, specifying the fraction of packets of Rate 1 and Rate 1/2 in the local uplink

C19..C21: Rate reduction parameter, specifying the fraction of packets of Rate 1 and Rate 1/2 in the local downlink

| | |
|------------------|---|
| C22: | Error concealment indicator. Set if the frame underwent frame erasure treatment in the local decoder. Cleared otherwise. |
| C23..C89: | Reserved. |
| D1..D9: | Packet type from local multiplex sublayer on received packet from local mobile. (See IS-127 Table 5.1.1-1) |
| Rate 1 packet: | |
| D10..D50: | LPC and pitch delay information bits s1 to s41 |
| D51..D53: | CRC over bits s1 to s41 |
| D54..D96: | Subframe 1 codebook information bits s42 to s84 |
| D97..D99: | CRC over bits s42 to s84 |
| D100..D142: | Subframe 2 codebook information bits s85 to s127 |
| D143..D145: | CRC over bits s85 to s127 |
| D146..D189: | Subframe 3 codebook information bits s128 to s171. Bit s171 is reserved according to IS-127 and is set to "0", to be consistent with Q13 specification. |
| D190..D192: | CRC over bits s128 to s171 |
| Rate 1/2 packet: | |
| D10..D38: | LPC and pitch delay information bits s1 to s29 |
| D39..D41: | CRC over bits s1 to s29 |
| D42..D58: | Subframe 1 codebook information bits s30 to s46 |
| D59..D61: | CRC over bits s30 to s46 |
| D62..D78: | Subframe 2 codebook information bits s47 to s63 |
| D79..D81: | CRC over bits s47 to s63 |
| D82..D98: | Subframe 3 codebook information bits s64 to s80 |
| D99..D101: | CRC over bits s64 to s80 |
| D102..D192: | Reserved. |
| Rate 1/8 packet: | |
| D10..D25: | LPC and gain information bits s1 to s16 |
| D26..D28: | CRC over bits s1 to s16 |
| D29..D192: | Reserved. |
| T1..T4: | Time Alignment Bits |

Note:

- See IS-127 Table 4.8-1 for information on speech parameter bit order.
- See IS-127 Table 2.2.1.2-1 for rate reduction parameter values.

7.1.3.6 TFO Frame Structure for CDMA Q13 (IS-733)

TFO frame structure for CDMA Q13 speech codec is as defined in Table 13. Each frame consists of 320 bits. For the purpose of this description the 320 bits of one TFO Frame are arranged in 40 rows (0..39), with 8 bits (1..8: one octet) each.

Table 13: TFO Frame Structure for CDMA Q13 speech codec

| Octet no. (n) | Bit number (m) | | | | | | | | |
|------------------|----------------|----------|--------------|----------|----------|----------|----------|----------|--|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | |
| 0 | 0 | S1 | 0 | S2 | 0 | S3 | 0 | S4 | |
| 1 | 0 | S5 | 0 | S6 | 0 | S7 | 0 | S8 | |
| 2 | 1 | C1 | C2 | C3 | C4 | C5 | C6 | C7 | |
| 3 | C8 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | |
| 4 | 1 | | | | | | | | |
| 5 | | | | | | | | | |
| 6 | 1 | | | | | | | | |
| 7 | | | | | | | | | |
| 8 | 1 | | | | | | | | |
| 9 | | | | | | | | | |
| 10 | 1 | | | | | | | | |
| 11 | | | | | | | | | |
| 12 | 1 | | | | | | | | |
| 13 | | | | | | | | | |
| 14 | 1 | | | | | | | | |
| 15 | | | | | | | | | |
| 16 | 1 | | | | | | | | |
| 17 | | | Other D bits | | | | | | |
| 18 | 1 | | | | | | | | |
| 19 | | | | | | | | | |
| 20 | 1 | | | | | | | | |
| 21 | | | | | | | | | |
| 22 | 1 | | | | | | | | |
| 23 | | | | | | | | | |
| 24 | 1 | | | | | | | | |
| 25 | | | | | | | | | |
| 26 | 1 | | | | | | | | |
| 27 | | | | | | | | | |
| 28 | 1 | | | | | | | | |
| 29 | | | | | | | | | |
| 30 | 1 | | | | | | | | |
| 31 | | | | | | | | | |
| 32 | 1 | | | | | | | | |
| 33 | | | | | | | | | |
| 34 | 1 | | | | | | | | |
| 35 | | | | | | | | | |
| 36 | 1 | | | | | | | | |
| 37 | D255 | D256 | D257 | D258 | D259 | D260 | D261 | D262 | |
| 38 | 1 | D263/C9 | D264/C10 | D265/C11 | D266/C12 | D267/C13 | D268/C14 | D269/C15 | |
| 39 | D270/C16 | D271/C17 | D272/C18 | D273/C19 | D274/C20 | D275/C21 | T1 | T2 | |

S1..S8: System identifier/TFO frame synch word

C1..C4: Codec type

C5: Embedded TFO message indicator bit

C6..C8: Rate reduction parameter, specifying the fraction of packets of rate 1, rate 1/2 or rate 1/4 in the local downlink

C9..C11: These bits are not defined for Rate 1 TFO frame. For Rate 1 TFO frame, these are interpreted as D bits.

Rate reduction parameter, specifying the fraction of packets of rate 1, rate ½ or rate ¼ in the local uplink.

C12..C21 These bits are not defined for Rate 1 TFO frame. For Rate 1 TFO frame, these are interpreted as D bits. Reserved.

Rate 1 packet:

D1..D4: 4-bit packet type code (0000, see Table 6). Packet type is from local multiplex sublayer on received packet from local mobile.

D5..D268: Information bits s265 to s10, s7 to s0.

D269..D275: CRC over bits s265 to s0 except all 112 CINDEXT bits and 2 reserved bits (see section 5.1.2.7).

Rate ½ packet:

D1..D9: 9-bit packet type code (1,1101,0101, see Table 6).

D10..D123: Information bits s123 to s0.

D124..D130: CRC over bits s123 to s0 except all 28 CINDEXT bits.

D131..D262: Reserved.

Rate ¼ packet:

D1..D9: 9-bit packet type code (0,1111,1000, see Table 6).

D10..D61: Information bits s53 to s6, s3 to s0.

D62..D68: CRC over bits s53 to s6, s3 to s0.

D69..D262: Reserved.

Rate 1/8 packet:

D1..D9: 9-bit packet type code (1,0110,0110, see Table 6).

D10..D25: All information bits s19 to s4.

D26..D32: CRC over bits s19 to s4 except all 4 CBSEED bits.

D33..D262: Reserved.

Blank:

D1..D9: 9-bit packet type code (0,0001,1111, see Table 6)

D10..D18: 9-bit packet type code (0,0001,1111)

D19..D27: 9-bit packet type code (0,0001,1111)

D28..D262: Reserved.

Insufficient Frame Quality (Erasure):

D1..D9: 9-bit packet type code (1,1010,1011, see Table 6)

D10..D18: 9-bit packet type code (1,1010,1011)

D19..D27: 9-bit packet type code (1,1010,1011)

D28..D262: Reserved.

T1..T2: Time Alignment Bits

Notes:

- See IS-733 Tables 2.4.7.1-1, 2.4.7.2-1, 2.4.7.3-1 and 2.4.7.4-1 for information on speech parameter bit order.

- Error concealment on received frames corrupted by channel noise on the uplink is to be done in the local decoder. The option of it being handled by either the local or remote processor and indicated by the error concealment indicator in the case of Q8 and EVRC is not available.
- See IS-733 Table 2.3.5.2-2 for rate reduction parameter values.
- See IS-733 Table 2.3.2.2-1 for packet type information.
- CRC bits do not protect fixed codebook index bits. The fixed codebook index bits are replaced by random indices on erasure (see IS-733 section 2.4.8.7.1).
- Q13 Rate 1, Rate $\frac{1}{4}$ and Rate $\frac{1}{8}$ reserved bits are not transmitted.

TFO Message Structure

Several TFO Messages are defined, based on the general IS_Message principle, as defined in Annex A.

Definition for ***Sender side***:

TFO_REQ (): Identifies the source of the message as a TFO capable device, using a defined speech Codec.

TFO_REQ contains the following parameters ():

- the specific Local_Signature of the sender;
- the Local_Used_Codec at sender side;
- the System_Identification.

TFO_REQ_L (): Is sent in case of Codec Mismatch or for sporadic updates of information.

TFO_REQ_L contains the following parameters ():

- the specific Local_Signature of the sender;
- the Local_Used_Codec at sender side;
- the System_Identification;
- the Local_Codec_List of alternative Codecs.

TFO_REQ_P ():

TFO_REQ_P contains the following parameters ():

- the specific Local_Signature of the Sender;
- the Preferred_Codec at sender side;
- the System_Identification;
- the Local_Codec_List of alternative Codecs.

TFO_ACK (): Is the response to a TFO_REQ Message. It contains the following parameters ():

- the Reflected_Signature, copied from the received TFO_REQ Message;
- the Local_Used_Codec at sender side;
- the System_Identification.

TFO_ACK_L (): Is the response to a TFO_REQ_L Message.

It contains the following parameters ():

- the Reflected_Signature, copied from the received TFO_REQ_L Message;
- the Local_Used_Codec at sender side;
- the System_Identification;
- the Local_Codec_List of alternative Codecs.

TFO_TRANS (): Commands possible IPEs to let the TFO Frames pass transparently within the LSB (8 kBit/s) or the two LSBs (16 kBit/s). TFO_TRANS contains the following parameter ():

- the Local_Channel_Type (i.e. 8 kBit/s or 16 kBit/s).

TFO_NORMAL: Commands possible IPEs to revert to normal operation.

TFO_NORMAL has no parameters.

TFO_DUP: Informs the distant partner that TFO Frames are received, while still transmitting PCM samples.

TFO_DUP has no parameters.

TFO_SYL: Informs the distant partner (if still possible) that TFO Frames are no longer received.

TFO_SYL has no parameters.

TFO_FILL: Message without specific meaning, used to pre-synchronise IPEs or to bridge over gaps in TFO protocols. TFO_FILL has no parameters.

Definition: A TFO Message is called "regular", if it is sent inserted into the PCM sample stream. A TFO Message is called "embedded", if it is sent together with (embedded into) TFO Frames, see also subclause 7.2. The bit stealing scheme (see Annex A) is identical for regular and embedded TFO Messages. Due to the specific construction of the TFO Messages, they replace some of the synchronisation bits of the TFO Frames. TFO Frame synchronisation is in case of embedded TFO Messages therefore different, however, not endangered. Data and other control bits of the TFO Frames are not affected by embedded TFO Messages.

8.1 Definition of the TFO_REQ Messages

Symbolic Notation: TFO_REQ (System_Identification, Signature, Used_Codec)
TFO_REQ_L (System_Identification, Signature, Used_Codec, Codec_List).
TFO_REQ_P (System_Identification, Signature, Preferred_Codec, Codec_List)

The TFO_REQ Messages conform to the IS_REQ Message, defined in the Annex A, with IS_System_Identification set accordingly, followed by the TFO_Req_Extension_Block and optionally by the Codec_List_Extension_Block.

TFO_REQ takes 140 ms for transmission, see Figure 2. TFO_REQ_L and TFO_REQ_P take 180 ms for transmission.

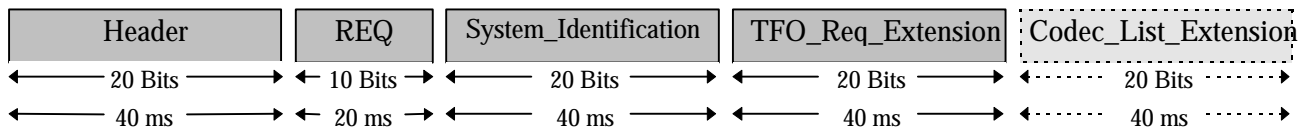


Figure 2: Construction of the TFO_REQ Messages

8.1.1 Definition of the TFO_Req_Extension_Block

The TFO_Req_Extension_Block consists of 20 bits, as defined in table 14.

Table 14: TFO_Req_Extension_Block

| Bit | Description | Comment |
|--------------|--|--|
| Bit 1 | "0" | normal IS-Message Sync Bit, constant. |
| Bit 2 | Req_Ident Req_Ident == "0" Req_Ident == "1" | Identifies the TFO_Req_Extension_Block REQ or REQ_L: Codec Field identifies the "used" Codec REQ_P: Codec Field identifies the "preferred" Codec |
| Bit 3..10 | Signature | An 8-bit random number to facilitate the detection of circuit loop back conditions and to identify the messages source |
| Bit 11 | "0" | normal IS-Message Sync Bit, constant |
| Bit 12.. 15: | Codec Codec == "0.0.0.0" Codec == "0.0.0.1" Codec == "0.0.1.0" | Identifies the Codec, which is currently used (Req_Ident == "0") or which is preferred (Req_Ident == "1") by the sender ANSI-136 Vselp (7.95 kBit/s) / ANSI-96-C Q8 ¹ ANSI-136 Acelp (7.40 kBit/s) / IS-127 EVRC ² ANSI-136 US1 (12.2 kBit/s) / IS-733 Q13 ³ |
| Bit 16..18: | CRC | CRC protecting Req_Ident, Signature and Codec, see 6.1.2 |
| Bit 19..20: | EX EX == "0.0" EX == "1.1" | The normal 2 bits for IS_Message Extension. REQ: No other extension block follows REQ_L or REQ_P: The Codec_List-Extension Block follows |

8.1.2 Cyclic Redundancy Check

The Cyclic Redundancy Check (CRC) is operating on the 15 bits consisting of Bit 1..15.

These 3 CRC bits are generated according to a degenerate (shortened) cyclic code using the generator polynomial:

$g(D) = D^3 + D + 1$. The encoding of the cyclic code is performed in a systematic form which means that, in GF(2), the polynomial: $d(m)D^{15} + d(m+1)D^{14} + \dots + d(m+12)D^3 + p(0)D^2 + p(1)D + p(2)$, where $p(0)$, $p(1)$ and $p(2)$ are the parity bits, when divided by $g(D)$, yields a remainder equal to: $1 + D + D^2$.

8.1.3 Definition of the Codec_List_Extension_Block

The Codec_List Extension Block consists of 20 bits, as defined in Table 15. It identifies the Codecs that are supported by the sender, respectively the BSS subsystem, including the mobile station and the radio resource, at sender side. The Codec_List must at least contain the Local_Used_Codec.

¹ Depending on system identification

² Depending on system identification

³ Depending on system identification

Table 15: Codec_List Extension Block

| Bit | Description | Comment |
|--------------|--------------------------|--|
| Bit 1 | "0" | normal IS-Message Sync Bit, constant. |
| Bit 2..10 | Codec_List_1 | First part of Codec_List. For each Codec one bit is reserved. If the bit is set to "0" then the specific Codec is not supported; if the bit is set to "1" then the specific Codec could be used. Bit 2: ANSI-136 Vselp / ANSI-96-C Q8 ⁴ Bit 3: ANSI-136 Acelp / IS-127 EVRC ⁵ Bit 4: ANSI-136 US1 / IS-733 Q13 ⁶ The remaining bits are reserved for future Codecs. |
| Bit 11 | "0" | normal IS-Message Sync Bit, constant |
| Bit 12.. 15: | Codec_List_2 | Second part of the Codec_List All four bits are reserved for future Codecs |
| Bit 16..18: | CRC | A 3-bit CRC protecting the Codec_List fields, see 6.1.2 |
| Bit 19..20: | EX EX == "0.0" | The normal 2 IS_Message Extension bits. No other extension block follows. |

8.2 Definition of the TFO_ACK Messages

Symbolic Notation: TFO_ACK (System_Identification, Reflected_Signature, Used_Codec)
TFO_ACK_L (System_Identification, Reflected_Signature, Used_Codec, Codec_List)

The TFO_ACK Messages conform to the IS_ACK Message, defined in the Annex A, with IS_System_Identification set accordingly, followed by the TFO_Ack_Extension_Block and optionally the Codec_List_Extension_Block.

TFO_ACK takes 140 ms for transmission, see Figure 3. TFO_ACK_L takes 180 ms for transmission.

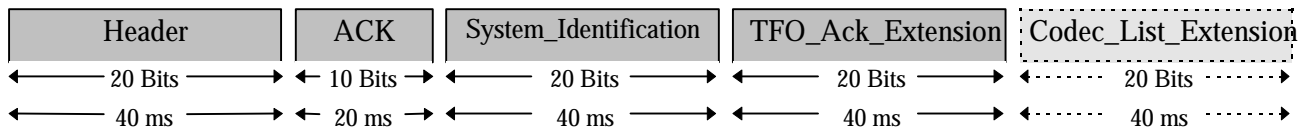


Figure 3: Construction of the TFO_ACK Message

8.2.1 Definition of the TFO_Ack_Extension_Block

The TFO_Ack_Extension_Block consists of 20 bits defined as (MSB := Bit 1) in table 16.

Table 16: TFO_Ack_Extension_Block

| Bit | Description | Comment |
|--------------|--|---|
| Bit 1 | "0" | normal IS-Message Sync Bit, constant. |
| Bit 2 | Ack_Ident Ack_Ident == "0" Ack_Ident == "1" | Identifies the TFO_Ack_Extension_Block ACK: Acknowledge to a received TFO_REQ Message reserved |
| Bit 3..10 | Signature | An 8-bit number containing the received Signature, <i>reflected</i> back. |
| Bit 11 | "0" | normal IS-Message Sync Bit, constant |
| Bit 12.. 15: | Codec | Identifies the Codec, which is currently used by the sender; see TFO_Req_Extension block. |
| Bit 16..18: | CRC | CRC protecting the Ack_Ident, Reply and Codec fields, see 6.1.2 |
| Bit 19..20: | EX EX == "0.0" EX == "1.1" | The normal 2 bits for IS_Message Extension. ACK: No other extension block follows ACK_L: The Codec_List_Extension Block follows |

⁴ Depending on System Identification

⁵ Depending on System Identification

⁶ Depending on System Identification

8.3 Definition of the TFO_TRANS Messages

Symbolic Notation: TFO_TRANS (Channel_Type).

Two TFO_TRANS Messages are defined in conformity to the IS_TRANS Messages in Annex A.

For 16 kbit/s submultiplexing channels the "TFO_TRANS (16k)" is used and is identical to "IS_TRANS_2_u".

TFO_TRANS takes 100 ms for transmission.

In most cases the respective TFO_TRANS Message shall be sent twice: once as a regular TFO Message, exactly before any series of TFO Frames, and once embedded into the first TFO Frames, see clause 10.

8.4 Definition of the TFO_NORMAL Message

Symbolic Notation: TFO_NORMAL.

The TFO_NORMAL Message is identical to the IS_NORMAL Message defined in the Annex A.

It shall be sent at least once whenever an established tandem free operation need to be terminated in a controlled way.

TFO_NORMAL takes 100 ms for transmission.

8.5 Definition of the TFO_FILL Message

Symbolic Notation: TFO_FILL.

The TFO_FILL Message is identical to the IS_FILL Message, defined in the Annex A.

TFO_FILL may be used to pre-synchronise IPEs. Since IS_FILL is one of the shortest IS Messages, this is the fastest way to synchronize IPEs, without IPEs swallowing other protocol elements. By default three TFO_Messages shall be sent at the beginning; this number may be, however, configuration dependent.

One TFO_FILL takes 60 ms for transmission.

8.6 Definition of the TFO_DUP Message

Symbolic Notation: TFO_DUP

The TFO_DUP Message is identical to the IS_DUP Message, defined in Annex A.

TFO_DUP informs the distant TFO Partner, that TFO Frames have been received unexpected, e.g. during Establishment.

This enables a fast re-establishment of TFO after a *local* HO.

TFO_DUP takes 60 ms for transmission.

8.7 Definition of the TFO_SYL Message

Symbolic Notation: TFO_SYL

The TFO_SYL Message is identical to the IS_SYL Message, defined in Annex A.

TFO_SYL informs the distant TFO Partner, that tandem free operation has existed, but suddenly no TFO Frame were received anymore. This enables a fast re-establishment of TFO after a *distant* HO.

TFO_SYL takes 60 ms for transmission.

9 Time Alignment of TFO Frames and TFO Messages

Any time alignment procedures which may be implemented on the TTL interface, are not affected by the TFO procedures on the PSTN interface.

TFO Frames and embedded TFO Messages are always exactly aligned with each other and follow the uplink TRAU Frames with a small, negligible, constant delay (Tultfo: some PCM samples).

9.1 Time Alignment of TFO Messages

At start up of the TFO Protocol the first regular TFO Message is aligned to an uplink TRAU Frame in the same way as a TFO Frame, respectively an embedded TFO Message would be aligned (see subclause 7.2). Then, after that, all regular TFO Messages follow contiguously, without any phase shift in time alignment, until the first TFO Frame needs to be sent (in general after the TFO_TRANS Message). Then the necessary number of T_Bits (if any) is inserted before the first TFO Frame, see subclause 7.2. Consequently all following, embedded TFO Messages are always aligned with the TFO Frames in a way, that the first bit of any TFO Messages is placed into the LSB of the first sample of a TFO Frame. Due to this definition, embedded TFO Messages only modify some of the synchronisation bits of the TFO Frames (the ones underlined and light shaded in column m=1 of Table 1 to 13 **Error! Reference source not found.**).

9.2 Time Alignment of TFO Frames

The contents of the Uplink TRAU Frame, received from the TRX via the TTL Interface, undergo the small, constant delay (Tultfo) required to perform the modifications of Sync bits, before being forwarded to the other TRAU over the PSTN Interface as TFO Frame. Since this delay is substantially smaller than the delay for the decoded speech signal, the TFO Frames precede the corresponding speech samples. Figure 4 shows the relations. Please note that no exact delay value for Tultfo is defined or need to be defined.

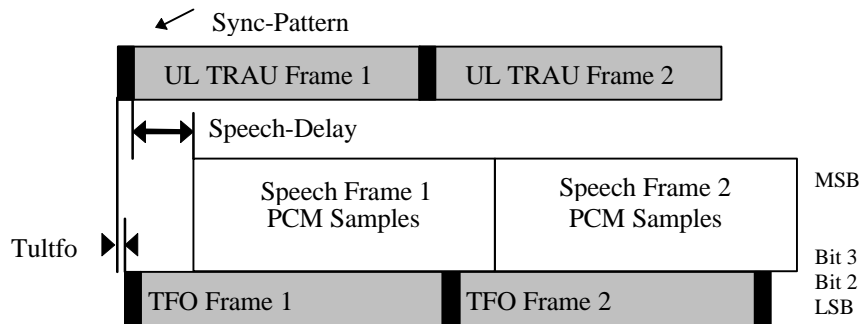


Figure 4: Uplink TFO Frame Time Alignment

On the transition between the sending of regular TFO Messages and the first TFO Frame on the PSTN interface, a sufficient number (up to a maximum of 159) of Time Alignment Bits, also called "T_Bits", are inserted into the LSBs of the PCM samples to align the TFO Frame as described above.

This insertion of Time Alignment Bits (if necessary) is started exactly with the 16th PCM sample after the last bit of the last regular TFO Message (i.e. the TFO_TRANS Message).

Whenever, in a later stage, the phase of the uplink TRAU Frame changes, then again T_Bits need to be inserted between two consecutive TFO Frames or deleted from the tail of the last TFO Frame to ensure proper alignment.

The insertion of T_Bits as a result of timing changes shall occur *between* TFO Frames and not within TFO Frames.

If the time alignment is necessary while a TFO Message is embedded into a series of TFO Frames, then the TFO Message may be cut into two parts with the T_Bits in between. Therefore, whenever an adjustment of the phase of the TFO Frames is necessary, then one additional TFO Message shall be embedded into the next TFO Frames (after the possibly ongoing TFO Message). If nothing else is to be transmitted, then the TFO_FILL Message shall be used. One TFO_TRANS Message is *always* embedded into the first TFO Frames. See the following Figure 5:

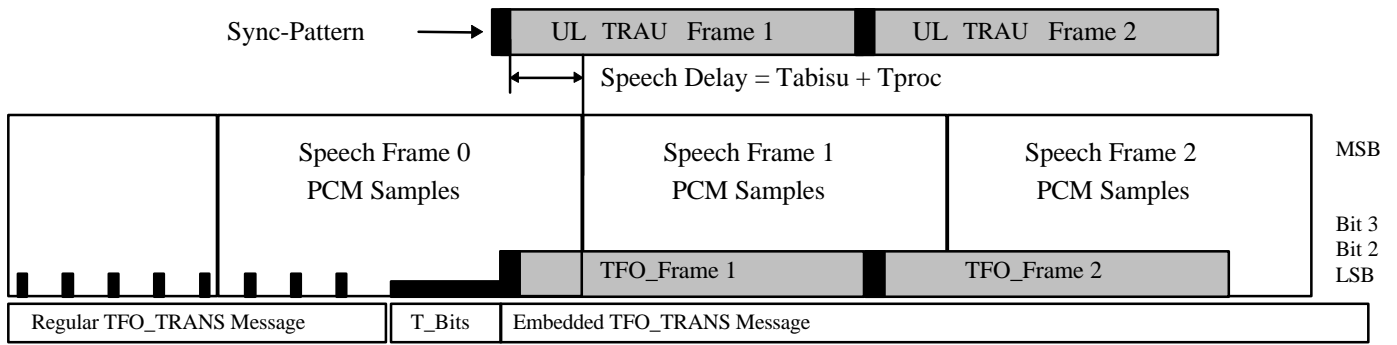


Figure 5: Time Alignment by inserting T_Bits and embedding one TFO_TRANS Message

9.3 Time Alignment of TFO Frames to Downlink TRAU Frames

The phase position of the downlink TRAU Frames is not affected by the TFO Protocol.

The phase difference between the received TFO Frames and the downlink TRAU Frames is in general constant, but arbitrary between 0 and 159 PCM samples. The time alignment of the TFO Frames to the downlink TRAU Frames must therefore be managed by buffering the TFO Frames within the receiving downlink TRAU. This can be done in one of two methods:

The received TFO Frame is buffered for a period between 0 to 159 PCM samples in addition to the processing delay (T_{bfh}) required to perform a suitable Bad Frame Handling on parameter level. Transmission of the downlink TRAU Frame may in this case begin *prior* to receipt of the complete TFO Frame.

NOTE 1: In this first method the overall one way signal delay will be between 30 ms and 10 ms lower than the delay in normal tandem connections.

Alternatively the received TFO Frame is buffered for a period between 160 to 319 PCM samples in addition to the processing delay required to perform a suitable Bad Frame Handling on parameter level (T_{bfh}). Transmission of the downlink TRAU Frame will in this case always begin *after* the receipt of the complete TFO Frame.

NOTE 2: In this second method the overall one way signal delay will always be up to 10ms lower or up to 10 ms higher than the delay in normal tandem connections.

NOTE 3: The two methods differ in one way signal delay always by exactly 20 ms. Figure 6 highlights the relations for an arbitrarily selected relative phase difference between TFO and TRAU Frames of 80 samples (10 ms). T_{bfh} is in the order of some PCM samples only, if error concealment is done "in advance" based on the parameters of the previous TFO Frame, before the actual TFO Frame is even received.

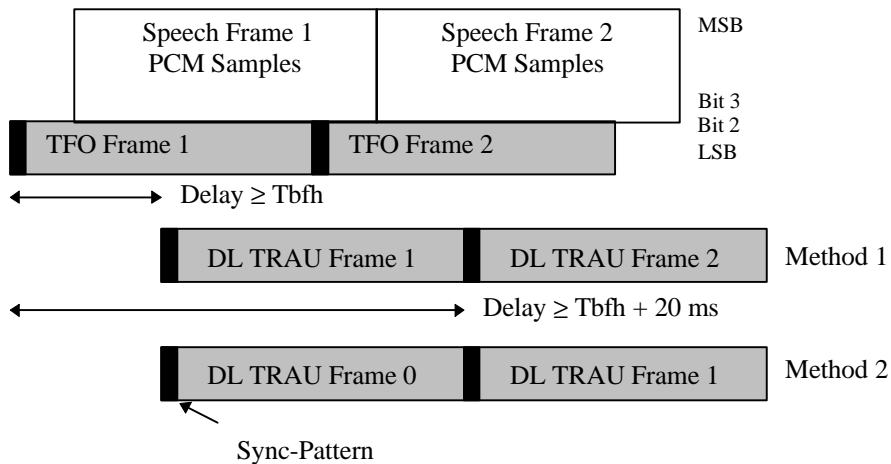


Figure 6: Downlink Time Alignment of TFO Frames

Please note that the repacking from TFO to TRAU Frames is depending on the TRAU Frame format used on the TTL interface. This format is manufacturer dependent and the repacking may involve higher delays due to bit reordering.

10 Processes for Tandem Free Operation

The following chapters describe the actions within the TRAU to establish and maintain Tandem Free Operation in terms of a State Machine, respectively TFO Processes, handling synchronisation and protocol. The description of the TFO Protocol does not reflect implementation details for the I/O Processes, but they may need to be considered for the exact understanding of the behaviour. Only the TFO_Protocol Process is detailed, which is responsible for the handling of the TFO Protocol.

The TFO_TRAU can be regarded as consisting of five processes, which are strongly coupled to each other, which run in parallel, but phase shifted. The TFO_Protocol Process communicates with the TFO I/O processes and, optionally, with its corresponding process within the MSC to resolve Codec Mismatch, see figure 7.

Under normal circumstances (exceptions occur during time alignments or octet slips) all TFO I/O Processes are triggered every 160 samples or every TRAU Frame of 20 ms. All events and actions are quantized in time into these smallest intervals.

It can be assumed that the processing times for the TFO Processes are very short and negligible.

However, it must be ensured that no timing ambiguity occurs between the Processes.

This means the processing and exchange of information between them do not overlap in time. Care must be taken especially when time alignment occurs, which may be completely independent in uplink and downlink.

During these time alignments the TFO Frames or TFO Messages may shift in time and consequently the triggering point for the related TFO Processes changes, too.

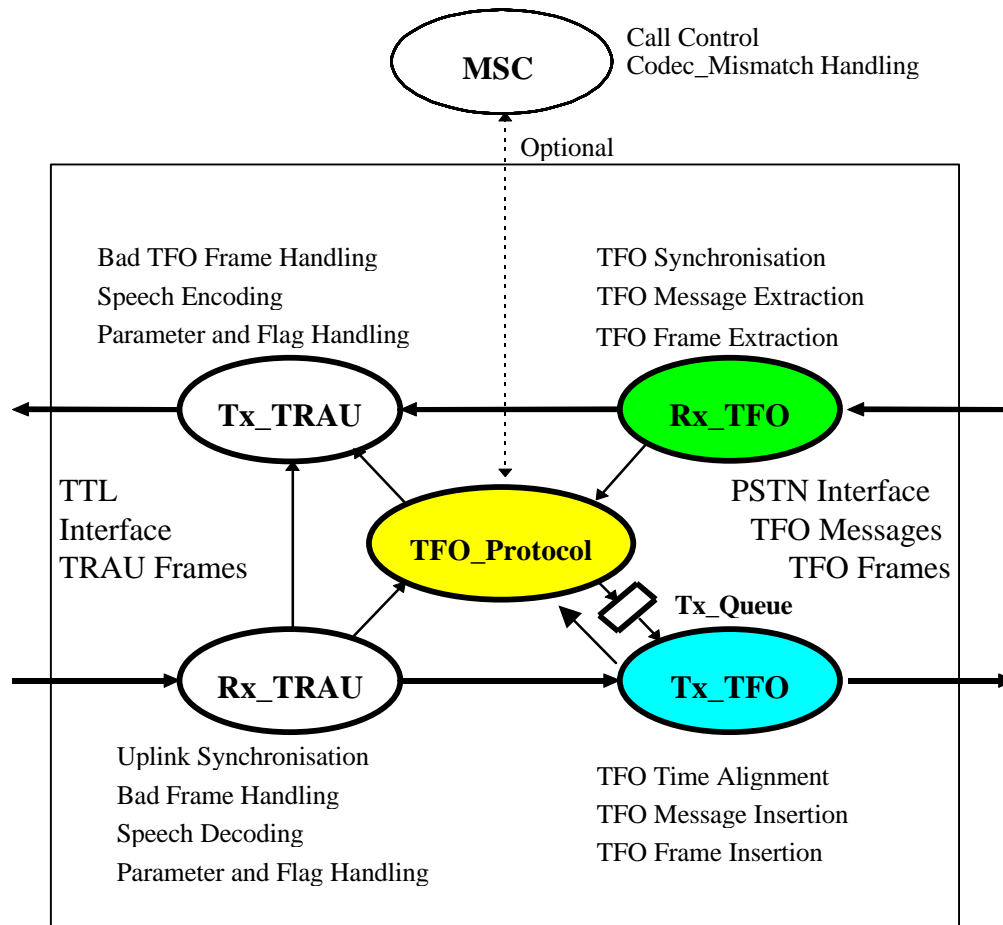


Figure 7: The TFO_TRAU consists of five Processes

10.1 Rx_TRAU Process

The Rx_TRAU Process receives Uplink TRAU Frames from the TTL Interface and synchronises to them, i.e. checks correct synchronisation and contents. It performs all actions of a conventional Uplink TRAU, e.g. calls the Bad Frame Handler and the Speech Decoder.

The resulting speech samples are handled to the Tx_TFO Process for output to the PSTN interface. In addition Rx_TRAU passes the Uplink TRAU Frames directly and unaltered to Tx_TFO.

10.2 Tx_TRAU Process

The Tx_TRAU Process builds autonomously the relevant Downlink TRAU Frames and sends them in the correct phase relation onto the TTL-Interface as required by the TRX.

Tx_TRAU has two major States: TFO == OFF (default at beginning) and TFO == ON, see Figure 8.

Toggleing between these two States is commanded by TFO_Protocol with Accept_TFO, respectively Ignore_TFO.

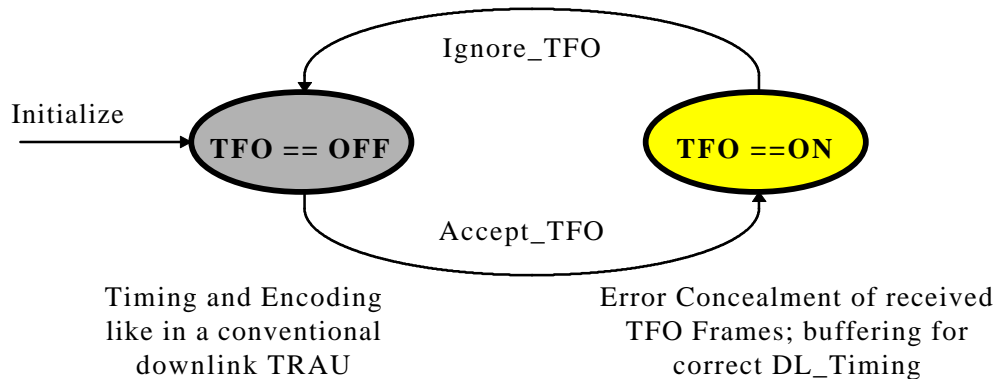


Figure 8: States of the Tx_TRAU Process

During TFO == OFF, Tx_TRAU performs all actions of a conventional downlink TRAU, as described in the appropriate standard (ANSI-95, ANSI-136). It samples one frame of speech samples in the correct phase position and calls the Speech Encoder. The resulting speech parameters are then transmitted downlink on the TTL interface.

During TFO == ON, Tx_TRAU performs Bad Frame Handling (and Comfort Noise Parameter Handling on parameter level in the case of ANSI-136) on the received TFO Frames, if necessary. The resulting speech parameters and control bits are buffered until they are passed as Downlink TRAU Frames in correct phase position to the TRX (see also subclause 7.3).

For ANSI-136, there are four possible cases regarding DTX in a Mobile-to-Mobile communication, as reflected in Table 17.

Table 17: DTX configurations in Mobile-To-Mobile communications

| Case | Local TRAU: Downlink | Distant TRAU: Uplink |
|------|----------------------|----------------------|
| 0 | No-DTX | No-DTX |
| 1 | No-DTX | DTX |

10.2.1 Downlink Speech Transmission if TFO is ON

During TFO == ON, the Tx_TFO Process receives TFO Frames from the PSTN Interface. It synchronises to them, i.e. checks correct synchronisation and contents. It extracts the data bits and calls, if appropriate (e.g. if ECI == "1" or if the TFO Frame is not-valid, see subclause 8.4.2), a Bad Frame Handler to derive suitable parameters for Downlink TRAU Frames. This Bad Frame Handler on parameter level is subject to manufacturer dependent future improvements and is not part of this recommendation.

10.2.2 DTX handling for ANSI-136

During TFO == ON and if distant Uplink DTX is active, (case 1 in Table 17), then the Rx_TFO Process receives TFO Frames containing speech parameters (SID == "0": handling as in case 0, see above), but also TFO Frames containing SID parameters and TFO Frames marked with ECI == "1" during speech inactivity. Tx_TRAU then calls a Comfort Noise Generator to derive suitable "speech" parameters for Downlink TRAU Frames. The SP flag shall always be set to SP = "1". The Downlink TRAU Frames shall not contain the SID codeword, but parameters that allow a direct decoding. Also this Comfort Noise Generator on parameter level is subject to manufacturer dependent future improvements and is not part of this recommendation.

10.2.3 Synchronisation and Bit Errors in Received TFO Frames

If Rx_TFO detects an error in the received TFO Frame synchronization or control bits or if a CRC error is detected, and the error is detected **prior** to beginning the output of the same frame (as a Downlink TRAU Frame), then Tx_TRAU shall either substitute parameters from the last good TFO Frame, or shall encode the received PCM samples for the duration of this frame.

If Rx_TFO detects an error in the received TFO Frame synchronization or control bits or if a CRC error is detected, and the error is detected **after** beginning of the output of the same frame (as a Downlink TRAU Frame), then Tx_TRAU shall deliberately corrupt the remaining, still unsent synchronization bits by setting them all to "0" and deliberately shall corrupt the remaining CRC bits. The effect of the frame error will subsequently be masked by the Mobile station's handling of bad frames.

10.3 Tx_TFO Process

The Tx_TFO Process gets directly the unaltered Uplink TRAU Frames (containing the speech parameters and the control bits) and the decoded speech PCM samples from Rx_TRAU. It further gets internal messages (commands) from TFO_Protocol via the Tx_Queue.

Tx_TFO has two major States: TFO == OFF (default at beginning) and TFO == ON, see Figure 9.

Toggleing between these two States is commanded by TFO_Protocol with Begin_TFO respectively Discontinue_TFO.

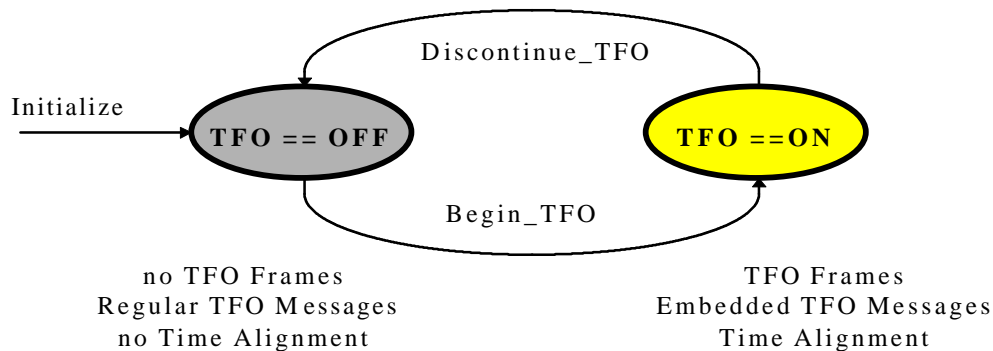


Figure 9: States of the Tx_TFO Process

During TFO == OFF, decoded speech PCM samples and regular TFO Messages (if any) are sent onto the PSTN interface. Time Alignment takes place only once, just before the beginning of the first regular TFO Message.

During TFO == ON, TFO Frames and embedded TFO Messages (if any) are sent. Time Alignment takes place just before the first TFO Frame and then whenever required in between two TFO Frames.

The Tx_TFO Process builds the relevant TFO Frames and sends them in the correct phase relation onto the PSTN Interface. Time alignment of TFO Messages and TFO Frames are handled autonomously and independent of the TFO_Protocol Process. Rx_TRAU informs Tx_TFO about any changes in the phase position of the Uplink TRAU Frame and Tx_TFO inserts automatically the correct number of T_Bits before the next TFO Frame, and embeds autonomously the next TFO_Message or a TFO_FILL Message, if necessary.

The TFO_Protocol Process can send internal messages into the **Tx_Queue** (First In, First Out). Tx_TFO shall take the message from the Tx_Queue one by one, shall process them autonomously and shall send the resulting TFO Messages in correct order and phase position, as regular or as embedded TFO Messages. Tx_TFO shall generate a Runout Message to TFO_Protocol, if the last TFO Message is sent without guarantee of a direct continuation by another TFO Message, i.e. if the (possible) IPEs may have run out of synchronisation (see chapter 10). TFO_Protocol may delete messages from Tx_Queue, as long as they are in there: Command "Clear Tx_Queue", at time *Tc*.

Basically, messages or commands that are already in processing by Tx_TFO at *Tc* can not be stopped, deleted or interrupted. The TFO Protocol is designed to work properly with that default handling, although not with fastest processing.

But: Tx_TFO shall investigate at *Tc*, how far the transmission of the current TFO Message is proceeded and shall **"Modify on the Fly"** this last TFO_Message before *Tc* into the first one after *Tc*, see Figure 10.

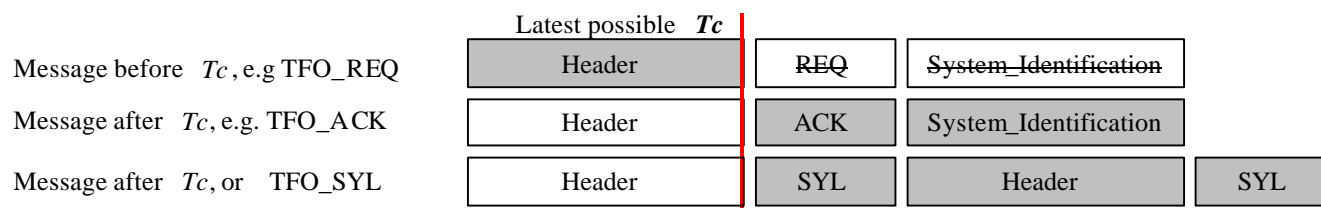


Figure 10: Modification on the Fly within the Header Transmission, examples

10.4 Rx_TFO Process

The Rx_TFO Process receives TFO Messages and TFO Frames from the PSTN Interface and synchronises to them, i.e. checks correct sync and contents. It bypasses all PCM samples and TFO Frames directly to Tx_TRAU for further processing. The Rx_TFO Process further extracts all the control bits and TFO Messages and sends corresponding Rx_TFO Messages to the TFO_Protocol Process.

10.4.1 Search for and Monitoring of TFO Synchronization

The monitoring of TFO Frame or TFO Message synchronisation shall be a continuous process. Typically, TFO Messages and TFO Frames follow each other with a well defined phase relation. Insertion of T_Bits or octet slips may, however, disturb that regular phase relation every now and then and shall be taken into account. In all error cases, the receiver shall investigate, if sync has been lost due to octet slip, phase adjustment or other events and shall try to resynchronize as fast as possible.

During Tandem Free Operation it is sometimes necessary to exchange TFO Messages by embedding them into the TFO Frame flow. This is indicated by a control bit (C5). Some of the TFO Frame synchronization bits are then replaced by bits of the TFO Message. TFO Messages follow specific design rules, which can be used to check if synchronisation is still valid.

The first TFO Message or the first TFO Frame (whatever comes first) shall be completely error free to be acceptable by Rx_TFO. After that all "valid" (see subclause 8.4.2) TFO Messages shall be reported to TFO_Protocol with a respective message. If a TFO Message has been received before and synchronisation is not found again for more than 60 ms, i.e. no "present" or "valid" TFO Message can be found during that time, then Rx_TFO shall generate a MSL (Message_Sync_Lost) Message to TFO_Protocol. A "not-valid", but "present" TFO Message shall not be reported, but also no MSL shall be reported, i.e. synchronisation is regarded as not lost, but the TFO Message is ignored. Similar, the first five "valid" TFO Frames shall be reported to TFO_Protocol with frame number n (n == 1,2,..5). Further valid TFO Frames do not need to be reported.

Similar, if the PCM channel is available for the first time, then a PCM_Channel_Available Message shall be sent to TFO_Protocol. Further on no more PCM_Channel_Available messages need to be reported.

If TFO Frame Synchronization is lost, or if too many errors are detected in TFO Frames (no present TFO Frames), then the Rx_TFO shall generate a FSL (Frame_Sync_Lost) Message to TFO_Protocol with frame number n (n == 1,2,3), the number of lost TFO Frames since the last valid TFO Frame. No more than three FSL Messages need to be reported in a series.

NOTE: The MSL and FSL Messages shall not be mixed up with the TFO_SYL Message, by which the distant TFO Partner reports lost synchronisation.

TFO Messages with Extension_Blocks that can not be understood by the receiving TRAU, but which follow the design rules for IS_Extension_Blocks, shall be ignored. This guarantees future expandability.

10.4.2 Errors in TFO Messages and TFO Frames

Some Definitions, which may serve as a guideline:

A TFO Message is called "**error-free**", if no error can be detected within the whole message.

A TFO Message is called "**single-error**", if no more than one bit position differs either in the IS_Header or the IS_Command_Block or the System_Ident_Block or the IPE_Mode_Block or the Sync bits and no errors are detectable within the CRC fields or the EX-fields.

A TFO Message may be regarded as "**correctable**", if the phase position is as in preceding TFO Messages and

- no more than 2 bit positions differ in the IS_Header; and
- no more than 1 error is detected within the IS_Command_Block; and
- no more than 3 errors are detected within the IPE_Mode_Block; and
- no more than 3 errors are detected within the System_Ident_Block; and
- no more than 1 error is detected within the Sync-Bit(s); and
- no more than 0 error is detected within the EX-field(s); and
- no more than 0 error is detected within the CRC-fields; and
- the total number of detected errors is not higher than 3.

TFO Messages, which are error-free, single-error or correctable are also called "valid" TFO Messages. All other TFO Messages are called "not-valid".

A TFO Message may be regarded as "**present**", if the phase position is as in preceding TFO Messages and

- no more than 4 bit positions differ in the IS_Header; and
- no more than 2 errors are detected within the IS_Command_Block; and
- no more than 3 errors are detected within the IPE_Mode_Block; and
- no more than 3 errors are detected within the System_Ident_Block; and
- no more than 2 errors are detected within the Sync-Bit(s); and
- no more than 1 error is detected within the EX-field(s); and
- no more than 1 error is detected within the CRC-fields; and
- the total number of detected errors is not higher than 5.

Sequences, which differ in more than "present" may not be recognized as TFO Messages at all ("**not-present**").

Note that the insertion of T_Bits may change the phase position of the TFO Frames and of bits of an embedded TFO Message. The TFO Message shall in that case be classified after the removal of the T_Bits.

An octet slip may also change the phase position of bits within a regular or embedded TFO Message.

If an error-free or a single-error TFO Message can be found after considering a hypothetical octet slip (± 1 sample), then it may be regarded as error-free or single-error and the new phase position shall be regarded as valid, if no valid or present TFO Message can be found at the old phase position.

A **TFO Frame** is called "**error-free**", if no error can be detected within the whole frame.

A TFO Frame is called "**single-error**", if no more than one bit position differs either in the synchronisation bits or the T_Bits and if no other errors can be detected. TFO Frames, which are error-free, or single-error are also called "**valid**" TFO Frames. All other TFO Frames are called "**not-valid**".

A TFO Frame may be regarded as "**present**", if

- no more than 4 bit positions differ in the synchronisation bits
- no more than 2 errors are detected within the T_Bits;
- no more than 1 error is detected within the control bits;
- no more than 1 error is detected within the CRC block; and
- the total number of detected errors is not higher than 5.

Sequences, which differ in more than "present" may not be recognized as TFO Frames at all ("**not-present**").

Note that the insertion or deletion of T_Bits may change the phase position of the TFO Frames. The TFO Frame shall in that case be classified after considering the T_Bits.

An octet slip may also change the phase position of bits within a TFO Frame. Typically a TFO Frame can not be corrected after an octet slip, but the next TFO Frame shall be found again.

The parameters of a valid TFO Frame shall be regarded as "bad", if the ECI flag is set to ECI == "1".

10.5 TFO_Protocol Process

The TFO_Protocol Process is typically invoked whenever a message is received, either from Rx_TRAU, Rx_TFO, Tx_TFO or the local MSC.

Two events are due to modifications of the local MS-MSC configuration,

- a modification of the used speech Codec (New_Local_Codec); and
- a modification of the list of the alternative speech Codecs (New_Local_Codec_List).

The New_Local_Codec is extracted from the uplink TRAU Frames and reported by Rx_TRAU.

The New_Local_Codec_List is reported by the MSC in a manufacturer dependent way.

It may happen during an established TFO connection that the used Codec is identified as not optimal. Then the distant partner (e.g. a TCME) may inform the TRAU by a TFO_REQ_P Message that another Codec would be preferred.

The TRAU has to inform the local MSC about the preferred Codec, but continues with TFO until an optional In_Call_Modification is performed by the local MSC.

10.5.1 Messages from Rx_TRAU or local MSC

| | |
|---|---|
| Rx == New_Speech_Call (Local_Used_Codec); | Rx_TRAU is activated by BS. |
| Rx == New_Local_Codec (New_Local_Used_Codec); | In Call Modification to other Codec Type. |
| Rx == Data_Call; | In Call Modification to Data_Call. |
| Rx == Local_Codec_List; | Manufacturer dependent, optional, from MSC. |
| Rx == TRAU_Idle; | Manufacturer dependent, either from TRX or MSC. |

10.5.2 Messages to Tx_TRAU

| | |
|------------------------|---|
| Tx_TRAU := Accept_TFO; | if TFO Frames are correctly received, they shall be used. |
| Tx_TRAU := Ignore_TFO; | TFO Frames, even if received correctly, shall be ignored. |

10.5.3 Optional Messages to the local MSC

MSC := TFO (Distant_Used_Codec, Distant_Codec_List, Distant_Preferred_Codec, ...).

10.5.4 Messages to and from Tx_TFO

The symbol () indicates that these Messages contain parameters, see clause 6.

| | |
|------------------------|--|
| Tx := TFO_REQ (); | main TFO_REQ Message. |
| Tx := TFO_ACK (); | main TFO_ACK Message, response only to TFO_REQ. |
| Tx := TFO_REQ_L (); | used in Mismatch, Operation and Periodic_Retry to inform about alternative Codecs. |
| Tx := TFO_ACK_L (); | response only to TFO_REQ_L. |
| (Tx := TFO_REQ_P ()); | undefined for TRAU, defined only for TCME. |
| Tx := TFO_TRANS (); | command IPEs to go transparent. |
| Tx := TFO_NORMAL; | reset IPEs into their normal operation. |
| Tx := TFO_FILL; | mainly to pre-synchronise IPEs. |
| Tx := TFO_DUP; | "I receive TFO Frames in Establishment". |
| Tx := TFO_SYL; | "I lost TFO Frame synchronisation". |
| Tx := Begin_TFO; | Insert TFO Frames from now on. |
| Tx := Discontinue_TFO; | Discontinue inserting TFO Frames. |
| Clear Tx_Queue; | Clears all remaining commands from Tx_Queue. |
| Rx == Runout; | Reports that the continuous stream of outgoing TFO Messages may be interrupted. |

10.5.5 Messages from Rx_TFO

The symbol () indicates that these Messages contain parameters, see clause 6.

| | |
|------------------|---|
| Rx == TFO_REQ () | Request from potential distant TFO partner. |
|------------------|---|

Rx == TFO_ACK () TFO acknowledge from distant TFO partner.
Rx == TFO_REQ_L () Local Codec and Codec list in the case of Codec mismatch.
Rx == TFO_ACK_L () Response to TFO_REQ_L Message.
Rx == TFO_REQ_P () Requests an other, preferred Codec, plus Codec_List.
Rx == TFO_TRANS () May serve as alternative TFO_ACK in some cases!
Rx == TFO_NORMAL Command IPEs to normal operation.
Rx == TFO_FILL In general no specific meaning.
Rx == TFO_DUP Distant partner has changed.
Rx == TFO_SYL Distant partner doesn't receive any more TFO messages.
Rx == TFO_Frame () TFO_Frame (Distant_Used_Codec; Number_of_Received_Frames).
Rx == Frame_Sync_Lost () Frame_Sync_Lost (Number_of_Lost_Frames).
Rx == Mess_Sync_Lost Message_Sync_Lost.
Rx == PCM_Channel_Available at the beginning of a period where a PCM channel on the PSTN interface is available.
The message "TFO_Frame ()" needs to be sent only at the first five occurrences, either after a not valid TFO Frame, or if the Distant_Used_Codec changed.
The message "Frame_Sync_Lost ()" needs to be sent only at the first five occurrences of errors in TFO Frames or loss of synchronisation, after a correctly received TFO Frame.
The message "Mess_Sync_Lost" is sent, when after a valid TFO Message no following TFO Message is found.

10.5.6 Messages from higher system level (CDMA only)

By default a TRAU will attempt TFO. This behaviour can be altered by commands from a higher system level, which may be implemented using A1-Messages.

- Inhibit_TFO This message is sent from higher system level to explicitly force the TRAU into tandem mode. It is only sent after the PCM channel is available. If the state of the PSTN interface can not be evaluated from a higher system level, and is received prior to availability of the PCM channel, the message shall be queued until the PCM channel is available.
- Allow_TFO This message is sent from a higher system level to enable TFO negotiation in the TRAU.

11 State Machine of the TFO_Protocol Process for CDMA

The TFO_Protocol Process can be described by a State Machine, consisting of 11 States: six main States with several sub-States, see figure 11:

- Initialisation (• Not_Active, • Wakeup).
- Establishment (• First_Try, • Continuous_Retry, • Periodic_Retry, • Monitor, • Mismatch).
- Contact (• Contact).
- Konnect (• Konnect).
- Operation (• Operation).
- Tandem (• Tandem)

It is assumed that Events (Conditions checking, Actions and Transition to an other State) are handled almost instantaneous and in any case significantly shorter than the time required to complete the transmission of any one TFO Message or TFO Frame.

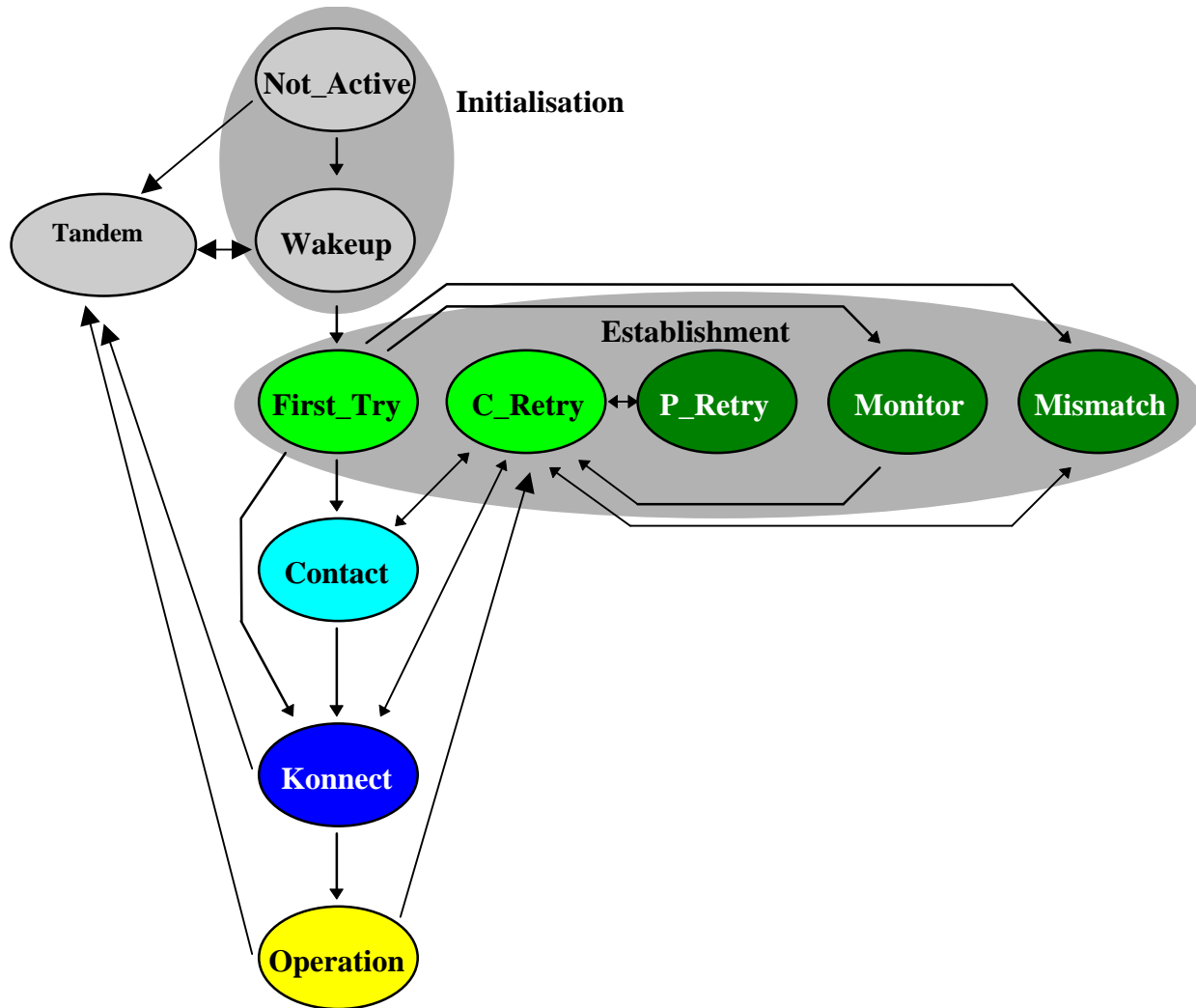


Figure 11: TFO_Protocol State Machine with most important transitions

11.1 Initialisation

11.1.1 Not_Active State

The state Not_Active is standard and manufacturer dependent and can not be described in detail here. The TRAU basically performs no speech coding, but may monitor the link for certain patterns (e.g. Idle patterns). The TRAU may also be in Data mode, which is also not described further, but is handled here as "Not_Active".

11.1.2 Wakeup State

The Wakeup State is entered, when the TRAU is activated by the appropriate event (e.g. activation by a MSC message or reception of TRAU Frames). The Wakeup state is characterized by the TRAU sending speech data at least in one direction.

The transition to Establishment is performed, if speech processing is performed in both. This is the point in time where the time out for TFO Messages starts, i.e. a maximum number of TFO_REQ Messages shall be sent after that.

11.2 Establishment

The Establishment submits several slightly different situations:

| | |
|------------------|---|
| First_Try | when the TRAU just has started; it sends TFO_REQ Messages continuously; |
| Continuous_Retry | when Contact to a TFO Partner has existed, but was interrupted recently; |
| Periodic_Retry | when Contact to a TFO Partner had existed, but was interrupted some time ago; |
| Monitor | when no TFO partner could be found, but the TRAU continues to monitor the PSTN Interface; |
| Mismatch | when a TFO partner with a different Codec has been identified. |

Loopback is a specific situation, when the call is still not through connected and the TRAU receives the own sent signals. No specific State is allocated to describe this situation. Instead, loopback is handled in First_Try and Continuous_Retry. Common to all these situations is that the TRAU does not know, if there is a distant TFO partner and/or if the links are digitally transparent. Typically, TFO_REQ Messages are sent and expected.

11.2.1 First_Try State

The TRAU sends and receives PCM samples on and from the PSTN interface. Regular TFO_REQ Messages are sent onto the PSTN interface continuously for a certain maximum time. After that, if no TFO Partner answers before, Tx_TFO reports a Runout of TFO Messages, and TFO_Protocol enters automatically into the Monitor State.

If TFO_REQ Messages are received with the same, own Signature, then a circuit loop back is assumed, i.e. the call is still not through-connected. The TRAU selects a new Signature and continues sending TFO_REQ Messages, until a different Signature is received. Since loop back delays may be substantial in some cases, the TRAU has to remember and compare also the previously selected own Signature. Care has to be taken that the Signature selection contains a true random element to avoid that two different TRAUs select by coincidence identical signatures again and again.

11.2.2 Continuous_Retry State

TFO Contact had existed, either by TFO Messages or by TFO Frames, but was interrupted and sync was lost. The TRAU sends a maximum number of regular TFO_REQ Messages continuously, to test, if TFO could be re-established. If Tx_TFO reports a Runout of TFO Messages, then the TFO_Protocol enters the Periodic_Retry State.

11.2.3 Periodic_Retry State

Entered from Continuous_Retry, TFO_Protocol tests from time to time by a single TFO_REQ_L, if TFO could be re-established. As soon as a TFO Message is received, TFO_Protocol leaves this State.

NOTE: Since no contiguous transmission of TFO Messages is ongoing, possible IPEs may be unsynchronized.

11.2.4 Monitor State

The TRAU monitors the PSTN interface for TFO Messages or TFO Frames, but it does not send TFO Messages or TFO Frames. As soon as a TFO Message from a distant partner (a TRAU or a TCME) has been received, the TRAU knows that a TFO Partner exists and it knows that the transmission path from the partner is digitally transparent.

The TRAU may already now see, whether TFO is possible, but it must ensure that all IPEs are synchronised. It therefore transits into the Continuous_Retry State. In case of Codec Mismatch, it terminates the TFO Protocol by sending TFO_REQ_L back, informs its local MSC and transits into Mismatch.

NOTE: Since no contiguous transmission of TFO Messages is ongoing, possible IPEs may be unsynchronized.

11.2.5 Mismatch State

From an previous contact it is obvious, that a distant TFO Partner exists, but the Codecs do not match.

The TRAU waits without sending TFO Messages or TFO Frames, if either the distant TFO Partner changes to the Local_Used_Codec, or the local MSC solves the Codec mismatch situation by an intra cell HO to the Distant_Used_Codec.

NOTE: Since no contiguous transmission of TFO Messages is ongoing, possible IPEs may be unsynchronized.

11.3 Contact State

There is a distant TFO Partner, which has sent TFO_REQ. The Codecs do match. The link from the distant partner is transparent. Now TFO_ACK need to be sent to check the transparency of the link to the distant partner.

As soon as a TFO_ACK or TFO_TRANS from a distant partner has been received, the TRAU knows that the links in both directions are digitally transparent. The TRAU sends TFO_TRANS to bypass the IPEs and starts sending TFO Frames. It transits into Konnect State.

11.4 Konnect State

The TRAU sends TFO Frames and possibly embedded TFO Messages as long as it receives correct TFO Messages.

The first received TFO Frame causes the transition into the final Operation State.

If no TFO Frames are received within a certain period, the TRAU transits to the Failure State.

11.5 Operation State

In this State - the main State of TFO_Protocol - the TRAU sends and receives TFO Frames, thus the TFO Connection is fully operating. TFO Messages may occur embedded into TFO Frames.

11.6 Tandem

This State is entered when either the distant partner shows an incorrect behaviour or a command from a higher system level has been received to enter this State. The TRAU then sends pure PCM samples onto the PSTN interface. It does not send TFO Frames or TFO Messages.

12 Detailed Description of TFO_Protocol for CDMA

The TFO_Protocol Process is always in one well defined State. An Event triggers Actions and a Transition into another State. The TFO Protocol is described in a table-wise manner, with a syntax as defined in Table 18.

Table 18: Definition of the Syntax for the State Machine Description

| | | | |
|-----------------|--|-----|--|
| Event: | <Received Message> | ... | <Other Event> |
| Number: | <running number> | | <running number> |
| Condition: | [<Condition>] & [<Condition>] | | [<Condition>] [<Condition>] |
| Comment: | [<Comment>] | | [<Comment>] |
| State: | | | |
| <Actual State>: | <Action Name>;[<Action Name>;] <Next State>; [<Comment>] | | <Action Name>;[<Action name>;] <Next State>; [<Comment>] |
| ... | | | |
| <Actual State>: | <Action Name>;[<Action Name>;] <Next State>; [<Comment>] | | <Action Name>;[<Action Name>;] <Next State>; [<Comment>] |

Several tables, table 21 to table 28, are necessary to describe the whole State Machine. If a table entry contains only a dashed line, it is assumed that this combination can not occur. If, however, such a combination is encountered during operation, it is treated as 'don't care'.

The Actions are described in Table 20, with a syntax as defined in table 19.

Table 19: Definition of Syntax for Action Table

| Name | Action List | Comment |
|---------------|-------------------------|-----------|
| <Action Name> | <Action >;[<Action >;] | <Comment> |
| ... | | |
| <Action Name> | <Action >;[<Action >;] | <Comment> |

Tx := TFO_REQ means, that TFO_Protocol places a command into Tx_Queue. Tx_TFO handles the details autonomously and generates a TFO_REQ Message for transmission over the PSTN interface, when it comes to that command.

Tx := 31*TFO_REQ means: put 31 TFO_REQ commands into Tx_Queue. Not necessarily all will really trigger TFO_REQ Messages. In most cases Tx_Queue will be cleared before. Similar definitions hold for the other messages. The Tx_Queue is a first_in_first_out command queue. It is filled by TFO_Protocol and read by Tx_TFO.

Clear Tx_Queue, means that all remaining commands are deleted from the Tx_Queue in that very moment (time T_c). Note that due to the duration time to transmit a TFO_Message completely, the TFO_Protocol Process is often already within an other State while still TFO Messages commanded in earlier States are within the Tx_Queue or under transmission.

MSC := TFO () means that a message is sent to the local MSC; similar

Tx_TRAU := ... means a message to Tx_TRAU.

An Event **TFO_REQ** means that a TFO_REQ Message was correctly received on the PSTN interface. The Rx_TFO Process has sent a message to TFO_Protocol, containing the new values for the respective variables. TFO_Protocol updates its variables with the new values. Similar definitions hold for the other messages.

One Timer **T := <Time_out>** is necessary to describe time out situations. The notation **T := DIS** means that the Timer is disabled. Positive values are decremented in an hidden background process in steps of 20 ms. When T gets to the value "0", then the TFO_Protocol process is invoked.

Local_Used_Codec (short form: **Luc**) means the type of speech Codec used in the local TRAU and.

New Local_Used_Codec (Nluc) refers to the new codec received in "In_Call_Modifications".

Distant_Used_Codec (Duc) means the type of speech Codec used by the distant partner, as reported in TFO_REQ... or TFO_ACK...

Distant_Preferred_Codec (DPC) means the type of speech Codec that the distant partner would prefer, as reported in TFO_REQ_P.

Local_Codec_List (LCL) means the list of all Codecs that could alternatively be used, i.e. which are supported by both the local MS and the local MSC. It always contains at least the Local_Used_Codec.

It is reported in TFO_REQ_L, TFO_ACK_L or TFO_REQ_P.

Distant_Codec_List (DCL) means the list of all Codecs that could alternatively be used, i.e. which are supported by the distant MS and the distant MSC. It always contains at least the Distant_Used_Codec.

All these variables are initialized to **UNKNOWN**, which means that the contents of the variables are not defined.

Local_Signature (Lsig) means the 8-bit random number in TFO_REQ, which identifies the local TFO_REQ Messages. It is also used in TFO_REQ_L.

Distant_Signature (Dsig) means the 8-bit random number as received in TFO_REQ, TFO_REQ_L and TFO_REQ_P, in TFO_ACK and TFO_ACK_L

If received in TFO_REQ, TFO_REQ_L and TFO_REQ_P, then it should be different to the Local_Signature, otherwise loop back must be assumed (exceptions exist).

If received in TFO_ACK or TFO_ACK_L, then it should be identical to the Local_Signature, otherwise the TFO_ACK is not a response to an own TFO_REQ respectively TFO_REQ_L, but maybe was created during a HO situation.

Local_Channel_Type (LCh) and **distant_Channel_Type (DCh)** refer to the 8 or 16 kBit/s transparent channel used by the local Tx_TFO respectively received by the distant TFO_TRANS.

Error protection and error handling: It is assumed that the defined error protection is strong enough for the error rates encountered on typical PSTN interface links. The few occurring errors are in practically all cases detected and possibly even corrected by Rx_TFO, before reported to TFO_Protocol. Therefore TFO_Protocol can rely on the correctness of the received Events. The protocol is, however, "selfhealing" and will handle the unlikely erroneous reported Events, too. The Event "**PCM_Channel_Available**" is given if in State Wakeup, if valid PCM samples are received on the PSTN interface. This event can be triggered by a message from a higher system level.

Fast HO handling: The defined protocol assumes that the new TRAU, to which the HO is performed, is already in State Wakeup before the PSTN Interface is switched to that TRAU. Only then the TFO Frames can be received by that TRAU and fast HO handling is possible.

Timing: If two Events occur by coincidence at the same time, then they shall be processed in the order given by the tables 20 to 27 (left to right). TFO Messages arrive always some time before the embedding TFO Frame and shall be handled therefore first.

Runout is the Event, when the last TFO Message has been taken from the Tx_Queue and the last 10 bits are going to be sent by Tx_TFO to the PSTN interface. So there is still some time for TFO_protocol to react and place a further TFO Message into Tx_Queue, which then shall be transmitted without gap to the messages before.

Table 20: Defined Actions

| Name | Actions | Comments |
|-------------|---|--|
| C | Clear Tx_Queue; T := DIS; | Initialise Tx_Queue and disable the timer |
| T1 | T := 1s; | Set Timeout of 1 second |
| T2 | T := 2s; | Set Timeout to 2 seconds |
| T5 | T := 5s; | Set Timeout to 5 seconds |
| NoAc | . | No Action required |
| S | Lsig := New_Random_Number; Old_Sig := UNKNOWN; | Generate new Signature and set Old_Sig to unknown; if no Loopback is assumed. |
| SO | Old_Sig := Lsig; Lsig := New_Random_Number; | Remember old Signature and generate a new Signature, if Loopback is assumed. |
| U | Old_Sig := UNKNOWN; | Reset Old_Sig before leaving FIT or COR |
| F | Tx := 3*TFO_FILL; | "Hello IPEs! Please synchronise!" |
| T | Tx := TFO_TRANS (); | "Hello IPEs! Please open a transparent channel!" |
| N | Tx := TFO_NORMAL; | "Hello IPEs! Please return to normal operation!" |
| REQ | Tx := 35*TFO_REQ; | "Hello Partner? Can You do TFO with me?" |
| ACK | Tx := 7*TFO_ACK; | "Yes, I can do TFO with You!" |
| L1 | Tx := TFO_REQ_L; | "Here is my Codec_List! Can you hear me?" |
| L | Tx := 6*TFO_REQ_L; | "Here is my Codec_List, please acknowledge!" |
| LA | Tx := TFO_ACK_L; | "Yes, I received Your Codec_List! Here is mine!" |
| BT | Tx := Begin_TFO; | Begin Transmission of TFO Frames |
| DT | Tx := Discontinue_TFO; | Discontinue Transmission of TFO Frames |
| IT | Tx_TRAU := Ignore_TFO; | Tx_TRAU works as conventional downlink TRAU |
| AT | Tx_TRAU := Accept_TFO; | Tx_TRAU bypasses TFO_Frames |
| B | MSC := TFO (); | "Hello MSC! Some news from the TFO_Scene!" |

Table 21: Call Setup and Loopback Handling

| Event: | New_Speech_Call | PCM_Channel Available | TFO_REQ | TFO_REQ |
|---|---|--|--|--|
| Number: | 24 | 29 | 0 | 0a |
| Condition: & | . | . | Duc==Luc Dsig==Lsig | Duc==Luc Dsig==Old_Sig |
| Comment: | Sytstem activates TRAU | PSTN-Interface gets active; occurs only at beginning | Loopback (LB), or distant handover (HO)? wrong Signature ? | Loopback (LB), or distant handover (HO)? |
| State: | | | | |
| NAC: Not_Active . | C;S;IT; WAK; typ. 1rst Event | ----- ----- . | ----- ----- . | ----- ----- . |
| WAK: Wakeup . | ----- ----- . | C;F;REQ; FIT; typ. 2nd Event | ----- ----- . | ----- ----- . |
| FIT: First_Try . | ----- ----- . | ----- ----- . | C;SO;REQ; FIT; LB! | NoAc; FIT; Ignore LB |
| COR: Continuous Retry . | ----- ----- . | ----- ----- . | C;SO;REQ; COR; LB!? | NoAc; COR; Ignore LB |
| PER: Periodic Retry . | ----- ----- . | ----- ----- . | C;F;S;ACK; CON; Dist. HO! | ----- ----- . |
| MON: Monitor . | ----- ----- . | ----- ----- . | C;F;S;REQ; FIT; Dist. HO! | ----- ----- . |
| MIS: Mismatch . | ----- ----- . | ----- ----- . | C;F;S;ACK; CON; Dist. HO! | ----- ----- . |
| CON: Contact . | ----- ----- . | ----- ----- . | C;SO;REQ; COR; safe way | ----- ----- . |
| KON: Konnect . | ----- ----- . | ----- ----- . | C;DT;SO;REQ;T1; COR; IPes transparent! | ----- ----- . |
| OPE: Operation . | ----- ----- . | ----- ----- . | C;DT;IT;SO;REQ; COR; | C;DT;IT;SO;REQ; COR; |
| TAN: Tandem . | ----- ----- . | ----- ----- . | NoAc; TAN; | ----- ----- . |

Table 22: Most Important Cases, Especially at Call Setup

| Event: | TFO_REQ | TFO_ACK | TFO_ACK | TFO_TRANS | TFO_FRAME |
|------------------------------------|---|---|---|--|--|
| Number: | 1 | 2 | 3 | 4 | 5 |
| Condition: & | Duc==Luc Dsig!=Lsig | Duc==Luc Dsig==Lsig | Duc==Luc Dsig!=Lsig | DCh==LCh . | Duc==Luc . |
| Comment: | Distant REQ | Distant ACK | Wrong Response | similar to ACK | one or more |
| State: | Good Signature . | Good Signature . | Handover? . | As response to loc ACK_? . | TFO Frames . |
| NAC: Not_Active . | ----- ----- . | ----- ----- . | ----- ----- . | ----- ----- . | ----- ----- . |
| WAK: Wakeup . | ----- ----- . | ----- ----- . | ----- ----- . | ----- ----- . | ----- ----- . |
| FIT: First_Try . | C;U;ACK; CON; typical | C;U;T;BT;T;T1; KON; typical; IPEs! | C;REQ; FIT; . | NoAc; FIT; wait f. Frames | C;REQ; FIT; restart |
| COR: Continuous Retry | C;U;ACK; CON; typical | C;U;T;BT;T;T1; KON; typical; IPEs! | C;REQ; COR; . | NoAc; COR; wait f. Frames | C;REQ; COR; 1: Call is back? |
| PER: Periodic Retry | C;F;ACK; CON; OK, Contact back | C;F;S;REQ; COR; rare case, test | C;F;REQ; COR; . | NoAc; PER; wait f. Frames | C;F;REQ; COR; 1: Call is back? |
| MON: Monitor . | C;F;REQ; FIT; IPEs? | C;F;S;REQ; FIT; Rare case, test | C;F;REQ; FIT; . | NoAc; MON wait f. Frames | C;F;REQ; FIT; 1: Call is back? |
| MIS: Mismatch . | C;F;ACK; CON; Mismatch resolved | C;F;S;REQ; COR; rare case, test | C;F;REQ; COR; . | NoAc; MIS; wait f. Frames | C;F;REQ; FIT; 1: Call is back? |
| CON: Contact . | C;ACK; CON; typical: wait | C;T;BT;T;T1; KON; typical: yes! | C;REQ; COR; . | C;T;BT;T;T1; KON; Fast way! | C;T;BT;T;T1; KON; Missed TRANS? |
| KON: Konnnect . | C;DT;REQ;T1; COR; IPEs transparent! | NoAc; KON; Typical: wait | NoAc; KON; . | NoAc; KON; typical: wait | AT;L;T2;B; OPE; typ: call setup |
| OPE: Operation . | C;DT;IT;REQ; COR; Dist. Back off | C;DT;IT;REQ; COR; Rare case, test | C;DT;IT;REQ; COR; Rare case, test | NoAc; OPE; typ in HO | NoAc; OPE; Main! TFO! |
| TAN: Tandem . | NoAc; TAN; . | NoAc; TAN; . | NoAc; TAN; . | NoAc; TAN; . | NoAc; TAN; . |

Table 23: In Call Modification and Handover

| Event: | New_Local_Codec | New_Local_Codec | TFO_SYL | TFO_DUP |
|---|---|---|---------------------------------------|-------------------------------------|
| Number: | 25 | 26 | 7 | 8 |
| Condition: & | Duc==Nluc . | Duc!=Nluc . | . . | . . |
| Comment: . State: | in Call Modif. Mismatch resolv. (Luc!=Nluc) | In Call Modif. Mismatch occurs! (Luc!=Nluc) | the dist. TRAU lost sync in OPE | the dist. TRAU recognised HO |
| NAC: Not_Active . | ----- ----- . | ----- ----- . | ----- ----- . | ----- ----- . |
| WAK: Wakeup . | NoAc; WAK; . | NoAc; WAK; . | ----- ----- . | ----- ----- . |
| FIT: First_Try . | C;REQ; FIT; restart | C;REQ; FIT; restart | NoAc; FIT; HO? Ignore | NoAc; FIT; HO? Ignore |
| COR: Continuous Retry . | C;REQ; COR; . | C;REQ; COR; . | NoAc; COR; ignore | NoAc; COR; ignore |
| PER: Periodic Retry . | L1;T5; PER; . | L1;T5; PER; . | C;F;REQ; COR; rare case, test | C;F;REQ; COR; rare case, test |
| MON: Monitor . | NoAc; MON . | NoAc; MON . | C;F;REQ; FIT; rare case, test | C;F;REQ; FIT; rare case, test |
| MIS: Mismatch . | C;F;REQ; COR; Mismatch res. | L;T2;B; MIS; Direct info. | C;F;REQ; COR; rare case, test | C;F;REQ; COR; rare case, test |
| CON: Contact . | ----- ----- . | C;REQ;B; COR; . | C;F;REQ; COR; rare case, test | C;F;REQ; COR; rare case, test |
| KON: Konnect . | ----- ----- . | C;DT;REQ;B; COR; . | C;DT;REQ; COR; . | C;DT;REQ; COR; . |
| OPE: Operation . | ----- ----- . | C;DT;IT;REQ;B; COR; . | C;DT;IT;REQ; COR; . | C;DT;IT;REQ; COR; . |
| TAN: Tandem . | ----- ----- . | NoAc; TAN; . | NoAc; TAN; . | NoAc; TAN; . |

Table 26: Mismatching TFO_TRANS and TFO Frames

| Event: | TFO_TRANS | TFO_FRAME | TFO_FRAME |
|---|----------------------------------|----------------------------------|--|
| Number: | 21 | 22 | 23 |
| Condition: & | DCh!=LCh . | Duc!=Luc n==1 | Duc!=Luc n>1 |
| Comment: State: | Mismatch of channel type . | Mismatch for one TFO Frame | Mismatch for at least two TFO Frames |
| NAC: Not_Active . | ----- ----- . | ----- ----- . | ----- ----- . |
| WAK: Wakeup . | ----- ----- . | ----- ----- . | ----- ----- . |
| FIT: First_Try . | C;U;L;T2;B; MIS; HO? | NoAc; FIT; HO? Be tolerant | C;U;L;T2;B; MIS; HO! Mismatch! |
| COR: Continuous Retry . | C;U;L;T2;B; MIS; . | NoAc; COR; Call Forw.? | C;U;L;T2;B; MIS; . |
| PER: Periodic Retry . | C;F;L;T2;B; MIS; . | NoAc; PER; Call Forw.? | C;F;L;T2;B; MIS; . |
| MON: Monitor . | C;F;L;T2;B; MIS; . | NoAc; MON Call Forw.? | C;F;L;T2;B; MIS; . |
| MIS: Mismatch . | C;L;T2;B; MIS; . | NoAc; MIS; Call Forw.? | C;L;T2;B; MIS; . |
| CON: Contact . | C;L;T2;B; MIS; . | NoAc; CON; . | C;L;T2;B; MIS; . |
| KON: Konnect . | C;DT;L;T2;B; MIS; . | NoAc; KON; . | C;DT;L;T2;B; MIS; . |
| OPE: Operation . | NoAc; OPE; ignore? | NoAc; OPE; Hard HO? | C;DT;L;T2;IT;B; MIS; Hard HO into TFO |
| TAN: Tandem . | NoAc; TAN; . | NoAc; TAN; . | NoAc; TAN; . |

Table 27: Local Events, Call Termination

| Event: | New_L_Codec_List | Data_Call | TRAU_Idle | TFO_FILL | TFO_NORMAL |
|------------------------------------|---------------------------------------|--|--|---------------------------|----------------|
| Number: | 30 | 27 | 28 | 37 | 33 |
| Condition: & | . | . | . | . | . |
| Comment: | from MSC | in Call Modif. stop TFO | Internal System Command to Reset TRAU | ignore, is just Filler | ignore |
| State: | . | . | . | . | . |
| NAC: Not_Active | NoAc; NAC; | NoAc; NAC; | NoAc; NAC; | ----- ----- | ----- ----- |
| WAK: Wakeup | NoAc; WAK; | NoAc; NAC; | NoAc; NAC; | ----- ----- | ----- ----- |
| FIT: First_Try | NoAc; FIT; update loc. Par. | C;N; NAC; | C;N; NAC; | NoAc; FIT; | NoAc; FIT; |
| COR: Continuous Retry | NoAc; COR; | C;N; NAC; | C;N; NAC; | NoAc; COR; | NoAc; COR; |
| PER: Periodic Retry | NoAc; PER; | C;N; NAC; | C;N; NAC; | NoAc; PER; | NoAc; PER; |
| MON: Monitor | NoAc; MON | C;N; NAC; | C;N; NAC; | NoAc; MON | NoAc; MON |
| MIS: Mismatch | C;L;T2; MIS; direct info | C;N; NAC; | C;N; NAC; | NoAc; MIS; | NoAc; MIS; |
| CON: Contact | NoAc; CON; | C;N; NAC; | C;N; NAC; | NoAc; CON; | NoAc; CON; |
| KON: Konnect | NoAc; KON; | C;DT;N; NAC; | C;DT;N; NAC; | NoAc; KON; | NoAc; KON; |
| OPE: Operation | L;T2; OPE; direct info | C;DT;IT;N; NAC; | C;DT;IT;N; NAC; | NoAc; OPE; | NoAc; OPE; |
| TAN: Tandem | NoAc; TAN; | C; NAC; exit from TAN | C; NAC; exit from TAN | NoAc; TAN; | NoAc; TAN; |

Table 28: Special Events, Timeouts

| Event: | Runout | T==0 | Frame_Sync_Lost | Frame_Sync_Lost | Mes_Sync_Lost |
|------------------------------------|--|--|---|---|--|
| Number: | 31 | 32 | 34 | 35 | 36 |
| Condition: & | . | . | n<3 | n>2 | . |
| Comment: | IPEs may be- come unsynch. | Time-Out | No Action | Stop TFO Frames, if 3 Frames missing | . |
| State: | . | . | . | . | . |
| NAC: Not_Active | ----- ----- | ----- ----- | ----- ----- | ----- ----- | ----- ----- |
| WAK: Wakeup | ----- ----- | ----- ----- | ----- ----- | ----- ----- | ----- ----- |
| FIT: First_Try | U;N; MON PSTN Call | ----- ----- | ----- ----- | ----- ----- | NoAc; FIT; |
| COR: Continuous Retry | U;L1;T5; PER; at end of COR | C;N;REQ; COR; Reset IPEs | ----- ----- | ----- ----- | NoAc; COR; |
| PER: Periodic Retry | NoAc; PER; | L1;T5; PER; Periodic Test | ----- ----- | ----- ----- | NoAc; PER; |
| MON: Monitor | ----- ----- | ----- ----- | ----- ----- | ----- ----- | ----- ----- |
| MIS: Mismatch | NoAc; MIS; typ. Final state | N;B; MIS; List not Ack_ed! | NoAc; MIS; | NoAc; MIS; | NoAc; MIS; |
| CON: Contact | REQ; COR; can this occur? | ----- ----- | ----- ----- | ----- ----- | C;REQ; COR; |
| KON: Konnect | NoAc; KON; may happen | C;DT;N; TAN; Misbehaviour! | ----- ----- | ----- ----- | C;DT;REQ;T1; COR; after Timeout: N |
| OPE: Operation | NoAc; OPE; typ. Final event | B; OPE; List not Ack_ed! | NoAc; OPE; 1: Alarm, go on | C;DT;IT; F;REQ; COR; 2: Alarm, stop! | NoAc; OPE; Typ. Final event |
| TAN: Tandem | NoAc; TAN; typical | ----- ----- | ----- ----- | ----- ----- | NoAc; TAN; don't trust! |

Table 29: System Commands

| | | |
|---------------------------------------|--------------------------------------|--------------------------------------|
| Event: | Inhibit_TFO | Allow_TFO |
| Number: | 38 | 39 |
| Condition: & | - . | - . |
| Comment: . | System command Received | System command Received |
| State: | . | . |
| NAC: Not_Active . | ----- ----- PCM not available! | ----- ----- PCM not available! |
| WAK: Wakeup . | ----- ----- PCM not available! | ----- ----- PCM not available! |
| FIT: First_Try . | C;N TAN | NoAc FIT |
| COR: Continuous Retry | C;N TAN . | NoAc COR |
| PER: Periodic Retry . | C;N TAN | NoAc PER |
| MON: Monitor . | C;N TAN | NoAc MON |
| MIS: Mismatch . | C;N TAN | NoAc MIS |
| CON: Contact . | C;N TAN | NoAc CON |
| KON: Konnnect . | C;DT;N TAN | NoAc KON |
| OPE: Operation . | C;DT;IT;N TAN | NoAc OPE |
| TAN: Tandem . | NoAc; TAN; | C; F; REQ FIT |

13State Machine of the TFO_Protocol Process for TDMA

The TFO_Protocol Process can be described by a State Machine, consisting of 15 States: five main States with several sub-States; exception handling needs further States, see figure 11:

- Initialisation (• Not_Active, • Wakeup).
- Establishment (• First_Try, • Continuous_Retry, • Periodic_Retry, • Monitor, • Mismatch).
- Contact (• Contact).
- Konnnect (• Konnect).
- Operation (• Operation).
- Local HO (• Fast_Try, • Fast_Contact).
- Distant HO (• Sync_Lost, • Re_Konnnect).
- Misbehaviour (• Failure).

It is assumed that Events (Conditions checking, Actions and Transition to an other State) are handled almost instantaneous and in any case significantly shorter than the time required to complete the transmission of any one TFO Message or TFO Frame.

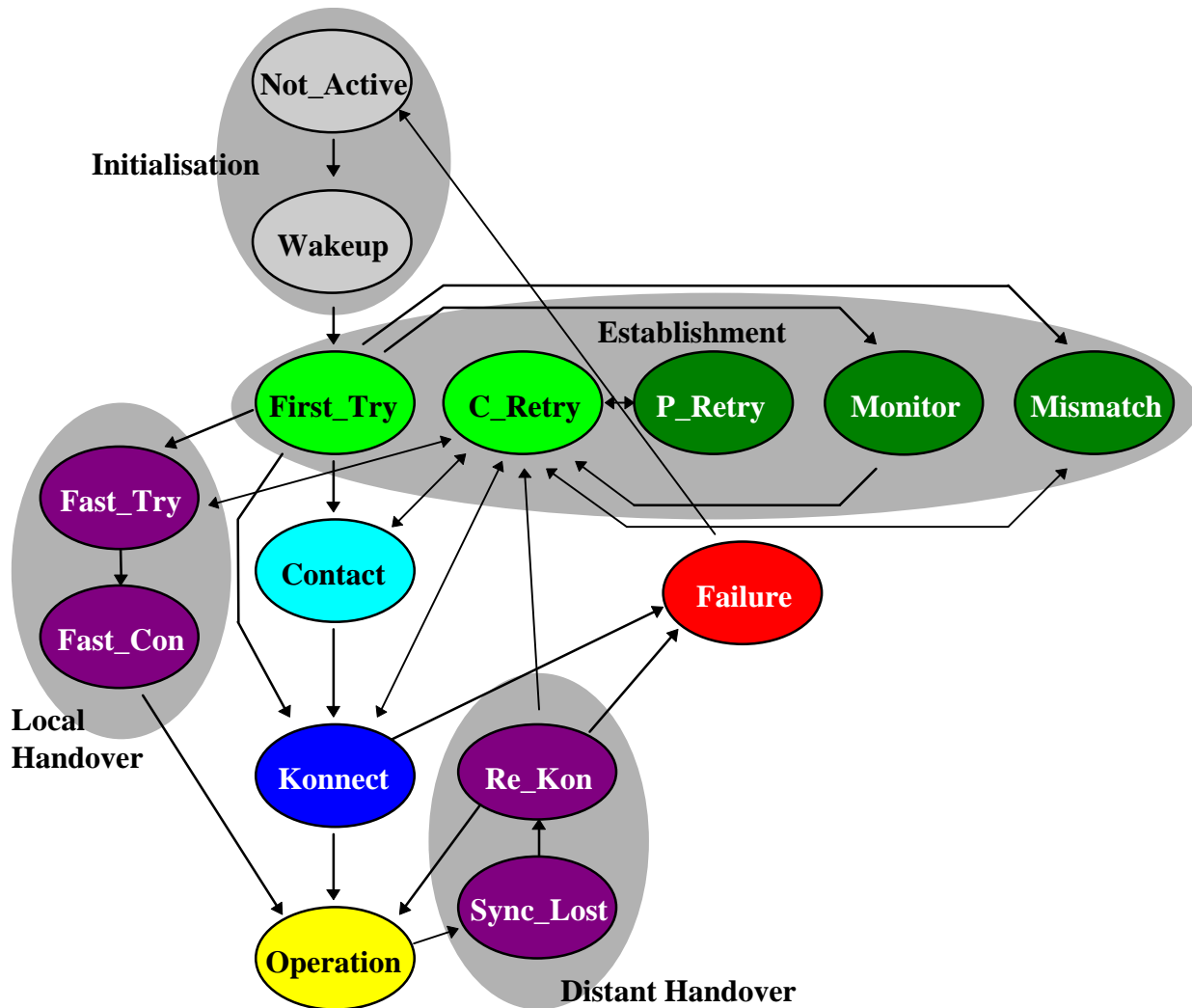


Figure 12: TFO_Protocol State Machine with most important transitions

13.1 Initialisation

13.1.1 Not_Active State

The state Not_Active is standard and manufacturer dependent and can not be described in detail here. The TRAU basically performs no speech coding, but may monitor the link for certain patterns (e.g. Idle patterns). The TRAU may also be in Data mode, which is also not described further, but is handled here as "Not_Active".

13.1.2 Wakeup State

The Wakeup State is entered, when the TRAU is activated by the appropriate event (e.g. activation by a MSC message or reception of TRAU Frames). The Wakeup state is characterized by the TRAU sending speech data at least in one direction.

The transition to Establishment is performed, if speech processing is performed in both. This is the point in time where the time out for TFO Messages starts, i.e. a maximum number of TFO_REQ Messages shall be sent after that.

13.2 Establishment

The Establishment submits several slightly different situations:

| | |
|------------------|---|
| First_Try | when the TRAU just has started; it sends TFO_REQ Messages continuously; |
| Continuous_Retry | when Contact to a TFO Partner has existed, but was interrupted recently; |
| Periodic_Retry | when Contact to a TFO Partner had existed, but was interrupted some time ago; |
| Monitor | when no TFO partner could be found, but the TRAU continues to monitor the PSTN Interface; |
| Mismatch | when a TFO partner with a different Codec has been identified. |

Loopback is a specific situation, when the call is still not through connected and the TRAU receives the own sent signals. No specific State is allocated to describe this situation. Instead, loopback is handled in First_Try and Continuous_Retry. Common to all these situations is that the TRAU does not know, if there is a distant TFO partner and/or if the links are digitally transparent. Typically, TFO_REQ Messages are sent and expected.

Due to HO cases it might, however, happen that a TRAU is initialised into an existing connection and therefore the other TFO Partner may be in any State and all other TFO Messages may be received, too.

Especially important is, when TFO Frames are received, since then it can be assumed that an existing TFO Connection was handed over to a newly initialised TRAU and the TFO should be continued - if possible uninterrupted - as soon as possible. The TRAU may see from the TFO Frames the Distant_Used_Codec of a TFO Partner and that the receiving path is digitally transparent, but it can not assume that the path to the other TFO Partner is also (already) transparent.

TFO_Protocol enters the exceptional State: Fast_Try, sending a specific, short TFO_DUP Message to test the other direction.

13.2.1 First_Try State

The TRAU sends and receives PCM samples on and from the PSTN interface. Regular TFO_REQ Messages are sent onto the PSTN interface continuously for a certain maximum time. After that, if no TFO Partner answers before, Tx_TFO reports a Runout of TFO Messages, and TFO_Protocol enters automatically into the Monitor State.

If TFO_REQ Messages are received with the same, own Signature, then a circuit loop back is assumed, i.e. the call is still not through-connected. The TRAU selects a new Signature and continues sending TFO_REQ Messages, until a different Signature is received. Since loop back delays may be substantial in some cases, the TRAU has to remember and compare also the previously selected own Signature. Care has to be taken that the Signature selection contains a true random element to avoid that two different TRAUs select by coincidence identical signatures again and again.

13.2.2 Continuous_Retry State

TFO Contact had existed, either by TFO Messages or by TFO Frames, but was interrupted and sync was lost. The TRAU sends a maximum number of regular TFO_REQ Messages continuously, to test, if TFO could be re-established. If Tx_TFO reports a Runout of TFO Messages, then the TFO_Protocol enters the Periodic_Retry State.

13.2.3 Periodic_Retry State

Entered from Continuous_Retry, TFO_Protocol tests from time to time by a single TFO_REQ_L, if TFO could be re-established. As soon as a TFO Message is received, TFO_Protocol leaves this State.

NOTE: Since no contiguous transmission of TFO Messages is ongoing, possible IPEs may be unsynchronized.

13.2.4 Monitor State

The TRAU monitors the PSTN interface for TFO Messages or TFO Frames, but it does not send TFO Messages or TFO Frames. As soon as a TFO Message from a distant partner (a TRAU or a TCME) has been received, the TRAU knows that a TFO Partner exists and it knows that the transmission path from the partner is digitally transparent.

The TRAU may already now see, whether TFO is possible, but it must ensure that all IPEs are synchronised. It therefore transits into the Continuous_Retry State. In case of Codec Mismatch, it terminates the TFO Protocol by sending TFO_REQ_L back, informs its local MSC and transits into Mismatch.

NOTE: Since no contiguous transmission of TFO Messages is ongoing, possible IPEs may be unsynchronized.

13.2.5 Mismatch State

From an previous contact it is obvious, that a distant TFO Partner exists, but the Codecs do not match.

The TRAU waits without sending TFO Messages or TFO Frames, if either the distant TFO Partner changes to the Local_Used_Codec, or the local MSC solves the Codec mismatch situation by an intra cell HO to the Distant_Used_Codec.

NOTE: Since no contiguous transmission of TFO Messages is ongoing, possible IPEs may be unsynchronized.

13.3 Contact State

There is a distant TFO Partner, which has sent TFO_REQ. The Codecs do match. The link from the distant partner is transparent. Now TFO_ACK need to be sent to check the transparency of the link to the distant partner.

As soon as a TFO_ACK or TFO_TRANS from a distant partner has been received, the TRAU knows that the links in both directions are digitally transparent. The TRAU sends TFO_TRANS to bypass the IPEs and starts sending TFO Frames. It transits into Konnect State.

13.4 Konnect State

The TRAU sends TFO Frames and possibly embedded TFO Messages as long as it receives correct TFO Messages.

The first received TFO Frame causes the transition into the final Operation State.

If no TFO Frames are received within a certain period, the TRAU transits to the Failure State.

13.5 Operation State

In this State - the main State of TFO_Protocol - the TRAU sends and receives TFO Frames, thus the TFO Connection is fully operating. TFO Messages may occur embedded into TFO Frames.

13.6 Local HO

13.6.1 Fast_Try State

When the TRAU in First_Try receives suddenly TFO Frames and the Codecs do match, then there is a high probability that a local HO has initialised the TRAU into an existing TFO connection and a fast TFO establishment is likely. The TFO_Protocol has still to check, whether the link to the distant TFO Partner is (already) transparent. This is done by the specific TFO_DUP Message.

Since the HO must have been a local HO, i.e. close to the (new) TRAU, it can be assumed that the possibly existing IPEs are still in transparent mode and TFO Messages therefore pass through directly.

13.6.2 Fast_Contact State

This State is entered from First_Try via Fast_Try, if TFO Frames and then TFO_SYL Messages are received. The TRAU continues to send TFO_DUP Messages, until TFO Frames are received again. Then it immediately starts to send TFO Frames, with a TFO_TRANS embedded into the first TFO Frames. The TRAU transits directly to Operation State.

13.7 Distant HO, TFO Interruption

13.7.1 Sync_Lost State

If the TRAU was in Operation State and suddenly the TFO Frame synchronisation is lost, then the TRAU enters the Sync_Lost State for a short while, before it transits to Continuous_Retry.

If synchronisation was lost due to a distant HO, then a fast TFO establishment might be possible and the TRAU enters Operation State soon again. In Sync_Lost it expects TFO_DUP Message as confirmation of the distant HO. Then it transits to Re_Konnect.

13.7.2 Re_Konnect State

This State is entered from Operation via Sync_Lost, if TFO_DUP Messages are received. The TRAU starts immediately to send TFO Frames again, with a TFO_TRANS embedded into the first TFO Frames. The TRAU transits back to Operation State, as soon as TFO Frames are received, again.

13.8 Failure State

This State is entered when the distant partner shows an incorrect behaviour. The TRAU then sends pure PCM samples onto the PSTN interface and waits for the failure to disappear. It does not send TFO Frames or TFO Messages.

14 Detailed Description of TFO_Protocol for CDMA

The TFO_Protocol Process is always in one well defined State. An Event triggers Actions and a Transition into another State. The TFO Protocol is described in a table-wise manner, with a syntax as defined in Table 30.

Table 30: Definition of the Syntax for the State Machine Description

| | | | |
|-----------------|--|-----|--|
| Event: | <Received Message> | ... | <Other Event> |
| Number: | <running number> | | <running number> |
| Condition: | [<Condition>] & [<Condition>] | | [<Condition>] [<Condition>] |
| Comment: | [<Comment>] | | [<Comment>] |
| State: | | | |
| <Actual State>: | <Action Name>;<Action Name>; <Next State>; [<Comment>] | | <Action Name>;<Action name>; <Next State>; [<Comment>] |
| ... | | | |
| <Actual State>: | <Action Name>;<Action Name>; <Next State>; [<Comment>] | | <Action Name>;<Action Name>; <Next State>; [<Comment>] |

Several tables, table 33 to table 40, are necessary to describe the whole State Machine. If a table entry contains only a dashed line, it is assumed that this combination can not occur. If, however, such a combination is encountered during operation, it is treated as 'don't care'.

The Actions are described in Table 32, with a syntax as defined in table 31.

Table 31: Definition of Syntax for Action Table

| Name | Action List | Comment |
|---------------|-------------------------|-----------|
| <Action Name> | <Action >;[<Action >;] | <Comment> |
| ... | | |
| <Action Name> | <Action >;[<Action >;] | <Comment> |

Tx := TFO_REQ means, that TFO_Protocol places a command into Tx_Queue. Tx_TFO handles the details autonomously and generates a TFO_REQ Message for transmission over the PSTN interface, when it comes to that command.

Tx := 31*TFO_REQ means: put 31 TFO_REQ commands into Tx_Queue. Not necessarily all will really trigger TFO_REQ Messages. In most cases Tx_Queue will be cleared before. Similar definitions hold for the other messages. The Tx_Queue is a first_in_first_out command queue. It is filled by TFO_Protocol and read by Tx_TFO.

Clear Tx_Queue, means that all remaining commands are deleted from the Tx_Queue in that very moment (time T_c). Note that due to the duration time to transmit a TFO_Message completely, the TFO_Protocol Process is often already within an other State while still TFO Messages commanded in earlier States are within the Tx_Queue or under transmission.

MSC := TFO () means that a message is sent to the local MSC; similar

Tx_TRAU := ... means a message to Tx_TRAU.

An Event **TFO_REQ** means that a TFO_REQ Message was correctly received on the PSTN interface. The Rx_TFO Process has sent a message to TFO_Protocol, containing the new values for the respective variables. TFO_Protocol updates its variables with the new values. Similar definitions hold for the other messages.

One Timer **T := <Time_out>** is necessary to describe time out situations. The notation **T := DIS** means that the Timer is disabled. Positive values are decremented in an hidden background process in steps of 20 ms. When T gets to the value "0", then the TFO_Protocol process is invoked.

Local_Used_Codec (short form: **Luc**) means the type of speech Codec used in the local TRAU and.

New Local_Used_Codec (Nluc) refers to the new codec received in "In_Call_Modifications".

Distant_Used_Codec (Duc) means the type of speech Codec used by the distant partner, as reported in TFO_REQ... or TFO_ACK...

Distant_Prefered_Codec (DPC) means the type of speech Codec that the distant partner would prefer, as reported in TFO_REQ_P.

Local_Codec_List (LCL) means the list of all Codecs that could alternatively be used, i.e. which are supported by both the local MS and the local MSC. It always contains at least the Local_Used_Codec.

It is reported in TFO_REQ_L, TFO_ACK_L or TFO_REQ_P.

Distant_Codec_List (DCL) means the list of all Codecs that could alternatively be used, i.e. which are supported by the distant MS and the distant MSC. It always contains at least the Distant_Used_Codec.

All these variables are initialized to **UNKNOWN**, which means that the contents of the variables are not defined.

Local_Signature (Lsig) means the 8-bit random number in TFO_REQ, which identifies the local TFO_REQ Messages. It is also used in TFO_REQ_L.

Distant_Signature (Dsig) means the 8-bit random number as received in TFO_REQ, TFO_REQ_L and TFO_REQ_P, in TFO_ACK and TFO_ACK_L

If received in TFO_REQ, TFO_REQ_L and TFO_REQ_P, then it should be different to the Local_Signature, otherwise loop back must be assumed (exceptions exist).

If received in TFO_ACK or TFO_ACK_L, then it should be identical to the Local_Signature, otherwise the TFO_ACK is not a response to an own TFO_REQ respectively TFO_REQ_L, but maybe was created during a HO situation.

Local_Channel_Type (LCh) and **distant_Channel_Type (DCh)** refer to the 8 or 16 kBit/s transparent channel used by the local Tx_TFO respectively received by the distant TFO_TRANS.

Error protection and error handling: It is assumed that the defined error protection is strong enough for the error rates encountered on typical PSTN interface links. The few occuring errors are in practically all cases detected and possibly even corrected by Rx_TFO, before reported to TFO_Protocol. Therefore TFO_Protocol can rely on the correctness of the received Events. The protocol is, however, "selfhealing" and will handle the unlikely erroneous reported Events, too.

The Event "**PCM_Channel_Available**" is given if in State Wakeup, if valid PCM samples are received on the PSTN interface. This event can be triggered by a higher system level.

Fast HO handling: The defined protocol assumes that the new TRAU, to which the HO is performed, is already in State Wakeup before the PSTN Interface is switched to that TRAU. Only then the TFO Frames can be received by that TRAU and fast HO handling is possible.

Timing: If two Events occur by coincidence at the same time, then they shall be processed in the order given by the tables 32 to 39 (left to right) . TFO Messages arrive always some time before the embedding TFO Frame and shall be handled therefore first.

Runout is the Event, when the last TFO Message has been taken from the Tx_Queue and the last 10 bits are going to be sent by Tx_TFO to the PSTN interface. So there is still some time for TFO_protocol to react and place a further TFO Message into Tx_Queue, which then shall be transmitted without gap to the messages before.

Table 32: Defined Actions

| Name | Actions | Comments |
|------|---|--|
| C | Clear Tx_Queue; T := DIS; | Initialise Tx_Queue and disable the timer |
| T1 | T := 1s; | Set Timeout of 1 second |
| T2 | T := 2s; | Set Timeout to 2 seconds |
| T5 | T := 5s; | Set Timeout to 5 seconds |
| NoAc | . | No Action required |
| S | Lsig := New_Random_Number; Old_Sig := UNKNOWN; | Generate new Signature and set Old_Sig to unknown; if no Loopback is assumed. |
| SO | Old_Sig := Lsig; Lsig := New_Random_Number; | Remember old Signature and generate a new Signature, if Loopback is assumed. |
| U | Old_Sig := UNKNOWN; | Reset Old_Sig before leaving FIT or COR |
| F | Tx := 3*TFO_FILL; | "Hello IPEs! Please synchronise!" |
| T | Tx := TFO_TRANS (); | "Hello IPEs! Please open a transparent channel!" |
| N | Tx := TFO_NORMAL; | "Hello IPEs! Please return to normal operation!" |
| REQ | Tx := 35*TFO_REQ; | "Hello Partner? Can You do TFO with me?" |
| ACK | Tx := 7*TFO_ACK; | "Yes, I can do TFO with You!" |
| SYL1 | Tx := TFO_SYL; | "Hello Partner! I lost one or more TFO_Frames!" |
| SYL | Tx := 4*TFO_SYL; | "Hello Partner! Serious interruption of TFO_Frames!" |
| DUP | Tx := 5*TFO_DUP; | HO? "Hey, I see Your TFO Frames, Fine!" |
| L1 | Tx := TFO_REQ_L; | "Here is my Codec_List! Can you hear me?" |
| L | Tx := 6*TFO_REQ_L; | "Here is my Codec_List, please acknowledge!" |
| LA | Tx := TFO_ACK_L; | "Yes, I received Your Codec_List! Here is mine!" |
| BT | Tx := Begin_TFO; | Begin Transmission of TFO Frames |
| DT | Tx := Discontinue_TFO; | Discontinue Transmission of TFO Frames |
| IT | Tx_TRAU := Ignore_TFO; | Tx_TRAU works as conventional downlink TRAU |
| AT | Tx_TRAU := Accept_TFO; | Tx_TRAU bypasses TFO_Frames |
| B | MSC := TFO (); | "Hello MSC! Some news from the TFO_Scene!" |

Table 33: Call Setup and Loopback Handling

| Event: | New_Speech_Call | PCM_Channel Available | TFO_REQ | TFO_REQ |
|---|---|--|---|-----------------------------------|
| Number: | 24 | 29 | 0 | 0a |
| Condition: & | . | . | Duc==Luc Dsig==Lsig | Duc==Luc Dsig==Old_Sig |
| Comment: | System activates TRAU | PSTN-Interface gets active; occurs only at beginning | Loopback (LB), or distant HO? wrong Signature ? | Loopback (LB), or distant HO? |
| State: | | | | |
| NAC: Not_Active . | C;S;IT; WAK; typ. 1rst Event | ----- ----- . | ----- ----- . | ----- ----- . |
| WAK: Wakeup . | ----- ----- . | C;F;REQ; FIT; typ. 2nd Event | ----- ----- . | ----- ----- . |
| FIT: First_Try . | ----- ----- . | ----- ----- . | C;SO;REQ; FIT; LB! | NoAc; FIT; Ignore LB |
| COR: Continuous Retry . | ----- ----- . | ----- ----- . | C;SO;REQ; COR; LB!? | NoAc; COR; Ignore LB |
| PER: Periodic Retry . | ----- ----- . | ----- ----- . | C;F;S;ACK; CON; Dist. HO! | ----- ----- . |
| MON: Monitor . | ----- ----- . | ----- ----- . | C;F;S;REQ; FIT; Dist. HO! | ----- ----- . |
| MIS: Mismatch . | ----- ----- . | ----- ----- . | C;F;S;ACK; CON; Dist. HO! | ----- ----- . |
| CON: Contact . | ----- ----- . | ----- ----- . | C;SO;REQ; COR; safe way | ----- ----- . |
| FAT: Fast Try . | ----- ----- . | ----- ----- . | C;SO;REQ; COR; safe way | ----- ----- . |
| FAC: Fast Contact . | ----- ----- . | ----- ----- . | C;SO;REQ; COR; safe way | ----- ----- . |
| KON: Konnect . | ----- ----- . | ----- ----- . | C;DT;SO;REQ;T1; COR; IPEs transparent! | ----- ----- . |
| REK: Re_Konnect . | ----- ----- . | ----- ----- . | C;DT;SO;REQ;IT;B;T1; COR; IPEs transparent! | ----- ----- . |
| SOS: Sync_Lost . | ----- ----- . | ----- ----- . | C;IT;S;REQ;B;T1; COR; Contact is back | ----- ----- . |
| OPE: Operation . | ----- ----- . | ----- ----- . | ----- ----- . | ----- ----- . |
| FAI: Failure . | ----- ----- . | ----- ----- . | NoAc; FAI; . | ----- ----- . |

Table 34: Most Important Cases, Especially at Call Setup

| Event: | TFO_REQ | TFO_ACK | TFO_ACK | TFO_TRANS | TFO_FRAME |
|------------------------------------|--|---|---|--|---|
| Number: | 1 | 2 | 3 | 4 | 5 |
| Condition: & | Duc==Luc Dsig!=Lsig | Duc==Luc Dsig==Lsig | Duc==Luc Dsig!=Lsig | DCh==LCh . | Duc==Luc n<3 |
| Comment: | Distant REQ | Distant ACK | Wrong Response | similar to ACK | one or two |
| State: | Good Signature . | Good Signature . | HO? . | As response to loc ACK_? . | TFO Frames . |
| NAC: Not_Active . | ----- ----- . | ----- ----- . | ----- ----- . | ----- ----- . | ----- ----- . |
| WAK: Wakeup . | ----- ----- . | ----- ----- . | ----- ----- . | ----- ----- . | ----- ----- . |
| FIT: First_Try . | C;U;ACK; CON; typical | C;U;T;BT;T;T1; KON; typical; IPEs! | C;REQ; FIT; . | NoAc; FIT; wait f. Frames | C;U;DUP; FAT; 1: HO |
| COR: Continuous Retry | C;U;ACK; CON; typical | C;U;T;BT;T;T1; KON; typical; IPEs! | C;REQ; COR; . | NoAc; COR; wait f. Frames | C;U;DUP; FAT; 1: Call is back? |
| PER: Periodic Retry | C;F;ACK; CON; OK, Contact back | C;F;S;REQ; COR; rare case, test | C;F;REQ; COR; . | NoAc; PER; wait f. Frames | C;DUP; FAT; 1: Call is back? |
| MON: Monitor . | C;F;REQ; FIT; IPEs? | C;F;S;REQ; FIT; Rare case, test | C;F;REQ; FIT; . | NoAc; MON wait f. Frames | C;DUP; FAT; 1: Call is back? |
| MIS: Mismatch . | C;F;ACK; CON; Mismatch resolved | C;F;S;REQ; COR; rare case, test | C;F;REQ; COR; . | NoAc; MIS; wait f. Frames | C;DUP; FAT; 1: Call is back? |
| CON: Contact . | C;ACK; CON; typical: wait | C;T;BT;T;T1; KON; typical: yes! | C;REQ; COR; . | C;T;BT;T;T1; KON; Fast way! | C;T;BT;T;T1; KON; Missed TRANS? |
| FAT: Fast Try | C;REQ; COR; safe way | C;REQ; COR; safe way | C;REQ; COR; safe way | NoAc; FAC; wait f. Frames | NoAc; FAT; 2: typ. Loc. HO |
| FAC: Fast Contact | C;REQ; COR; safe way | C;REQ; COR; safe way | C;REQ; COR; safe way | NoAc; FAC; wait f. Frames | C;BT;T;L;T2;AT;B; OPE; 5: typ. Loc. HO |
| KON: Konnnect . | C;DT;REQ;T1; COR; IPEs transparent! | NoAc; KON; Typical: wait | NoAc; KON; . | NoAc; KON; typical: wait | AT;L;T2;B; OPE; typ: call setup |
| REK: Re_Konnnect . | C;DT;REQ;IT;B;T1; COR; IPEs transparent! | C;DT;REQ;IT;B;T1; COR; . | C;DT;REQ;IT;B;T1; COR; . | NoAc; REK; wait f. Frames | AT;L;T2;B; OPE; 5: typ. Dis. HO |
| SOS: Sync_Lost . | C;IT;REQ;B;T1; COR; Contact is back | C;IT;REQ;B;T1; COR; Contact is back | C;IT;REQ;B;T1; COR; Contact is back | NoAc; SOS; wait f. Frames | C;BT;T;L;T2;B; OPE; short Interrupt? |
| OPE: | ----- | ----- | ----- | NoAc; | NoAc; |

| | | | | | |
|-----------------------------|--------------------|--------------------|--------------------|--------------------|---------------------------|
| Operation . | ----- . | ----- . | ----- . | OPE; typ in HO | OPE; Main! TFO! |
| FAI: Failure . | NoAc; FAI; . | NoAc; FAI; . | NoAc; FAI; . | NoAc; FAI; . | NoAc; FAI; . |

Table 35: In Call Modification and HO

| Event: | New_Local_Codec | New_Local_Codec | TFO_FRAME | TFO_SYL | TFO_DUP |
|------------------------------------|---|---|----------------------------------|---|--|
| Number: | 25 | 26 | 6 | 7 | 8 |
| Condition: & | Duc==Nluc . | Duc!=Nluc . | Duc==Luc n>2 . | . . | . . |
| Comment: State: | in Call Modif. Mismatch resolv. (Luc!=Nluc) | In Call Modif. Mismatch occurs! (Luc!=Nluc) | Three or more TFO Frames . | the dist. TRAU lost sync in OPE | the dist. TRAU recognised HO |
| NAC: Not_Active | ----- ----- . | ----- ----- . | ----- ----- . | ----- ----- . | ----- ----- . |
| WAK: Wakeup | NoAc; WAK; . | NoAc; WAK; . | ----- ----- . | ----- ----- . | ----- ----- . |
| FIT: First_Try | C;REQ; FIT; restart | C;REQ; FIT; restart | ----- ----- . | NoAc; FIT; HO? Ignore | NoAc; FIT; HO? Ignore |
| COR: Continuous Retry | C;REQ; COR; . | C;REQ; COR; . | ----- ----- . | NoAc; COR; ignore | NoAc; COR; ignore |
| PER: Periodic Retry | L1;T5; PER; . | L1;T5; PER; . | ----- ----- . | C;F;REQ; COR; rare case, test | C;F;REQ; COR; rare case, test |
| MON: Monitor | NoAc; MON . | NoAc; MON . | ----- ----- . | C;F;REQ; FIT; rare case, test | C;F;REQ; FIT; rare case, test |
| MIS: Mismatch | C;F;REQ; COR; Mismatch res. | L;T2;B; MIS; Direct info. | ----- ----- . | C;F;REQ; COR; rare case, test | C;F;REQ; COR; rare case, test |
| CON: Contact | ----- ----- . | C;L;T2;B; MIS; . | ----- ----- . | C;F;REQ; COR; rare case, test | C;F;REQ; COR; rare case, test |
| FAT: Fast Try | ----- ----- . | C;L;T2;B; MIS; . | NoAc; FAC; . | NoAc; FAC; 3: typ. Loc. HO | C;F;REQ; COR; rare case, test |
| FAC: Fast Contact | ----- ----- . | C;L;T2;B; MIS; . | C;BT;T;L;T2;AT;B; OPE; . | NoAc; FAC; 4: typ. Loc. HO | C;F;REQ; COR; rare case, test |
| KON: Konnnect | ----- ----- . | C;DT;L;T2;B; MIS; . | ----- ----- . | NoAc; KON; wait, short int? | NoAc; KON; other TRAU? |
| REK: Re_Konnnect | ----- ----- . | C;DT;IT;L;T2;B; MIS; . | ----- ----- . | C;DT;SYL; SOS; IPEs not transp? | NoAc; REK; 4: typ. Dist. HO |
| SOS: Sync_Lost | ----- ----- . | C;IT;L;T2;B; MIS; . | ----- ----- . | NoAc; SOS; short Inter? | C;BT;T;T1; REK; 3: typ. Dis. HO |
| OPE: Operation | ----- ----- . | C;DT;IT;L;T2;B; MIS; . | NoAc; OPE; Main! TFO! | NoAc; OPE; Short interrupt? | NoAc; OPE; Typical |
| FAI: Failure | ----- ----- . | NoAc; FAI; . | NoAc; FAI; . | NoAc; FAI; . | NoAc; FAI; . |

Table 38: Mismatching TFO_TRANS and TFO Frames

| Event: | TFO_TRANS | TFO_FRAME | TFO_FRAME |
|---|----------------------------------|-----------------------------------|--|
| Number: | 21 | 22 | 23 |
| Condition: & | DCh!=LCh . | Duc!=Luc n==1 | Duc!=Luc n>1 |
| Comment: State: | Mismatch of channel type . | Mismatch for one TFO Frames | Mismatch for at least two TFO Frames |
| NAC: Not_Active . | ----- ----- . | ----- ----- . | ----- ----- . |
| WAK: Wakeup . | ----- ----- . | ----- ----- . | ----- ----- . |
| FIT: First_Try . | C;U;L;T2;B; MIS; HO? | NoAc; FIT; HO? Be tolerant | C;U;L;T2;B; MIS; HO! Mismatch! |
| COR: Continuous Retry . | C;U;L;T2;B; MIS; . | NoAc; COR; Call Forw.? | C;U;L;T2;B; MIS; . |
| PER: Periodic Retry . | C;F;L;T2;B; MIS; . | NoAc; PER; Call Forw.? | C;F;L;T2;B; MIS; . |
| MON: Monitor . | C;F;L;T2;B; MIS; . | NoAc; MON Call Forw.? | C;F;L;T2;B; MIS; . |
| MIS: Mismatch . | C;L;T2;B; MIS; . | NoAc; MIS; Call Forw.? | C;L;T2;B; MIS; . |
| CON: Contact . | C;L;T2;B; MIS; . | NoAc; CON; . | C;L;T2;B; MIS; . |
| FAT: Fast Try . | C;L;T2;B; MIS; . | NoAc; FAT; . | C;L;T2;B; MIS; . |
| FAC: Fast Contact . | C;L;T2;B; MIS; . | NoAc; FAC; . | C;L;T2;B; MIS; . |
| KON: Konnnect . | C;DT;L;T2;B; MIS; . | NoAc; KON; . | C;DT;L;T2;B; MIS; . |
| REK: Re_Konnnect . | C;DT;L;T2;IT;B; MIS; . | NoAc; REK; . | C;DT;L;T2;IT;B; MIS; . |
| SOS: Sync_Lost . | C;L;T2;IT;B; MIS; . | NoAc; SOS; . | C;L;T2;IT;B; MIS; . |
| OPE: Operation . | NoAc; OPE; ignore? | NoAc; OPE; Hard HO? | C;DT;L;T2;IT;B; MIS; Hard HO into TFO |
| FAI: Failure . | NoAc; FAI; . | NoAc; FAI; . | NoAc; FAI; . |

Table 39: Local Events, Call Termination

| Event: | New_L_Codec_List | Data_Call | TRAU_Idle | TFO_FILL | TFO_NORMAL |
|------------------------------------|---------------------------------------|------------------------------------|-------------------------------------|---------------------------|----------------|
| Number: | 30 | 27 | 28 | 37 | 33 |
| Condition: & | . | . | . | . | . |
| Comment: | from MSC | in Call Modif. stop TFO | Command from BS to Reset TRAU | ignore, is just Filler | ignore |
| State: | . | . | . | . | . |
| NAC: Not_Active | NoAc; NAC; | NoAc; NAC; | NoAc; NAC; | ----- ----- | ----- ----- |
| WAK: Wakeup | NoAc; WAK; | NoAc; NAC; | NoAc; NAC; | ----- ----- | ----- ----- |
| FIT: First_Try | NoAc; FIT; update loc. Par. | C;N; NAC; | C;N; NAC; | NoAc; FIT; | NoAc; FIT; |
| COR: Continuous Retry | NoAc; COR; | C;N; NAC; | C;N; NAC; | NoAc; COR; | NoAc; COR; |
| PER: Periodic Retry | NoAc; PER; | C;N; NAC; | C;N; NAC; | NoAc; PER; | NoAc; PER; |
| MON: Monitor | NoAc; MON | C;N; NAC; | C;N; NAC; | NoAc; MON | NoAc; MON |
| MIS: Mismatch | C;L;T2; MIS; direct info | C;N; NAC; | C;N; NAC; | NoAc; MIS; | NoAc; MIS; |
| CON: Contact | NoAc; CON; | C;N; NAC; | C;N; NAC; | NoAc; CON; | NoAc; CON; |
| FAT: Fast Try | NoAc; FAT; | C;N; NAC; | C;N; NAC; | NoAc; FAT; | NoAc; FAT; |
| FAC: Fast Contact | NoAc; FAC; | C;N; NAC; | C;N; NAC; | NoAc; FAC; | NoAc; FAC; |
| KON: Konnnect | NoAc; KON; | C;DT;N; NAC; | C;DT;N; NAC; | NoAc; KON; | NoAc; KON; |
| REK: Re_Konnnect | NoAc; REK; | C;DT;IT;N; NAC; | C;DT;IT;N; NAC; | NoAc; REK; | NoAc; REK; |
| SOS: Sync_Lost | NoAc; SOS; | C;IT;N; NAC; | C;IT;N; NAC; | NoAc; SOS; | NoAc; SOS; |
| OPE: Operation | L;T2; OPE; direct info | C;DT;IT;N; NAC; | C;DT;IT;N; NAC; | NoAc; OPE; | NoAc; OPE; |
| FAI: Failure | NoAc; FAI; | C; NAC; exit from FAI | C; NAC; exit from FAI | NoAc; FAI; | NoAc; FAI; |

Table 40: Special Events, Timeouts

| Event: | Runout | T==0 | Frame_Sync_Lost | Frame_Sync_Lost | Mes_Sync_Lost |
|------------------------------------|--|--|---|---|---|
| Number: | 31 | 32 | 34 | 35 | 36 |
| Condition: & | . | . | n<3 | n>2 | . |
| Comment: | IPEs may be- come unsynch. | Time-Out | start to send SYL already | Stop TFO Frames, if 3 Frames missing | . |
| State: | . | . | . | . | . |
| NAC: Not_Active | ----- ----- | ----- ----- | ----- ----- | ----- ----- | ----- ----- |
| WAK: Wakeup | ----- ----- | ----- ----- | ----- ----- | ----- ----- | ----- ----- |
| FIT: First_Try | U;N; MON PSTN Call | ----- ----- | ----- ----- | ----- ----- | NoAc; FIT; |
| COR: Continuous Retry | U;L1;T5; PER; at end of COR | C;N;REQ; COR; Reset IPEs | ----- ----- | ----- ----- | NoAc; COR; |
| PER: Periodic Retry | NoAc; PER; | L1;T5; PER; Periodic Test | ----- ----- | ----- ----- | NoAc; PER; |
| MON: Monitor | ----- ----- | ----- ----- | ----- ----- | ----- ----- | ----- ----- |
| MIS: Mismatch | NoAc; MIS; typ. Final state | N;B; MIS; List not Ack_ed! | NoAc; MIS; | NoAc; MIS; | NoAc; MIS; |
| CON: Contact | REQ; COR; can this occur? | ----- ----- | ----- ----- | ----- ----- | C;REQ; COR; |
| FAT: Fast Try | REQ; COR; fast HO failed | ----- ----- | NoAc; FAT; typical in HO | NoAc; FAT; typical in HO | C;REQ; COR; fast HO failed |
| FAC: Fast Contact | REQ; COR; fast HO failed | ----- ----- | NoAc; FAC; typical in HO | NoAc; FAC; typical in HO | C;REQ; COR; fast HO failed |
| KON: Konnect | NoAc; KON; may happen | C;DT;N; FAI; Misbehaviour! | ----- ----- | ----- ----- | C;DT;REQ;T1; COR; after Timeout: N |
| REK: Re_Konnect | NoAc; REK; may happen | C;DT;N;IT;B; FAI; Misbehaviour! | ----- ----- | ----- ----- | C;DT;REQ;IT;B;T1; COR; after Timeout: N |
| SOS: Sync_Lost | REQ;IT;B;T1; COR; after Timeout: N | ----- ----- | ----- ----- | NoAc; SOS; wait for Runout | C;REQ;IT;B;T1; COR; after Timeout: N |
| OPE: Operation | NoAc; OPE; typ. Final event | B; OPE; List not Ack_ed! | SYL1; OPE; 1: Alarm, go on | C;DT;SYL; SOS; 2: Alarm, stop! | NoAc; OPE; Typ. Final event |
| FAI: Failure | NoAc; FAI; typical | ----- ----- | ----- ----- | ----- ----- | NoAc; FAI; don't trust! |

15 Codec Mismatch Resolution and Codec Optimization⁷

It is not mandatory for a MSC to support the resolution of Codec Mismatch or the Codec Optimization. In that case the Local_Codec_List shall include only the Local_Used_Codec. However, in the optional case, if a MSC sends a Local_Codec_List that includes more than the Local_Used_Codec, then it is mandatory for that MSC to support the resolution of Codec Mismatch or the Codec Optimisation, considering the reported Codec_Types.

The determination of the Local_Codec_List (i.e. the list of all Codecs supported by the local radio leg, consisting of the local MS, the local MSC and the local radio resources) and the communication of the TRAU with the local MSC, is a MSC specific matter and is outside the scope of this specification. However, only Codec_Types that are real alternatives, considering all resources, shall be reported within the Local_Codec_List. The Local_Codec_List shall be updated and resent as soon as these local resource conditions have changed, if the MSC wants a have these new conditions considered within the Codec Mismatch Resolution or Codec Optimisation.

Whenever a new Distant_Codec_List or a new Local_Codec_List becomes available, then the MSC shall attempt to resolve the Codec_Mismatch or optimize the Codec_Type as soon as possible by following the rules outlined in Table 29 41 and shall perform a subsequent intra cell HO to the new Local_Used_Codec.

Table 41: Rules for Resolving Codec Mismatch in ANSI-136

| Radio Leg 1 (T1) | Radio Leg 2 (T2) | | | | | | | | | | | |
|------------------|------------------|--------|--------|-----|------------------|------------------|----------------|-------|------------------|----------------|------------------|-------|
| | US1 AC VC | US1 AC | US1 VC | US1 | AC US1 VC | AC US1 | AC VC | AC | VC US1 AC | VC AC | VC US1 | VC |
| US1 AC VC | = | = | = | = | T2=US1 | T2=US1 | T1=AC | T1=AC | T2=US1 | T1=AC T2=AC | T2=US1 | T1=VC |
| US1 AC | | = | = | = | T2=US1 | T2=US1 | T1=AC | T1=AC | T2=US1 | T1=AC T2=AC | T2=US1 | MIS |
| US1 VC | | | = | = | T2=US1 | T2=US1 | T1=VC T2=VC | MIS | T2=US1 | T1=VC | T2=US1 | T1=VC |
| US1 | | | | = | T2=US1 | T2=US1 | MIS | MIS | T2=US1 | MIS | T2=US1 | MIS |
| AC US1 VC | | | | | T1=US1 T2=US1 | T1=US1 T2=US1 | = | = | T1=US1 T2=US1 | T2=AC | T1=US1 T2=US1 | T1=VC |
| AC US1 | | | | | | T1=US1 T2=US1 | = | = | T1=US1 T2=US1 | T2=AC | T1=US1 T2=US1 | MIS |
| AC VC | | | | | | | = | = | T2=AC | T2=AC | T1=VC | T1=VC |
| AC | | | | | | | | = | T2=AC | T2=AC | MIS | MIS |
| VC US1 AC | | | | | | | | | T1=US1 T2=US1 | T1=AC T2=AC | T1=US1 T2=US1 | = |
| VC AC | | | | | | | | | | T1=AC T2=AC | = | = |
| VC US1 | | | | | | | | | | | T1=US1 T2=US1 | = |
| VC | | | | | | | | | | | | = |

⁷ Codec resolution is not supported by ANSI-95 in the current stage. It will be introduced at a later stage.

The first columns of Table 41 contain in each cell a definition of the Used_Codec in Radio Leg 1 followed on the next line by the list of supported Codecs. The first row contains similar information for the other, Radio Leg 2. The matrix elements indicate the change to be made in the Used_Codec. For example T2=VC means that Radio Leg 2 shall use the Vselp Codec. The grey shaded area is intentionally left blank, since it would contain redundant information. The '=' sign indicates that no mismatch is present. The 'MIS' indicates that mismatch can not be resolved. The light (green) shaded areas represent no Codec mismatch, but in several cases double sided HO is recommended to gain speech quality. Notation: US1 := US 1 Codec, 12.2 kBit/s; VC: Vselp Codec, 7.95 kBit/s; AC: Acelp Codec, 7.40 kBit/s

16 Annex A (Normative): Inband Signalling Protocol: Generic Structure

Scope

Inband Signalling Messages (IS Messages) can be used to construct a specific IS Protocol for the communication between telecommunication entities for various purposes. The original purpose is to establish tandem free operation of mobile-to-mobile calls in mobile networks. The IS Messages provide communication channels inside the speech signal paths between the speech transcoders.

In addition IS Messages allow the control of equipment within the speech signal paths between these telecommunication entities (e.g. speech transcoders). These equipments are termed „In Path Equipments“ (IPEs).

Annex A defines the generic structure of these IS Messages and rules for the IS_Sender.

Annex B defines the generic rules with respect to these IS Messages for the IPEs.

Annex A is mandatory for TFO_TRAU Equipment and informative for IPEs.

Annex B is informative for TFO_TRAU Equipment.

Annex B shall be followed by IPEs, which want to be compatible to IS Messages.

16.1 Generic Structure of Inband Signalling Messages

All IS Messages follow a set of design rules, or a generic structure, which allow to identify and bypass them by IPEs without detailed knowledge of the IS Protocol served. The principle of the IS Protocol shall in that sense be future proof: it can be enhanced and extended to other applications without modifying the IPEs.

The IS Messages replace some of the LSBs of the PCM samples of the Speech, Audio or Modem signal.

By construction the introduced signal distortion is practically inaudible in case of Speech signals.

Modem signals will in most cases not be affected with respect to their data transmission performance.

16.1.1 Frequency and Order of Bit Transmission

IS Messages are transferred within the Least Significant Bit (LSB) of PCM samples on 64 kBit/s links, by replacing the LSB of every 16th consecutive PCM sample with one bit of the IS Message (16_PCM_Sample_Grid).

This is equivalent to an average bit rate of 10 bit per 20 ms or 500 bits per second. See Figure 12:

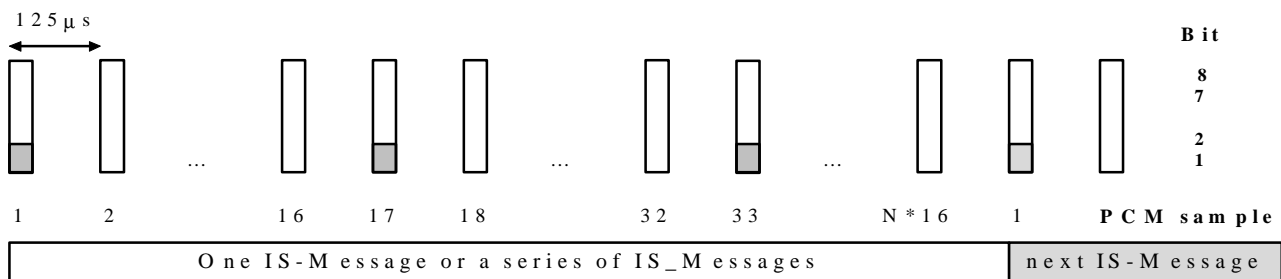


Figure 13: Inband Signalling Structure

A vertical bar denotes an 8-bit PCM sample, the shadowed box in bit 1 (LSB) represents an inserted bit of the IS-Message. By definition each IS Message "occupies" an integer multiple of 16 PCM samples. Especially the 15 PCM samples after the last inserted bit of an IS Message "belong" still to that IS Message.

All IS Messages, whichever type, have by construction "0"-Bits at every 10th position, starting with position 1, 11, 21 and so on. This "0"-Bits occur therefor regularly every 20 ms and may be used for synchronization purposes.

Each IS Message consists of an IS_Header followed by an IS_Command_Block. Most IS Messages have a number of further IS_Extension_Blocks. Figure 13 shows an example with two IS_Extension_Blocks.

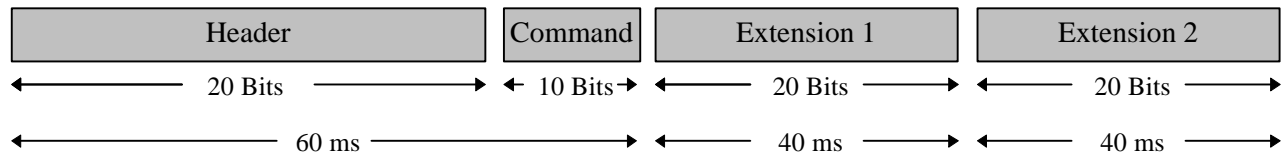


Figure 14: Example for IS Message with two IS_Extension_Blocks

The MSB of each constituent field is transmitted first. The IS_Header is transmitted first, followed by the IS_Command_Block and - if applicable - any further IS_Extension_Block(s).

By construction all IS Messages do have lengths of integer multiples of 10 bits, thus occupying integer multiples of 160 PCM samples, thus lasting integer multiples of 20 ms. The shortest IS Message has a length of 60 ms.

16.1.2 IS_Header

The IS_Header consists of a 20-Bit long sequence, as defined in Figure 14:

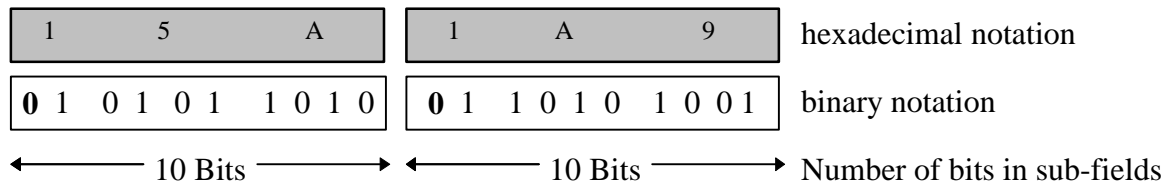


Figure 15: Structure of the 20 bit IS_Header

16.1.3 IS_Command_Block

The IS_Command identifies the IS Message and/or serves for the control of IPEs. The names of the IS_Commands and their codes in hexadecimal notation in the IS_Command_Block are given in the Table 42.

Table 42: Defined IS_Commands

| Index | Command | Code | Meaning / Action |
|-------|-------------|------------------------|--|
| | | hexadecimal Nibble 1-3 | |
| 0 | reserved | 0x000 | no extension |
| 1 | REQ | 0x05D | Denotes an IS_REQ Message, with extension |
| 2 | ACK | 0x0BA | Denotes an IS_ACK Message, with extension |
| 3 | IPE | 0x0E7 | Denotes an IS_IPE Message, with extension, i.e. an IS_TRANS or the IS_NORMAL Message |
| 4 | FILL | 0x129 | Denotes the IS_FILL Message, no extension |
| 5 | DUP | 0x174 | Denotes the IS_DUP Message, no extension |
| 6 | SYL | 0x193 | Denotes the IS_SYL Message, no extension |
| 7 | reserved | 0x1CE | no extension |

All other values are reserved for future use.

Each IS_Command is protected by the binary, systematic (9,3) block code with generator polynomial $g(x) = x^6 + x^4 + x^3 + x^2 + 1$. The minimum Hamming distance of this code is $d_{min} = 4$, which allows the correction of up to one bit error within each code word of length 9 bits.

The first bit (MSB) of the IS_Command_Block is defined to be "0", for synchronisation purposes, see Figure 15.

Table 42 gives the hexadecimal notation of the complete IS_Command_Block.

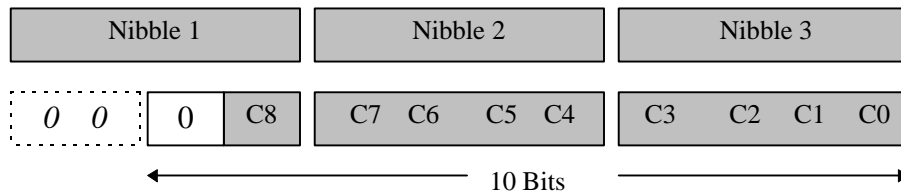


Figure 16: General Construction of an IS_Command_Block

16.1.4 IS_Extension_Block(s)

Most IS Messages have one or more IS_Extension_Block(s). Each IS_Extension_Block is 20 bits long and shall consist of two "0"-Synchronization_Bits at position 1 (MSB) and 11, a 16-bit Information_Field (split into two fields of 9 and 7 bits, respectively) and a 2-bit Extension_Field (EX), see Figure 16:

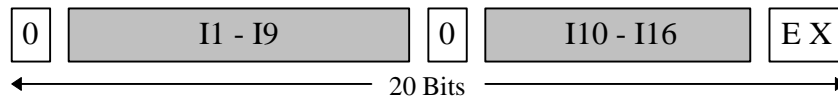


Figure 17: General Construction of an IS_Extension_Block

The Extension_Field indicates if an other IS_Extension_Block is following (EX := "1.1") or not (EX := "0.0"). All other codes are reserved. This may be used to detect transmission errors within the Extension_Field.

16.2 Detailed Specification of IS Messages

16.2.1 IS_REQ Message

With the IS_REQ Message an IS_Sender can test, if there is an IS Partner and indicates that it is willing to negotiate. IS_REQ is used to initiate the IS Protocol or to indicate changes in the configuration, etc. IS_REQ has at least one IS_Extension_Block, containing the IS_System_Identification. (see A.5). Other IS_Extension_Blocks may follow, see Figure 17.

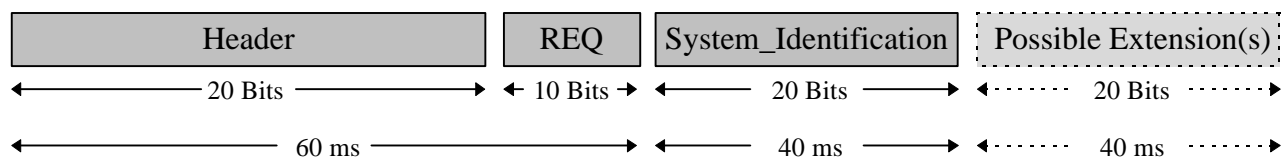


Figure 18: General Construction of an IS_REQ Message

In general an IS_REQ Message shall be as short as possible. Special care must be taken in the design of the IS_Extension_Blocks to avoid audible effects, since sometimes an IS_REQ Message may be transmitted for quite some time (several seconds).

16.2.2 IS_ACK Message

With the IS_ACK Message an IS Partner typically answers an IS_REQ Message or an IS_ACK Message. It can also be used to submit further information to the other IS Partner. IS_REQ and IS_ACK are the main message types between IS Partners.

The IS_ACK has at least an IS_Extension_Block containing the IS_System_Identification (see A.5). Other IS_Extension_Blocks may follow, see Figure 18.

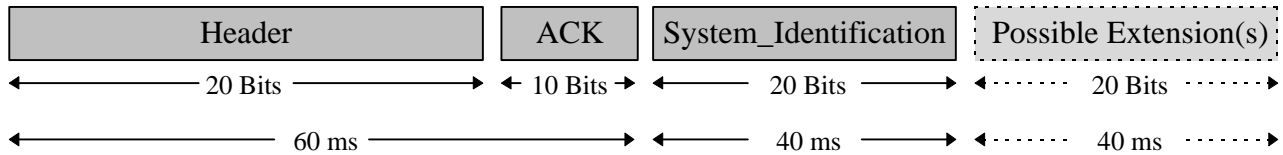


Figure 19: General Construction of an IS_ACK Message

No specific design constraints with respect to audibility exist, since IS_ACK is typically not sent very often.

16.2.3 IS_IPE, IS_TRANS and IS_NORMAL Messages

The IPE command denotes IS_IPE Messages. An IPE shall always look for this type of message and follow the instruction. An IS_Sender shall use this IS_IPE Message to command all IPEs into a specific mode of "Bit Transparency". This Message has one IS_Extension_Block, indicating the requested IPE_Mode. See Figure 19.

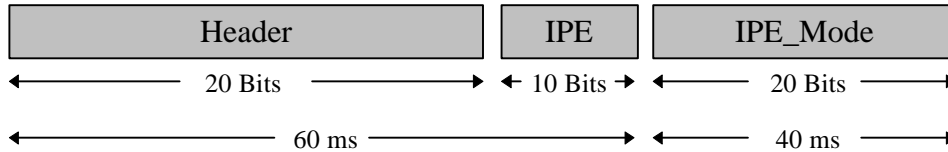


Figure 20: General Construction of an IS_IPE Message

No specific design constraints with respect to audibility exist, since IS_IPE is typically not sent very often. Table 43 defines 16 out of 32 possible IPE_Commands. The other codes are reserved for future extensions.

Table 43: Defined IPE_Modes

| Index | IPE_Mode | Code | MEANING / ACTION |
|--------|-------------|--------------------------|---|
| | | hexadecimal Nibble 1 - 5 | |
| 0 | Normal | 0x0000 | Normal Operation |
| 1 | Trans_1_u | 0x044DC | pass 1 LSB; 7 upper Bits are used |
| 2 | Trans_2_u | 0x089B8 | pass 2 LSBs; 6 upper Bits are used |
| 3 | Trans_3_u | 0x0CD64 | pass 3 LSBs; 5 upper Bits are used |
| 4 | Trans_4_u | 0x11570 | pass 4 LSBs; 4 upper Bits are used |
| 5 | Trans_5_u | 0x151AC | pass 5 LSBs; 3 upper Bits are used |
| 6 | Trans_6_u | 0x19CC8 | pass 6 LSBs; 2 upper Bits are used |
| 7 | Trans_7_u | 0x1D814 | pass 7 LSBs; 1 upper Bit is used |
| 8 | Transparent | 0x22CE0 | Full Transparent Mode for all eight bits |
| 9 | Trans_1 | 0x2683C | pass 1 LSB; 7 upper Bits are free and unused |
| 10 | Trans_2 | 0x2A558 | pass 2 LSBs; 6 upper Bits are free and unused |
| 11 | Trans_3 | 0x2E184 | pass 3 LSBs; 5 upper Bits are free and unused |
| 12 | Trans_4 | 0x33990 | pass 4 LSBs; 4 upper Bits are free and unused |
| 13 | Trans_5 | 0x37D4C | pass 5 LSBs; 3 upper Bits are free and unused |
| 14 | Trans_6 | 0x3B028 | pass 6 LSBs; 2 upper Bits are free and unused |
| 15 | Trans_7 | 0x3F4F4 | pass 7 LSBs; 1 upper Bit is free and unused |
| 16 | reserved | 0x41D1C | reserved |
| 17..31 | reserved | reserved | reserved |

The IPE_Mode is protected by the binary, systematic (16,5) block code with generator polynomial $g(x) = x^{11} + x^7 + x^5 + x^4 + x^2 + x + 1$. The minimum Hamming distance of this code is $d_{min}=7$, which allows the correction of up to 3 bit errors within each code word of length 16 bits.

Bits 1 (MSB) and 11 are the synchronisation bits and set to "0", see Figure 20. The EX field is set to "0.0" in all currently defined IPE_Modes, i.e. no further IS_Extension_Block is following.

Table 43 defines the coding in hexadecimal notation for the complete IPE_Mode_Extension_Block, with EX := 00.

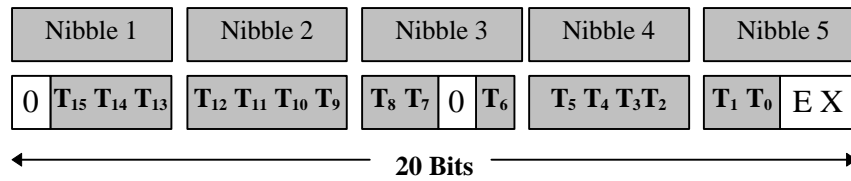


Figure 21: IPE_Mode_Extension_Block for the IS_IPE Message

An IS_IPE Message containing the NORMAL command is termed **IS_NORMAL Message**.

An IS_IPE Message containing a TRANS_x command is termed **IS_TRANS_x Message**.

An IS_IPE Message containing a TRANS_x_u command is termed **IS_TRANS_x_u Message**.

The latter two are sometimes also termed **IS_TRANS Message**, if the details are not important.

The behaviour of IPEs, when receiving such commands, is described in Annex B.

The first IS Message in a series is often "swallowed" by IPEs (see Annex B). An IS_IPE Message must therefore never be the first message of a series of IS Messages, i.e. it shall be sent as an isolated IS Message or after a (sufficiently long) uninterrupted IS Protocol.

16.2.4 IS_FILL Message

The IS_FILL Message has no IS_Extension_Block and no specific meaning. An IS_Sender can use the IS_FILL Message to fill a temporary gap in the protocol flow. This may be important to keep all IPEs in synchronization and open for further IS Messages. See Figure 21. An IS_FILL Message shall also be used by the IS_Sender to resynchronize all IPEs in case of a phase shift of the Keep_Open_Indication.

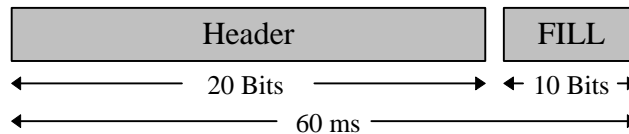


Figure 22: Construction of the IS_FILL Message

IS_FILL is designed in a way that multiple repetitions cause minimal audible effects.

16.2.5 IS_DUP Message

The IS_DUP Message may be used between IS Partners to indicate an half duplex mode. It may be especially important in HO situations. The IS_DUP Message has no IS_Extension_Block, see Figure 22.

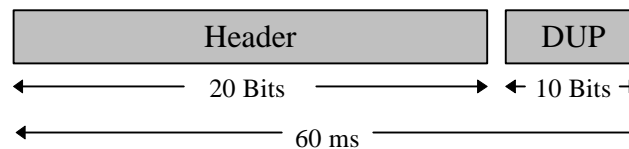


Figure 23: Construction of the IS_DUP Message

16.2.6 IS_SYL Message

The IS_SYL Message may be used between IS Partners to indicate the loss of synchronisation. It may be especially important in HO situations. The IS_SYL Message has no IS_Extension_Block, see Figure 23.

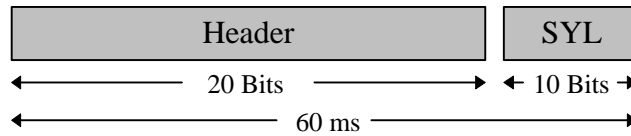


Figure 24: Construction of the IS_SYL Message

16.3 Keep_Open_Indication

In Transparent_Mode, i.e. after properly receiving an IS_TRANS Message, all IPEs shall monitor the bypassing bit stream for the Keep_Open_Indication (definition see below). If this Keep_Open_Indication is not seen for some time, then the IPEs shall fall automatically back into normal operation, i.e. the mode of operation before the IS_TRANS Message. This automatic fall back shall have the same effect as the IS_NORMAL Message would have. By definition the Keep_Open_Indication is a continuous bit stream of one "0"-Bit in the LSB of every 160th PCM sample, i.e. every 20 ms. At least one "1"-Bit must be present within the LSBs of the other 159 PCM samples. See Figure 24.

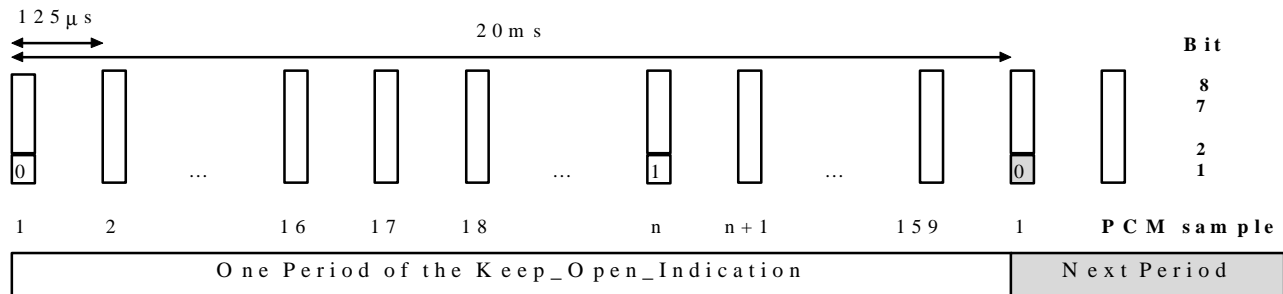


Figure 25: Keep_Open_Indication

The "0"-Bit stream of the Keep_Open_Indication shall always be present as long as the IPEs need to be in Transparent_Mode.

The Keep_Open_Indication shall be in phase with the preceding IS Messages., i.e. the first bit of the Keep_Open_Indication shall be in the position of the first bit of the (hypothetical) next IS Message. In fact, the IS Messages themselves contain this Keep_Open_Indication by definition.

In case of a known phase shift of the Keep_Open_Indication, the IS_Sender has to send at least one IS Message, which defines the new phase position of the Keep_Open_Indication. If no other IS Message is to be sent, then the IS_FILL Message shall be used. If an IS Message longer than 160 ms is scheduled for transmission, then an IS_FILL Message should be inserted before, to guarantee fast resynchronization of the IPEs.

16.4 Rules for Sending of IS Messages

IS Messages replace some bits of the PCM samples and therefore cause a minimal signal distortion. Therefore IS Messages shall be used with care and not longer than necessary. The IS Protocol is kept to a minimum to avoid unnecessary complexity. One basic assumption is that only one IS Protocol is active at a time between two IS Partners.

Only specific telecommunication entities shall be allowed to initiate IS Protocols. They are called **IS_Active** or active IS Partners. In principle these shall only be terminal devices or their "representatives" within the network. Examples are ISDN-Terminals, Speech-Servers, TRAU's (in mobile networks as representatives of the MSs).

Other telecommunication entities shall only react on IS Protocols. They are called **IS_Passive**. Most IPEs are of this type. They bypass the IS Messages, they obey the IS_IPE Messages, but they never initiate IS Messages.

Other telecommunication entities are IS_Passive by default. But if they receive IS Protocols that they can understand, then they may become IS_Active and start to initiate IS Protocols. They thus become active IS Partners and shall take care that only one IS Protocol is active on both of their sides. They are called **IS_Responsive**. Examples are TCMEs.

Active IS Partners shall send

either continuous sequences of IS Messages without interruption of the 16_PCM_Sample_Grid:

- or isolated IS Messages with same message lengths;

- or isolated IS Messages with sufficient distance between them, if shorter IS Messages follow longer IS Messages. The latter case is important, because shorter isolated IS Messages travel faster through IPEs than longer ones, see annex B. As said above, after initialization of an IS Message sequence, no interruption of the 16_PCM_Sample_Grid shall occur within the sequence. Adjustments of the phase position of the Keep_Open_Indication shall be done only after the IS_TRANS Message by inserting the necessary number *n* (with $0 < n < 160$) of "1" Bits (termed "T_Bits") into the LSBs of the PCM samples that have to be skipped. The first PCM sample for this insertion of T_Bits is the one where the next regular IS Message or next regular Keep_Open_Indication would begin. At the new phase position the next IS Message or the IS_FILL Message shall be sent, to allow IPEs to resynchronize fast. See Figure 25.

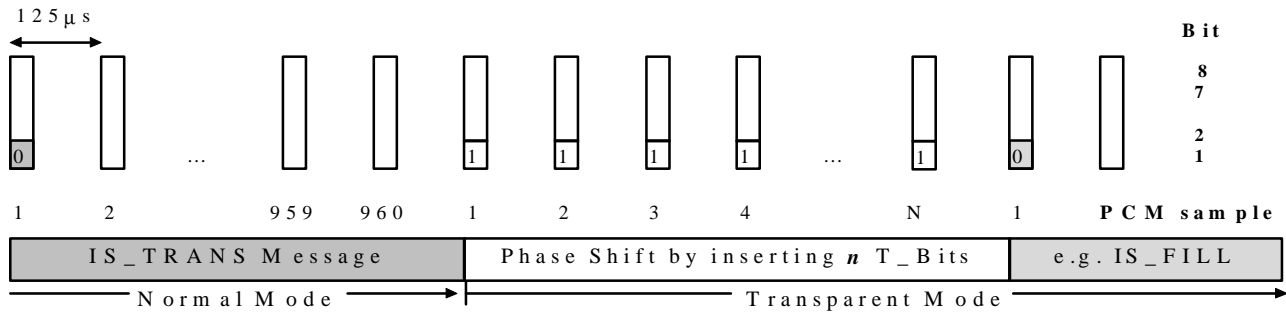


Figure 26: Phase Shift of the 16_PCM_Sample_Grid by inserting T_Bits

Similarly, the adjustment of the phase between two Keep_Open_Indications shall be done by inserting the necessary number of T_Bits and by sending an IS Message - preferably, but not necessarily - the IS_FILL. Finally a "negative" phase adjustment between two Keep_Open_Indications shall be allowed by shortening the cycle by a maximum of 2 PCM samples and sending an IS Message (see above) at the new phase position.

16.5 IS_System_Identification_Block

The IS_System_Identification_Block is a mandatory IS_Extension_Block for the IS_ACK and IS_REQ messages with the 16-bit Information_Field containing the IS_System_Identification. It identifies the system within which the message is generated. Table 44 shows the defined IS_System_Identification codes.

Table 44: Defined IS_System_Identification Codes

| System | Code (in hex) |
|-----------------------------|--|
| GSM: 0x53948 | either 0x53948, if EX == "0.0" or 0x5394B, if EX == "1.1" |
| ANSI-136: 0x53414 | either 0x53414 if EX == "0.0" or 0x53417 if EX == "1.1" |
| ANSI-95: 0x528AC | either 0x528AC if EX == "0.0" or 0x528AF if EX == "1.1" |
| all other (253) code points | reserved |

The only defined codes so far are GSM_Identification, ANSI_95_Identification and ANSI_136_Identification, see also Figure 26.

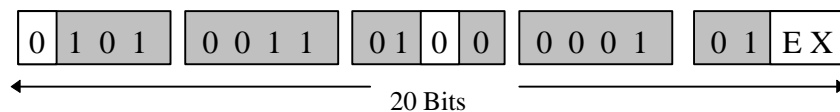


Figure 27: IS_System_Identification for ANSI-136

All other codes are reserved. Further IS_System_Identification Codes for other systems shall be defined in a way that the audibility is minimal and the hamming distance to the already defined once is maximal.

The IS_System_Identification is protected by the binary, *systematic* (16,8) block code with generator polynomial $g(x) = x^8 + x^7 + x^6 + x^4 + x^2 + x + 1$. The minimum Hamming distance of this code is $d_{min}=5$, which allows the correction of up to 2 bit errors within each code word of length 16 bits.

Code word 0x0000 = 0000.0000.0000.0000 is per definition used for GSM.

Code word 0x01D3 = 0000.0001.1101.0111 is per definition used for ANSI-136.

Code word 0x0279 = 0000.0010.0111.1001 is per definition used for ANSI-95.

The first, upper eight bits represent the systematic part (underlined), the following eight bits the redundant part of the Code words.

Please note: the *systematic part* is also used within the TFO Frames of GSM, ANSI-136 and ANSI-95 systems for System_Identification (S1..S8), see main part of the document, chapter 5.

The resulting 16 bits are placed into the IS_System_Extension_Block as shown in Figure 26 and then the whole 20 bit word is additionally EXORed with the fixed code word 0x53948 to minimise audible effects.

17 Annex B (Informative): In Path Equipment: Generic Rules and Guidelines

Scope

Inband Signalling Messages (IS Messages) can be used to construct a specific IS Protocol for the communication between telecommunication entities for various purposes. The original purpose is to establish tandem free operation of mobile-to-mobile calls in mobile networks. The IS Messages provide communication channels inside the speech signal paths between the speech transcoders.

In addition IS Messages allow the control of equipment within the speech signal paths between these telecommunication entities (e.g. speech transcoders). These equipments are termed „In Path Equipments“ (IPEs).

Annex A defines the generic structure of these IS Messages and rules for the IS_Sender.

Annex B defines the generic rules with respect to these IS Messages for the IPEs.

Annex A is mandatory for TFO_TRAU Equipment and **informative for IPEs**.

Annex B is informative for TFO_TRAU Equipment.

It shall be followed by IPEs, which want to be compatible to IS Messages.

17.1 Types of In Path Equipment

The term "In Path Equipment" (IPE) is used for any telecommunication equipment within the (64 kBit/s) transmission path for the speech signal between two entities, which want to communicate via IS Messages, i.e. the IS Partners.

In modern telecommunication networks most of these IPEs are digitally transparent for the complete 64 kBit/s data stream all the time after call establishment until call release. These IPEs are optimal and need no consideration here.

Some IPEs are most of the time digitally transparent, but disturb the link every now and then. Examples are:

- switches, which interrupt the link during HO;
- switches, which insert a kind of conference bridge for a short while during HO;
- links, which do octet deletions or insertions (octet slips);
- DTMF generators, which insert DTMF tones sometimes for a short while; and more.

Other IPEs are digitally transparent in one direction, but not in the other. Examples are:

- DTMF generators (again), which insert the DTMF tones only in one direction;
- Network Echo Cancellers (NEC), which let the signal pass unaltered towards the PSTN, but cancel the echo; and more.

Other IPEs are semi-transparent, i.e. let most or some of the bits pass, but not all. Examples are:

- A/ μ _Law converters;
- μ /A_Law converters; and
- especially the tandem connection of A/ μ _Law and μ /A_Law converters, or vice versa.
- links, which insert inband signalling by bit stealing (T1 links); and more.

Other IPEs are not transparent at all to the digital bit stream, although the speech signal pass more or less unaltered.

Examples:

- level shifters, which adjust the signal levels, e.g. between national networks;
- DCMEs (Digital Circuit Multiplication Equipment), which compress the bit stream by encoding/decoding the speech signal for cost efficient transmission; and more.

Many of these IPEs - for some time - will be not compliant with the IS Message principle described above. The IS Messages will not pass these non-compliant IPEs or not in both directions, or not always. Care must be taken to identify situations where IPEs are part-time-transparent or semi-transparent, when applying IS Messages. Other IPEs - at some point in time in the future - will be compliant to the IS Message principle. The rules they have to fulfil are described below.

17.2 IS_Compliant IPEs

17.2.1 Typical IPEs are IS_Passive

General: An IPE shall *never* actively initiate the exchange of IS Messages. The active initiation is only done by terminals or their "representatives". This avoids uncontrolled and unnecessary fluctuation of IS Messages within the network. Most IPEs shall never actively respond to IS Messages by sending other IS Messages. They are called *IS_Passive*. They need not and do not understand the IS Protocol, but let it just pass unaltered and obey the relevant IS_IPE Messages. Some IPEs may, however, respond on received IS Messages, modify these and/or respond with own IS Messages, *if* they understand the IS Protocol and can take or bring advantage to the overall system performance or system quality. These IPEs are called *IS_Responsive*. Examples are TFO-specific Digital Circuit Multiplication Equipments (TCMEs), which reduce transmission costs without degrading the speech quality. These IPEs may be able to step into the IS Protocol, interpret and respond to it and modify the speech signal in an system_compliant way. Thus they become *IS_Active* Partners themselves.

17.2.2 IS Message_Transparency

When commanded into a Transparent Mode, the IPEs are fully transparent at least for the LSBs in all PCM samples. Therefore the following rules are needed only and only do apply for the IPEs, when in Normal_Mode:

IPEs shall let the IS Messages bypass, respectively re-insert them, from their input to their respective output. They shall not alter them, nor do any kind of error correction. Exceptions are the IS_Responsive IPEs.

17.2.2.1 First IS Message

During its **Normal_Mode** an IS_Compliant IPE shall always monitor the incoming PCM data stream for the occurrence of the IS_Header sequence. If the IS_Header is detected after a period without IS Messages, the IPE shall store the following IS_Command and IS_Extension_Block(s). During reception of this first IS Message, the normal operation of the IPE is maintained with the consequence that the first IS Message may not appear at the output of the IPE.

17.2.2.2 IS Messages within a Sequence

All further IS Messages which follow directly after the first detected IS Message in the same phase position shall be passed unaltered to the output of the IPE with exactly that delay the IPE would later introduce when commanded into Transparent_Mode by one of the IS_TRANS commands, see Figure 27.

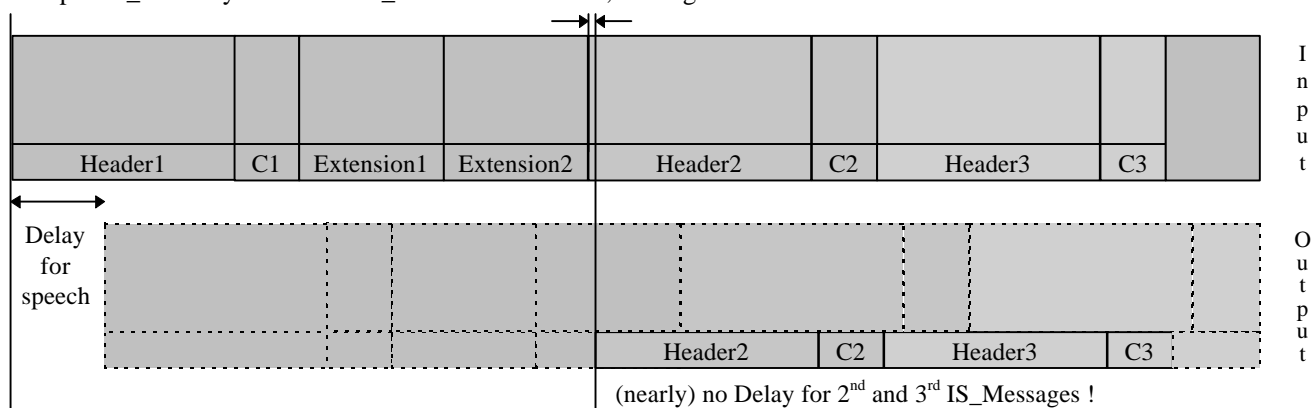


Figure 28: Transparency and Delay for first and following IS Messages

The upper row symbolizes the speech signal at the input of the IPE, with the PCM samples drawn vertically and the IS Messages inserted into the LSBs. The lower row symbolizes the speech signal at the output of the IPE. The vertical lines denote the boundaries of the IS Message elements.

Figure 27 shows an example where the first IS Message is detected, but not passed through. The distortion caused by the first IS Message is still "somehow" there (indicated by the empty dashed boxes in the LSB), but the message is destroyed.

The second and third IS Messages are passed through unaltered. Note, however, that the delay of the speech signal is (in this example) substantially higher than the delay of the IS Messages. They travel faster than the speech signal through this IPE.

17.2.2.3 Isolated IS Message

In cases where the first detected IS Message is not immediately followed by further IS Messages, the IPE shall insert this first IS Message (which the IPE has stored) into its output PCM bit stream, with exactly the delay and phase position a second IS Message would have, see Figure 28, which shows an example where an isolated IS Message is travelling through an IPE.

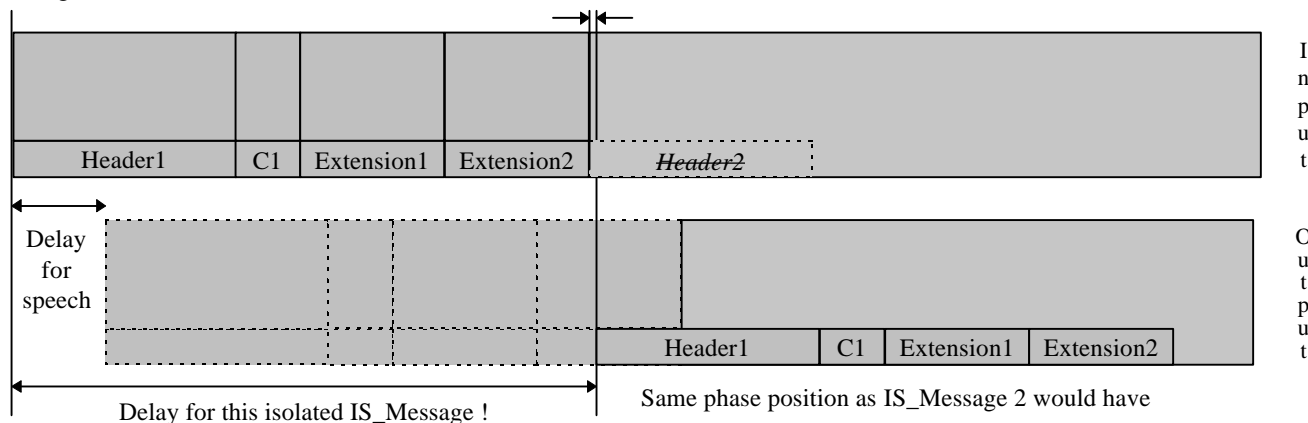


Figure 29: Transparency and Delay for an isolated IS Message

Note that the delay of an isolated IS Message is depending on its own length! Longer IS Messages will have more delay, shorter less. It could - in principle - happen that a second, shorter isolated IS Message would "bypass" the first longer IS Message - with the consequence that the first one would be destroyed. This is especially important when there are several IPEs in the path, since the delay effects accumulate. Therefore it is not allowed to send shorter isolated IS Messages too close after longer IS Messages. IS Messages with same length have no restriction.

In summary: the first IS Message in a series of IS Messages is "swallowed" by an IPE, while all the following IS Messages pass unaltered and with minimal delay. If an IS Message occurs isolated, then it is not swallowed, but delayed by exactly its own length. The latter mechanism ensures that isolated IS Messages can pass through an unlimited number of IPEs.

17.2.2.4 Check if IS Message is following

The checking, whether an other IS Message is following or not is done "on the fly", i.e. bit by bit. This is possible due to the fact that all messages begin with exactly the same IS_Header. The decision, whether an IS Message is an isolated message or the first message in a series, can be done latest after the last bit of the (next) IS_Header. See Figure 28. Consequently: after detection of the first IS Message, the IS_Header is in any case inserted at the output in the correct position, regardless, whether a second message follows or not.

17.2.3 IPE State Representation

Concerning the IS Protocol, an IPE can be described with five major States in two main Modes, where the States describe the IPE with respect to the IS Protocol and the Modes describe the IPE with respect to the operation on PCM data. Figure 29 shows a graphical representation of the State diagram of an IPE.

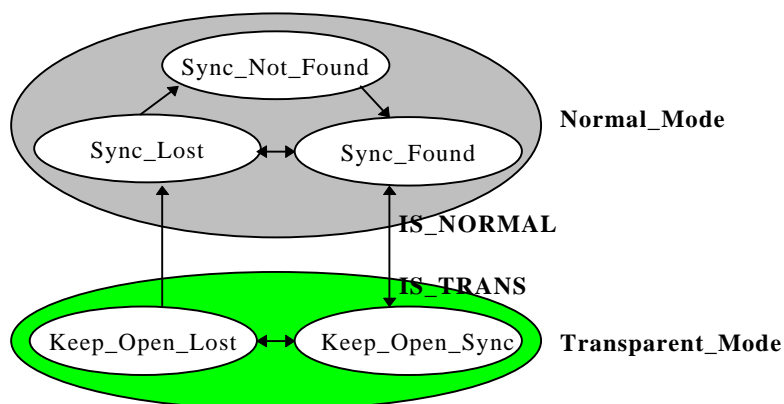


Figure 30: Principle of a State Diagram of an IPE

Some Definitions:

An IS Message shall be recognized as "error-free", if no error can be detected, neither within the IS_Header, nor in the IS_Command nor in any IS_Extension_Block.

An IS Message shall be recognized as "single-error", if no more than one bit position differs in the IS_Header or the IS_Command_Block or the IPE_Mode_Block or one EX-field or one Sync bit.

An IS Message shall be recognized as "correctable", if the phase position is as in preceding IS Messages and:

- no more than 2 bit position differs in the IS_Header; and
- no more than 1 error is detected within the IS_Command_Block; and
- no more than 3 errors are detected within the IPE_Mode_Block; and
- no more than 0 error is detected within the EX-field(s); and
- no more than 1 error is detected within the Sync-Bit(s); and
- the total number of detected errors is not higher than 3.

IS Messages, which are error-free, single-error or correctable are also called "valid" IS Messages.

An IS Message shall be recognized as "present", if the phase position is as in preceding IS Messages and:

- no more than 4 bit position differs in the IS_Header and
- no more than 2 errors are detected within the IS_Command_Block; and
- no more than 3 errors are detected within the IPE_Mode_Block; and
- no more than 1 error is detected within the EX-field(s); and
- no more than 2 errors are detected within the Sync-Bit(s); and
- the total number of detected errors is not higher than 4.

Sequences, which differ in more than "present" are not recognized as IS Messages at all ("not_present").

Note that the insertion of T_Bits may change the phase position of an IS Message. The IS Message shall in that case be classified after the removal of the T_Bits.

An octet slip may also change the phase position of an IS Message. If an error-free or a single-error IS Message can be found after considering a hypothetical octet slip (± 1 sample), then it may be regarded as error-free or single-error and the new phase position shall be regarded as valid, if no valid or present IS Message can be found at the old phase position.

17.2.4 IPE in Sync_Not_Found

After start-up or after a long interruption of the IS Protocol an IPE is in Normal_Mode, performing its normal operation. IS Messages have not been found and consequently no bypassing of IS Messages is performed.

The algorithm for initial synchronization shall be able to detect each single IS Message, especially the first or an isolated one. An IPE shall always, during Normal_Mode and during Transparent_Mode, search for the IS_Header and consequently for complete IS Messages. When found, it can be assumed that with high probability the following IS Messages and the Keep_Open_Indication will stay within the found "grid" or "phase" of every 16th PCM sample, the **16_PCM_Sample_Grid**.

An IPE transits from Sync_Not_Found into Sync_Found, if and only if an error_free IS Message is detected. Then the IPE lets the following IS Messages bypass, as described above.

If the first IS Message is an error_free IS_TRANS Message, then the IPE transits directly into the Transparent_Mode.

17.2.5 IPE in Sync_Found

The IPE continues its normal operation, but opens an "*IS_Door*" every 16th LSB for the bypassing IS Messages. An IPE shall regard sync as continued, i.e. stay in Sync_Found, if after each IS Message another valid IS Message follows within the same phase position, i.e. within the 16_PCM_Sample_Grid. For any deviations from a valid IS Message, the IPE transits to Sync_Lost. If an error_free or correctable IS_TRANS is received in Sync_Found, then the IPE transits into the Transparent_Mode.

17.2.6 IPE in Sync_Lost

In Sync_Lost, an IPE shall search for IS Messages on all positions as for initial synchronisation. In parallel, an IPE shall bypass not_valid, but present IS Messages at the found phase position for a maximum of one second. An IPE shall close the IS_Door after that, if no valid IS Message is following, i.e. transit into Sync_Not_Found. A single valid IS Message brings the IPE back into Sync_Found. As soon as the IPE detects in Sync_Found or in Sync_Lost a single or more deviations from an error_free IS Message, then the IPE may optionally open the IS_Door also at positions ±1 around the present (0) phase position for a maximum of one second] to allow other IPEs in the path for parallel re-synchronization. See Figure 30. The IPE may try to find a continuation of the disturbed IS Message at these 3 positions. If the IPE can detect an error-free or a single-error IS Message in this way, then it shall accept the new phase position, if no IS Message can be found at the old phase position anymore.

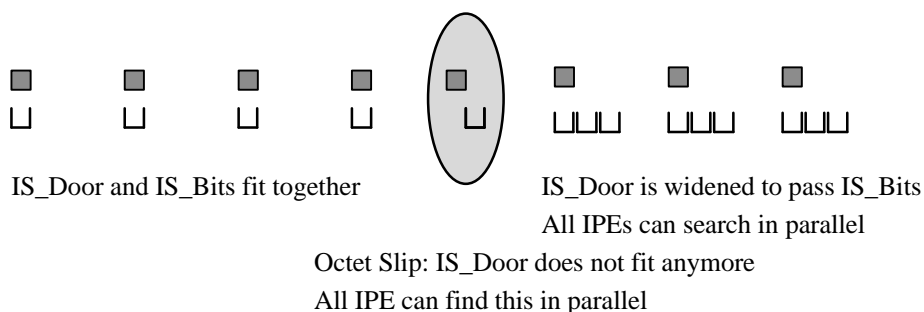


Figure 31: Handling of octet slip for fast and parallel re-synchronization of all IPEs (optional)

17.2.7 IPE in Keep_Open_Sync

The IPE enters this State by receiving a valid IS_TRANS Message. This is the main State of the Transparent_Mode. It depends on the specific IPE, if this Transparent_Mode is active only for the commanded direction (that is the default assumption) or in both directions (because for a specific IPE it might be useless or impossible to maintain Normal_Mode in one direction and Transparent_Mode in the other one). The IPE shall bypass the commanded LSBs and handle the upper bits accordingly (IPE specific). The IPE shall search in parallel for IS_IPE Messages (IS_TRANS, IS_NORMAL) and transit - if necessary - to Normal_Mode or an other Transparent_Modes (other number of transparent LSBs). The IPE shall monitor the bypassing bit stream for the Keep_Open_Indication and accept the Keep_Open_Indication only at the phase position defined by the preceding IS Message. If the Keep_Open_Indication is not seen anymore then the IPE transits into Keep_Open_Lost.

17.2.8 IPE in Keep_Open_Lost

The IPE shall continue its operation in Transparent_Mode and Keep_Open_Lost for a maximum of one second before it shall return to Normal_Mode. During that time the IPE shall try to resynchronize either by finding an IS Message or by finding the Keep_Open_Indication at positions ±1 and 0 around the present phase position (handle of Octet Slip). The IPE may take advantage of the fact that T_Bits are inserted or deleted by the IS_Sender in case of an intentional phase adjustment. An IS Message at any arbitrary phase position followed by a valid Keep_Open_Indication is accepted as re-defining the Keep_Open phase position, if and only if the Keep_Open_Indication is no longer present at the old phase position.

A Keep_Open_Indication at a phase position ± 1 PCM sample interval around the old phase position is accepted as re-defining the Keep_Open phase position, if and only if the Keep_Open_Indication is no longer present at the old phase position.

The Keep_Open_Indication is *valid*, as long as at least 40 "0"-Bits are seen at the correct positions within a sliding window of length of one second. At least one "1"-Bit must be seen in between each pair of the expected "0"-Bits.

17.3 IPE Error Handling

The **first** IS_Message shall only be accepted, if there is no detectable error.

For all following IS_Messages it shall apply:

Errors in IS Messages shall be passed unaltered through the IPEs. This shall hold for all IS Messages.

Only error-free or correctable IS_IPE Message shall be applied by the IPE to its own operation. Other IS_IPE Messages shall be ignored, but bypassed.

17.4 IPE Transmission Delay

The transmission delay introduced by an IPE for the speech, audio or modem signal is in general different in Normal_Mode and Transparent_Mode. Some IPEs may have several different Normal_Modes with possibly different signal delays. IS Messages are transmitted within the regular 16_PCM_Sample_Grid. It is important that this regularity is not disturbed. Therefore care must be taken at the transition between these modes.

The transmission delay of a specific IPE is in general lower for IS Messages than for speech, audio or modem signals.

17.4.1 IPE Transmission Delay in Normal_Mode

The delay for IS Messages in Normal_Mode shall be identical to the delay in that Transparent_Mode, that follows after the first IS_TRANS Message. If different Transparent_Modes with different delays could follow, then the shortest delay of all possible Transparent Modes shall be selected for IS Messages in Normal_Mode.

If an IPE in Normal_Mode has to change its transmission delay, then this shall not affect the delay of the IS Messages.

17.4.2 IPE Transmission Delay in Transparent_Mode

In the majority of all cases the IPE will keep the transmission delay for the IS Messages in Normal_Mode also in Transparent_Mode for the transmission of the commanded transparent LSBs. IPEs which do not understand the IS Protocol shall never modify the transparent bits, so they are also not allowed to change delay.

Some IPEs, which understand a specific IS Protocol, may have even different Transparent_Modes and also here the transmission delays may differ. Examples are TCMEs.

If an IPE has to change its transmission delay at the transition from Normal_Mode to Transparent_Mode, then the IPE shall readjust the phase of the Keep_Open_Indication after transition into the Transparent_Mode with higher delay by inserting the relevant number of T_Bits after the first IS_TRANS Message and before the next IS Message. If no other IS Message is following, then the IS_FILL shall be inserted, obeying all other relevant rules of the specific IS Protocol (e.g. EMBED bit C5 in TFO Frames).

If an IPE has to change from one Transparent_Mode to an other one with a different transmission delay, then the IPE shall readjust the phase of the Keep_Open_Indication after transition into the new Transparent_Mode by inserting the relevant number of T_Bits. If no other IS Message is following, then the IS_FILL shall be inserted at the new phase position to mark the new grid position of the 16_PCM_Sample_Grid and to allow other IPEs to resynchronize, obeying all other relevant rules of the specific IS Protocol (e.g. EMBED bit C5 in TFO Frames).

17.5 Compliance to IS Messages

An IS_Compliant IPE shall be capable of interpreting and obeying the IS_IPE Messages.

It depends on the intelligence and task of an IPE, how many and which of the other IS Messages it needs to understand. The IPEs shall synchronise to all IS Messages, especially to find or re-find the Keep_Open_Indication. All IPEs shall resynchronize, if they see an IS Message in a new phase position, and if the synchronization can not be found in the old phase position anymore.

17.5.1 Compliance to IS_REQ and IS_ACK Messages

Most IPEs need not and do not understand these messages. They just synchronise to them and let them pass unaltered. Only IS_Responsive IPEs may take advantage. This is system specific and IPE specific.

17.5.2 Compliance to IS_NORMAL Message

The IPE shall act in response to the receipt of an IS_NORMAL Message such that:

The IPE shall synchronise to it. The message shall appear unchanged at the output of the IPE.

The IPE shall resume its Normal_Mode of operation for all data received subsequent to the IS_NORMAL Message, until a different command is received.

It depends on the type and operation of the specific IPE, whether the Normal_Mode is resumed in both directions, or only in the direction in which the IS_NORMAL Message flows. It must be assumed that in general only this one direction is affected.

17.5.3 Compliance to IS_TRANS_x Messages

The IPE shall act in response to the receipt of an IS_TRANS_x Message (x in the range 1 to 8) such that:

The IPE shall synchronise to it. The IS_TRANS_x Message shall appear unchanged at the output of the IPE.

The IPE shall be transparent in all x LSBs of all PCM samples received subsequent to the IS_TRANS Message.

The transparency shall persist as long as the Keep_Open_Indication persists, or until a different command is received.

The (8-x) upper bits of the PCM samples are not of interest and may be modified arbitrarily by the IPE.

It depends on the type and operation of the specific IPE, whether the Transparent_Mode is resumed in both directions, or only in the direction in which the IS_TRANS Message flows. It must be assumed that in general only this one direction is affected.

17.5.4 Compliance to IS_TRANS_x_u Messages

The IPE shall act in response to the receipt of an IS_TRANS_x_u Message (x in the range 1 to 7) such that:

The IPE shall synchronise to it. The messages shall appear unchanged at the output of the IPE.

The IPE shall be transparent in all x LSBs of all PCM samples received subsequent to the IS_TRANS Message.

The transparency shall persist as long as the Keep_Open_Indication persists, or until a different command is received.

The (8-x) upper bits of the PCM samples are important and in general shall not be modified by the IPE, but shall be bypassed transparently in exactly the same manner and delay as the x LSBs. It is important that this transparency for the upper bits is provided by IPEs that do not understand the specific IS Protocol (e.g. do not understand the IS_System_Identification or the protocol of the transmitted parameters).

Only IPEs which do *exactly* understand the specific IS Protocol shall take advantage of the opportunities given with the IS_TRANS_x_u Messages. An example is the TCME, which transmits internally only the coded speech parameters and re-generates the upper x bits at its output (termed here as "first solution"). The resulting delay in the upper 8-x bits shall be identical to the delay in the x LSBs.

If this transparency of the upper (8-x) bits or their re-generation can not be established, then the upper bits shall contain a constant pattern, giving the least output energy (PCM_Silence). This "second solution" may cause temporary interruptions of the speech signal in some transition cases (e.g. HO in some tandem free mobile-to-mobile calls). Therefore the first solution is the preferred one.

IPEs, which implements the second solution shall switch to the full transparent 64 kBit/s channel as soon as they loose synchronisation with the protocol of the transmitted parameters (e.g. the "TFO Frames" in mobile Systems). The full transparency shall be executed for both directions. The near side shall be fully transparent in less than 60 ms and the other side the one way delay of that IPE later.

It depends on the type and operation of the specific IPE, whether the Transparent_Mode is resumed in both directions, or only in the direction in which the IS_TRANS Message flows. It must be assumed that in general only this one direction is affected.

17.5.5 Compliance to IS_FILL Message

The IS_FILL Message has no specific meaning, but may serve for two purposes.

First of all, it can be used to close the gap in an IS Protocol to keep all IPEs synchronized. Otherwise - in case of an interruption - the n IPEs in the path would swallow the next n IS Messages again.
Second, an IS_FILL Message can be used to resynchronize all IPEs to a new grid position, if necessary.

17.5.6 Compliance to IS_DUP Messages

The IS_DUP Message is sent by an IS Partner to the distant IS Partner to inform about a specific Half_Duplex reception. Most IPEs need not and do not understand this message. They just synchronize to it and let it pass unaltered. Only IS_Responsive IPEs may take advantage. This is system specific and IPE specific.

17.5.7 Compliance to IS_SYL Messages

The IS_SYL Message is sent by an IS Partner to the distant IS Partner to inform about a specific Sync_Lost Situation. Most IPEs need not and do not understand this message. They just synchronize to it and let it pass unaltered. Only IS_Responsive IPEs may take advantage. This is system specific and IPE specific.

18 History

| Document history | | |
|-------------------------|-------------------|---|
| V2.0.0 | February 1999 | |
| V7.0.0 | February 1999 | |
| TDMA_V0.1.0 | 14. March 1999 | Ericsson: Adaptation to TDMA, including VSELP, ACELP and US1 |
| TDMA_V0.1.3w | 20. March 1999 | Ericsson: editorial |
| TFO_V0.1.4 | 21. June 1999 | Ericsson: editorial |
| TFO_V0.2.0 | 13. October 1999 | Ericsson internal: Inclusion of ANSI-95 and rework of ANSI-136 |
| TFO_V0.2.1 | 16. October 1999 | Ericsson: Revised |
| TFO_V0.2.2 | 18. October 1999 | Ericsson: for 3GPP2 meeting 25./26. of October |
| TFO_V0.2.3 | 12. November 1999 | Ericsson: updated after teleconference 10. November |
| TFO_V0.2.5 | 9. December 1999 | Nortel: Update Chapters 5, 9 and 10 after San Diego interim meeting |
| TFO_V0.3.0 | 21. December 1999 | Ericsson: Final Updates after San Diego interim meeting |
| TFO_V1.0.0 | 21. December 1999 | Sprint PCS: Final Updates -- modified cover page and header. |