3GPP2 A.S0004-A Release A

Version 1.0

Date: June 13, 2001

3RD GENERATION PARTNERSHIP PROJECT 2 "3GPP2"

3GPP2 Tandem Free Operation Specification

Version 1.0

June 13, 2001

Contents

Foreword

This standard is derived from TDMA/CDMA TFO standard TIA/EIA-829 which in turn is based on ETS GSM 08.62, Version 7.0.0, released Feb-1999. This is CDMA Tandem Free Operation (TFO) standard version 1.0. It is modified in order to adapt the Tandem Free Operation to the North American Standard TIA/EIA-2000.

The present document introduces the Inband Signaling Protocol between Transcoder/Rate Adapter 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 and may change following the appropriate standard procedures.

1 Introduction

This service description document details the Inband Signaling Protocol between Transcoder/Rate Adapter 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 standard TIA/EIA-2000.

Annex A is mandatory and defines the capability and requirements used in the development of this TFO standard.

Annex B is mandatory and describes the general Inband Signaling (IS) Principle, which is applicable to TFO specifications for all systems (e.g., TIA/EIA-136, TIA/EIA-2000, GSM) and only updated to include the System Identifications for TIA/EIA-136 and TIA/EIA-2000.

Annex C is informative and gives the rules for In Path Equipment (IPE). It is identical for all systems (e.g., TIA/EIA-136, TIA/EIA-2000, GSM).

2 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.

TIA/EIA/IS-2000-A, cdma2000 Standards for Spread Spectrum Systems Release A, March 2000.

TIA/EIA-96C “Speech Service Option Standard for Wideband Spread Spectrum Systems”, 1998

TIA/EIA/IS-127-1 “Enhanced Variable Rate Codec, Speech Service Option 3 for Wideband Spread Spectrum Digital Systems”, 1998

TIA/EIA/IS-733-1 “High Rate Speech Service Option 17 for Wideband Spread Spectrum Communication Systems”, 1999

GSM 08.62: "Digital cellular telecommunication system (Phase 2+); Inband Tandem Free Operation (TFO) of Speech Codecs; Version 7.0.0".

ITU-T Recommendation I.130: "Method for the characterization of telecommunication services supported by an ISDN and network capabilities of an ISDN".

ANSI/TIA/EIA-829, Interoperability Specification for Tandem Free Operation, June, 2000. Refer also to 3GPP2 A.S0004.

3 Definitions and Abbreviations

3.1 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. TRAUs).
TTL is used to denote the link between the TRAU and the TRX, independent of the system.
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 kbps 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 kbps PCM line.

3.2 Abbreviations

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

BFI	Bad Frame Indicator
BS	Base Station
BSC	Base Station Controller
BTS	Base Transceiver System
ECI	Error Concealment Indicator
EVRC	Enhanced Variable Rate Codec (TIA/EIA/IS-127-1)
HO	Hard Handoff
IPE	In Path Equipment
LPC	Linear Predictive Codec
MS	Mobile Station
MSC	Mobile Switching Center
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 (125µs).
PCM_Silence	either PCM_Alaw_Silence, or PCM_µLaw_Silence, dependent on application
Q8	Speech Codec Service Option 1 for TIA/EIA-2000 at 8 kbps (TIA/EIA-96-C)
Q13	Speech Codec Service Option 17 for TIA/EIA-2000 at 13.3 kbps (TIA/EIA/IS-733-1)
TCME	TFO Circuit Multiplication Equipment
TFO_ACK	TFO Acknowledgement Message
T_Bits	Time Alignment Bits
TFO	Tandem Free Operation
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
TRAU	Transcoder and Rate Adapter Unit - this unit performs speech encoding and decoding on the network side of the communications system according to the codec standard selected
TRX	Radio transceiver station
TTL	TRAU-TRX-Link

4 General Approach

4.1 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 64kbps traffic links. This second TRAU encodes the speech signal a second time for the transmission on the second air interface, and the associated second mobile station decodes it again. The two Codecs (Encoder-Decoder pair) of the connection are in "Tandem Operation".

Tandem Free Operation overcomes the disadvantage of degraded speech quality caused by the two consecutive encoding/decoding processes required in Tandem Operation.

Tandem Free Operation requires a bi-directional "transparent" digital channel or path 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 signaling** is not trivial. In particular, out-of-band signaling has insufficient speed to fall back to normal operation in case of sudden interruption of the transparency of the links.

This TFO recommendation defines therefore an **inband signaling 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.
- **provides** a fast fall back procedure for sudden TFO interruption, and

This version does not support resolution of codec mismatch.

Although Tandem Free Operation only requires changes to the TRAUs, network IPEs may also need to be modified to be compliant with TFO.

4.2 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 kbps) by inband signaling, i.e. inserting them into the PCM sample bit stream.

TFO frames have a fixed size (and length) of 320 bits (20 ms) and are carried by 16 kbps traffic channels mapped onto the two least significant bits of the PCM samples.

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

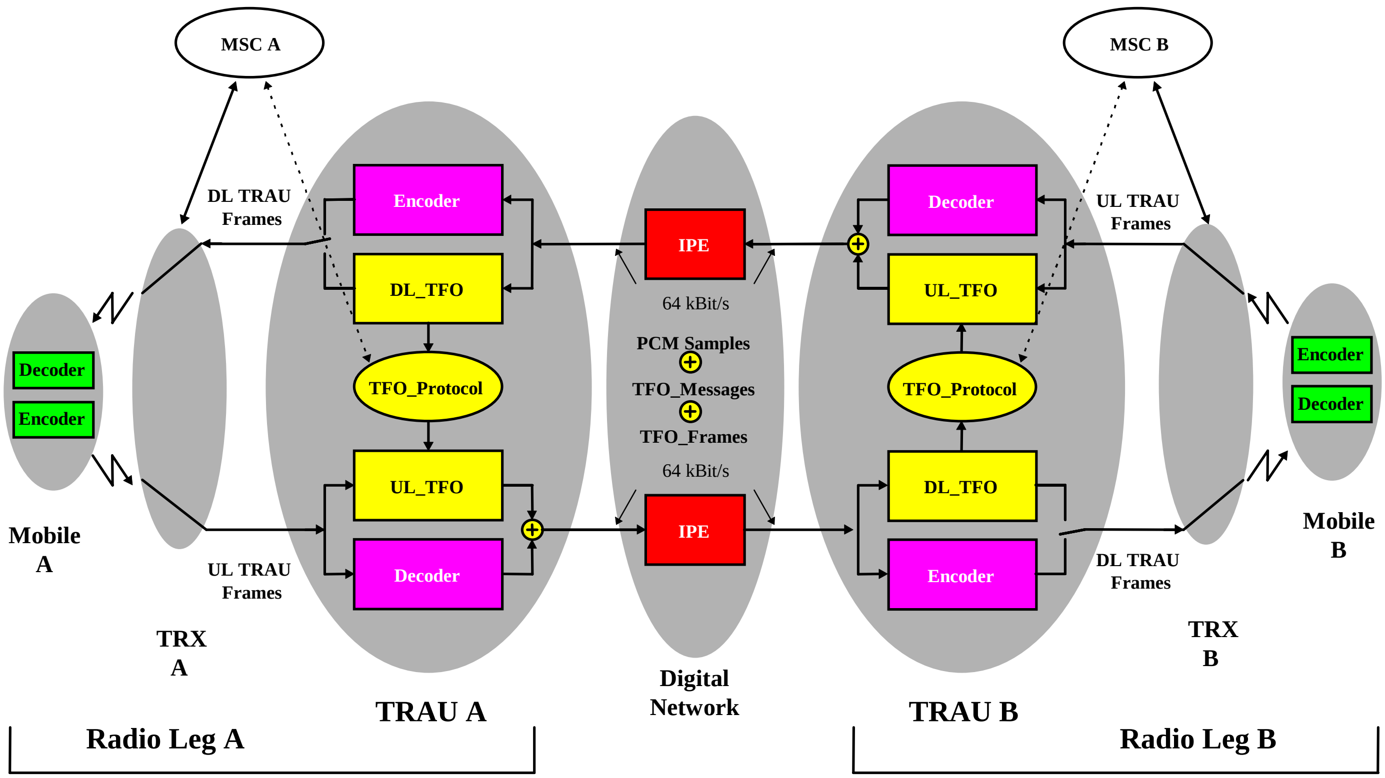

Figure 1: Functional Entities for Handling of Tandem Free Operation in MS-MS calls

TFO provides for transparent transmission of speech data from the Encoder of Mobile A to the Decoder of Mobile B and vice versa.

4.3 CDMA TFO Version Identification

Each release of the CDMA TFO standard specification is uniquely identified with a version number. The version number serves as a universal identification of the capabilities and limitations of the local and remote TFO implementations in a mobile-to-mobile call. It facilitates proper TFO operations between the two parties. Features introduced in various TFO standard updates would be appropriately supported or skipped to ensure proper TFO operations. The text below lists the assumptions and requirements of the CDMA TFO version number.

- CDMA TFO specification version number is uniquely and universally defined and updated for each new release. It is defined independently of the standard document number selected by individual CDMA standard body.
- CDMA TFO version identification is for information only and does not require any message responses from the remote TFO partner.
- TFO operation after any TFO interruptions such as handoffs and conference calls shall be re-established through the normal negotiation process. TFO version information communication between the local and remote parties is ensured.
- This information is optional for TFO implementations of TIA/EIA-829[1] or its equivalence. This information is mandatory for all TFO implementations of new CDMA TFO standard releases replacing TIA/EIA-829 or its equivalence. TFO implementations carrying no TFO version information is assumed to be implementations of TIA/EIA-829 or its equivalence.
- When the local and remote parties have implementations based on different versions of TFO specifications, TFO specification of the earlier version is assumed.

TFO version identification is carried by version extension blocks, which shall be transmitted on successful TFO negotiation and establishment. This is required to avoid additional negotiation delay and to avoid unnecessary information transmission to a TFO incompatible or non-TFO capable far end, e.g. in a mobile to land call. Specifically, the information shall be carried by the version extension blocks of the TFO_IPE messages transmitted immediately before and during transmission of the first TFO frame.

TFO_IPE messages with version extension blocks are also optionally transmitted as embedded messages after TFO establishment on a periodic basis. The exact period could potentially be affected by other higher priority embedded message transmissions during TFO operation and is defined at the vendors' discretion.

"0.0.0 0.0.0" TFO implementation based on TIA/EIA-829/3GPP2 A.S004-0.1.

"0.0.1 0.0.0" TFO implementation based on this version of the standard.

Annex B details the version extension block structure.

5 TFO Frame Structure

5.1 TFO Frame Structure for 16 kbps Sub-Multiplexing

This section defines TFO frame structures for various CDMA speech codecs. These definitions include structures for Q8 (IS-96-C), EVRC (TIA/EIA/IS-127-1), Q13 (TIA/EIA/IS-733-1).

5.1.1 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 speech codecs.

- 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 16^{th} 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 or 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".

5.1.2 Coding of TFO Frames for 16 kbps 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.

5.1.2.1 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 identification bits is the same as coding of System Identification in TFO_REQ_Messages and TFO_ACK_Messages. See Annex B and section 6.

5.1.2.2 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, TIA/EIA/IS-127-1)
		0010 (Q13, TIA/EIA/IS-733-1)
		All others (13) reserved

5.1.2.3 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.

5.1.2.4 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 IS-96-C

Q8 packet type	Code index	D1 D2 … D9
Rate 1	0	0,0000,0000
Rate ½	1	0,0001,1111
Rate ¼	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 TIA/EIA/IS-127-1

EVRC packet type	Code index	D1 D2 … D9
Rate 1	0	0,0000,0000
Rate ½	1	0,0001,1111
Rate ¼	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 TIA/EIA/IS-733-1

Q13 packet type	Code index	D1 … D9
Rate 1*	0	0,000
Blank	1	0,0001,1111
Rate ¼	2	0,1111,1000
Rate 1/8	3	1,0110,0110
Rate ½	4	1,1101,0101
Insufficient frame quality (erasure)	5	1,1010,1011

* Rate 1 uses only a 4-bit code (D1 … D4).

5.1.2.5 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 µs, 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 µs, then T3 and T4 are not transferred. If the timing of the frame is to be delayed by 125 µs (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 TIA/EIA/IS-733-1 can only be advanced by 125 µs, as only two T-Bits (T1, T2) are available.

5.1.2.6 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 5.1.3.

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

$$\mathbf{g(D) = D^3 + D + 1}$$

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

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

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

$$\mathbf{1 + D + D^2}$$

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.

5.1.2.7 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 5.1.3.

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.

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

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.

5.1.2.8 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.

5.1.2.9 Dim-And-Burst with CDMA Variable Rate Codecs

The Dim-And-Burst feature of TIA/EIA-2000 requires the TRAU to lower the coding rate upon request. During TFO, however, the coding rate is not determined by the local TRAU itself, but by the remote mobile. For this situation, the TRAU can send a message to the mobile station by reducing a rate 1 frame to a frame of a lower rate or by using Blank-and-Burst signaling. The design of a suitable algorithm is manufacturer dependent and not in the scope of this document.

5.1.3 TFO Frames for 16 kbps 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)** (gray 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)** (unshaded columns) and **n = (0...39)**.

The underlined synchronization bits, as shown on the frame structure diagrams, can be used for embedded TFO messages.

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

TFO frame structure for CDMA Q8 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 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				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 ½ in the local uplink
C19...C21: Rate reduction parameter, specifying the fraction of packets of Rate 1 and Rate ½ 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).

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 packet with bit errors
D10..D192: See Rate 1 packet

Rate ½ 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 ¼ 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.

Blank:
D1...D9: 9-bit packet type code (1,1000,1100, see Table 4)
D10…D18: 9-bit packet type code (1,1000,1100)
D19…D27: 9-bit packet type code (1,1000,1100)
D28…D192: See section 6.8.1..

Insufficient Frame Quality (Erasure):
D1...D9: 9-bit packet type code (1,1010,1011, see Table 4)
D10...D18: 9-bit packet type code (1,1010,1011)
D19...D27: 9-bit packet type code (1,1010,1011)
D28...D192: Reserved.

T1...T4: Time Alignment Bits

Note:
See IS-96-C for information on speech parameter bit order and the rate reduction parameter values.

5.1.3.2 TFO Frame Structure for CDMA EVRC (TIA/EIA/IS-127-1)

TFO frame structure for CDMA EVRC 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 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 ½ in the local uplink
C19...C21: Rate reduction parameter, specifying the fraction of packets of Rate 1 and Rate ½ 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 TIA/EIA/IS-127-1)

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 TIA/EIA/IS-127-1 and is set to "0", to be consistent with Q13 specification.
D190...D192: CRC over bits s128 to s171

Rate 1 packet with bit errors:
D10..D192: See Rate 1 packet

Rate ½ 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.

Blank:
D1…D9: 9-bit packet type code (1,1000,1100, see Table 5)
D10...D18: 9-bit packet type code (1,1000,1100)
D19...D27: 9-bit packet type code (1,1000,1100)
D28...D192: bInsufficient Frame Quality (Erasure):
D1…D9: 9-bit packet type code (1,1010,1011, see Table 5)
D10...D18: 9-bit packet type code (1,1010,1011)
D19...D27: 9-bit packet type code (1,1010,1011)
D28...192: Reserved.

Rate 1/8 with all bits set to '1':
D1…D9: 9-bit packet type code (0,0111,0011, see Table 5)
D10...D25: all bits set to 1
D26...D192: Reserved.

T1...T4: Time Alignment Bits

Notes:
– See TIA/EIA/IS-127-1 for information on speech parameter bit order and rate reduction parameter values.
–

5.1.3.3 TFO Frame Structure for CDMA Q13 (TIA/EIA/IS-733-1)

TFO frame structure for CDMA Q13 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 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 ½ or rate ¼ 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. For all other packets, these bits are 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 CINDEX 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...D133: Information bits s123 to s0.
D134...D140: CRC over bits s123 to s0 except all 28 CINDEX bits.
D141...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: See section 6.8.1.

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 TIA/EIA/IS-733-1 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 far/remote decoder. The option of it being handled by either the local or remote processor and indicated by the error concealment indicator as in the case of Q8 and EVRC is not available for Q13.
- See TIA/EIA/IS-733-1 for rate reduction parameter values.
- See TIA/EIA/IS-733-1 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 TIA/EIA/IS-733-1).
- Q13 Rate 1, Rate ¼ and Rate 1/8 reserved bits are not transmitted.

6 TFO Message Structure

Several TFO Messages are defined, based on the general IS_Message principle, as defined in Annex B.
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 kbps) or the two LSBs (16 kbps). TFO_TRANS contains the following parameter ():
- the Local_Channel_Type (i.e. 8 kbps or 16 kbps).

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-synchronize 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 section 7.2. The bit-stealing scheme (see Annex B) is identical for regular and embedded TFO Messages. Due to the specific construction of the TFO Messages, they replace some of the synchronization bits of the TFO Frames. TFO Frame synchronization 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.

Restrictions:

1. It is not allowed to send short and isolated TFO messages too close to the end of longer TFO messages. TFO messages with the same length have no such restriction. For more information, see section C 3.2.3.
2. This release of the TFO standard does not support codec negotiation in the event of a codec mismatch in a CDMA mobile-to-mobile call. TFO messages TFO_REQ_L, TFO_REQ_P and TFO_ACK_L are defined only for harmonization with other TFO standards. They are defined for CDMA Rx_TFO operation for intersystem calls only. When used for CDMA TFO operations, the Codec_List contains only the Local_Used_Codec.
3. This release of the TFO standard defines a simplified TFO state machine for CDMA TFO operations. TFO messages TFO_DUP and TFO_SYL are defined for TDMA TFO operations only and are not defined for CDMA TFO operations.

6.1 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 Annex B, 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. Both TFO_REQ_L and TFO_REQ_P are not CDMA-CDMA calls.

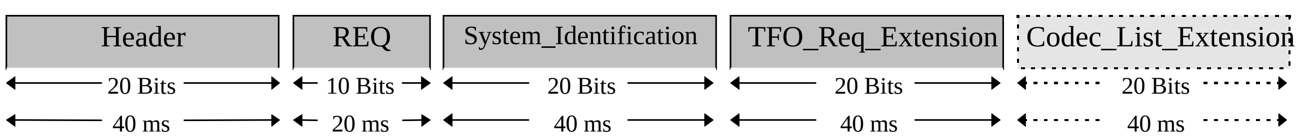

Figure 2: Construction of the TFO_REQ Messages

6.1.1 Definition of the TFO_Req_Extension_Block

The TFO_Req_Extension_Block consists of 20 bits, as defined in table 11.

Table 11: 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 TIA/EIA-136 Vselp (7.95 kbps) / TIA/EIA-96-C Q8 TIA/EIA-136 Acelp (7.40 kbps) / TIA/EIA/IS-127-1 EVRC TIA/EIA-136 US1 (12.2 kbps) / TIA/EIA/IS-733-1 Q13
Bit 16..18:	**CRC**	CRC protecting Req_Ident, Signature and Codec, see 61.2
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

6.1.2 Cyclic Redundancy Check

The Cyclic Redundancy Check (CRC) is operating on the 13 bits consisting of bits 2…10 and 12…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: $d(m) D^{15} + d(m+1)D^{14} + \ldots + d(m+12)D^3 + p(0)D^2 + p(1)D + p(2)$, (where d(m), d(m+12), and p(0) corresponds to bits 2, 15, and 16, respectively and p(0), p(1) and p(2) are the parity bits), when divided by g(D), yields a remainder equal to: $1 + D + D^2$.

6.1.3 Definition of the Codec_List_Extension_Block

The Codec_List Extension Block consists of 20 bits, as defined in Table 12. 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.

Table 12: 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: TIA/EIA-136 Vselp / TIA/EIA-96-C Q8Bit 3: TIA/EIA-136 Acelp / TIA/EIA/IS-127-1 EVRCBit 4: TIA/EIA-136 US1 / TIA/EIA/IS-733-1 Q13The remaining bits are set to "0" and 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 set to "0" and reserved for future Codecs.
Bit 16..18:	**CRC**	A 3-bit CRC protecting the Codec_List fields, see 61.2
Bit 19..20:	**EX** EX == "0.0"	The normal 2 IS_Message Extension bits. No other extension block follows.

6.2 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 Annex B, 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. TFO_ACK_L is not for CDMA-CDMA calls.

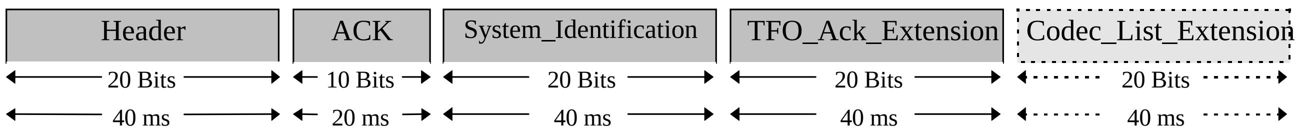

Figure 3: Construction of the TFO_ACK Message

6.2.1 Definition of the TFO_Ack_Extension_Block

The TFO_Ack_Extension_Block consists of 20 bits defined as in table 13.

Table 13: 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, ***reflected*** 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 61.2
Bit 19..20:	**EX**	The normal 2 bits for IS_Message Extension.
	EX == "0.0" EX == "1.1"	ACK: No other extension block follows ACK_L: The Codec_List_Extension Block follows

6.3 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 B. For 16 kbps submultiplexing channels the ”**TFO_TRANS (16k)**” is used and is identical to ”IS_TRANS_2_u”. TFO_TRANS takes 140 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 section 10.

6.4 Definition of the TFO_NORMAL Message

Symbolic Notation: TFO_NORMAL.

The TFO_NORMAL Message is identical to the IS_NORMAL Message defined in Annex B.
It shall be sent at least once whenever an established tandem free operation needs to be terminated in a controlled way.
TFO_NORMAL takes 100 ms for transmission.

6.5 Definition of the TFO_FILL Message

Symbolic Notation: TFO_FILL.

The TFO_FILL Message is identical to the IS_FILL Message, defined in Annex B.
TFO_FILL may be used to pre-synchronize 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.

6.6 Definition of the TFO_DUP Message (Not for CDMA-CDMA calls)

Symbolic Notation: TFO_DUP

The TFO_DUP Message is identical to the IS_DUP Message, defined in Annex B.
TFO_DUP informs the distant TFO Partner, that TFO Frames have been received unexpectedly, e.g. during Establishment. This enables a fast re-establishment of TFO after a ***local*** HO.
TFO_DUP takes 60 ms for transmission.

This message is defined for CDMA Rx_TFO operations for intersystem calls only (e.g. CDMA-TDMA calls) and are undefined for CDMA Tx_TFO operations.

6.7 Definition of the TFO_SYL Message (Not for CDMA-CDMA calls)

Symbolic Notation: TFO_SYL

The TFO_SYL Message is identical to the IS_SYL Message, defined in Annex B.
TFO_SYL informs the distant TFO Partner that tandem free operation has existed, but suddenly no TFO Frames were received anymore. This enables a fast re-establishment of TFO after a ***distant*** HO.
TFO_SYL takes 60 ms for transmission.
This message is defined for CDMA Rx_TFO operations for intersystem calls only (e.g. CDMA-TDMA calls) and are undefined for CDMA Tx_TFO operations.

6.8 Definition of the TFO_DTMF message and TFO_DTMF_ACK message

Symbolic Notation: TFO_DTMF and TFO_DTMF_ACK

The CDMA TFO Version 0.0, does not provide a way to do mobile-to-mobile DTMF signalling without switching into tandem mode. The TIA/EIA-2000 standards for CDMA provide the Reverse (Uplink) DTMF Signalling Message and Forward (Downlink) DTMF Signalling Message. Without a DTMF signalling mechanism in TFO, it is necessary to go into tandem mode (i.e., without TFO) whenever DTMF signalling is needed. Mobile-to-mobile DTMF signalling is needed especially in fixed wireless applications.
This session defines the TFO_DTMF message and the TFO_DTMF_ACK message which use the Reserved bits in Blank TFO frame (defined in section "TFO Frame Structure for CDMA") to carry the DTMF signalling information and DTMF

Acknowledgement information while the call is in TFO mode. The structure of the Blank TFO Frame is modified to support DTMF signalling and its acknowledgement, as well as for future support of generalised mobile-to-mobile messaging.
The acknowledgement procedures facilitate the reliable exchange of DTMF signalling messages between the local (transmitting) TRAU and the remote (receiving) TRAU. This acknowledgement procedures only apply to the Burst DTMF signalling. The local TRAU uses the fields MSG_SEQ (message sequence number) and ACK_REQ (acknowledgement required indicator) to provide a reference for acknowledgements. The DTMF acknowledgement is optional for the local TRAU. A DTMF signalling message requires acknowledgements when the ACK_REQ field is set to '1'.
Upon receiving a Burst DTMF signalling message containing ACK_REQ = 1, the remote TRAU shall respond within T1 seconds (T1 = 100ms) with a "successful" acknowledgement (DTMF_ACK = 0) if the DTMF signalling message is received successfully. Otherwise, it shall respond with an "unsuccessful" acknowledgement (DTMF_ACK = 1) if the DTMF signalling message transmission fail. The ACK_SEQ number in the DTMF Acknowledgement Message is set to the MSG_SEQ number in the received DTMF Signalling Message. It is desirable to minimise the effect of transmitting the acknowledgement blank frame from the remote TRAU. One such optional minimisation technique is for the remote TRAU to wait for either an 1/8, ¼ or ½ rate voice frame arriving from the remote mobile station to use as the blank frame for the acknowledgement. It is recommended that the remote TRAU should wait for the 1/8, 1/4 or 1/2 rate frame within T2 seconds (T2 = 80ms, T2 and T1 timers were started at the same time), otherwise it will blank the next frame and send a TFO_DTMF_ACK regardless of the voice traffic rate.
The local TRAU shall not retransmit a DTMF signalling message for which it has received an acknowledgement with successful indication (DTMF_ACK = 0) and expected ACK_SEQ number. If there are more than one DTMF signalling messages needed to be transmitted and if retransmission is supported, the local TRAU should wait for a positive acknowledgement for each message before transmitting the next DTMF signalling message. If the local TRAU has not received an acknowledgement within T3 seconds (T3 = 160ms) after transmitting the DTMF signalling message, the local TRAU should retransmit the DTMF signalling message with the same MSG_SEQ number. If the local TRAU has received an acknowledgement containing unsuccessful indication (DTMF_ACK = 1); the local TRAU should retransmit the DTMF signalling message. The local TRAU shall support maximum N1 retransmissions (recommend N1 = 2) in response to TFO_DTMF signalling failure. As a result of more than N1 consecutive DTMF signalling failures, the local TRAU may or may not exit TFO mode. As a result of the acknowledgement failure for N1 times, the local TRAU may or may not exit the TFO mode.

6.8.1 Message structure of TFO_DTMF and TFO_DTMF_ACK

(See the TFO Frame Structure for CDMA Q8, EVRC and Q13 speech codec bit position)

- System identifier (8 bits, defined in TFO standards) (S1…S8)
- Codec type (4 bits, defined in TFO standards) (C1…C4)
- Embedded TFO message indicator (1 bit, defined in TFO standards) (C5)
- Packet type (27 bits, use the value for Blank frame type, defined in TFO standards) (D1…D9, D10…D18, D19…D27)
- Blank Frame Message Type (New field, 4 bits): (D28 … D31)
  - 0.0.0.0: Blank Frame only
  - 0.0.1.1: TFO DTMF Signaling Message
  - 1.0.1.0: TFO DTMF Acknowledgment Message
  - All other values: Reserved

Message Body for TFO_DTMF Signaling Message:

Note: Message sequence number is defined by the local Tx_TFO.

- MSG_SEQ (3 bits) (D32 … D34)
  - 0.0.0 … 1.1.1: Message sequence number
- ACK_REQ (1 bit) (D35)
  - 0: Acknowledgment is not required
  - 1: Acknowledgement is required

- NUM_DIGITS (6 bits) (D36…D41)
  - 1 … 32: Number of DTMF digits
  - Note: TIA/EIA-2000 standards use 8 bits for NUM_DIGITS in a message. If more than 32 DTMF digits are requested by the mobile in a single message, then up to 8 TFO DTMF Signalling Messages shall be generated sequentially to transmit all of the requested digits.
  - CRC (3 bits) (D42…D44)
  - CRC over bits D28 to D41
  - Note: The CRC computation shall follow the TFO standards, see section "Cyclic Redundancy Check"
- DTMF_ON_LENGTH (3 bits) (D45…D47)
  - 0.0.0 … 1.0.1: DTMF on length, defined in TIA/EIA-2000 standards.
  - 1.1.1: continuous signaling
  - All other values: Reserved
- DTMF_OFF_LENGTH (3 bits) (D48…D50)
  - 0.0.0 … 0.1.1: DTMF off length, defined in TIA/EIA-2000 standards.
  - 1.1.1: continuous signaling
  - All other values: Reserved
- DIGITi (4 bits per digit, 4 x NUM_DIGITS) (D51…D51+(4 x NUM_DIGITS) - 1)
- 0.0.0.1 … 1.1.0.0: DTMF digit, defined in TIA/EIA-2000standards
  - Use 0.0.0.0 to fill unused bits
- CRC (3 bits, append to the last digit) (D51+(4 x NUM_DIGITS)…D51+(4 x NUM_DIGITS) + 2)
  - CRC over bits D45 to (D51+ (4 x NUM_DIGITS) –1)
- All unused bits are reserved

Message Body for TFO_DTMF_ACKnowledgment Message:

Note: The TFO DTMF Acknowledgment Message is only for the Burst DTMF Signaling.

- DTMF_ACK (1 bits) (D32)
  - 0: Successful
  - 1: Unsuccessful
- ACK_SEQ (3 bits) (D33…D35)
  - 0.0.0 … 1.1.1: Acknowledgment sequence number
- CRC (3 bits) (D36…D38)
  - CRC over bits D28 to D35
  - Note: The CRC computation shall follow the TFO standards, see section "Cyclic Redundancy Check"

All unused bits are reserved

7 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).

7.1 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 section 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 section 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 synchronization bits of the TFO Frames (the ones underlined and light shaded in column m=1 of Tables 8 to 10).

7.2 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 generate the corresponding TFO frames for transmission to the remote TRAU over the PSTN interface. 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.

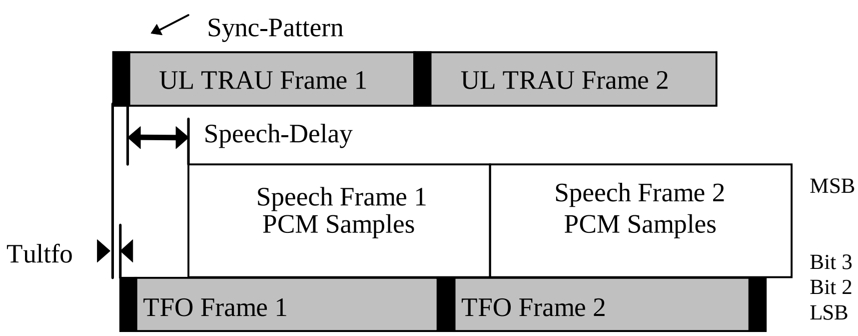

Figure 4: Uplink TFO Frame Time Alignment

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 16^{th} 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:

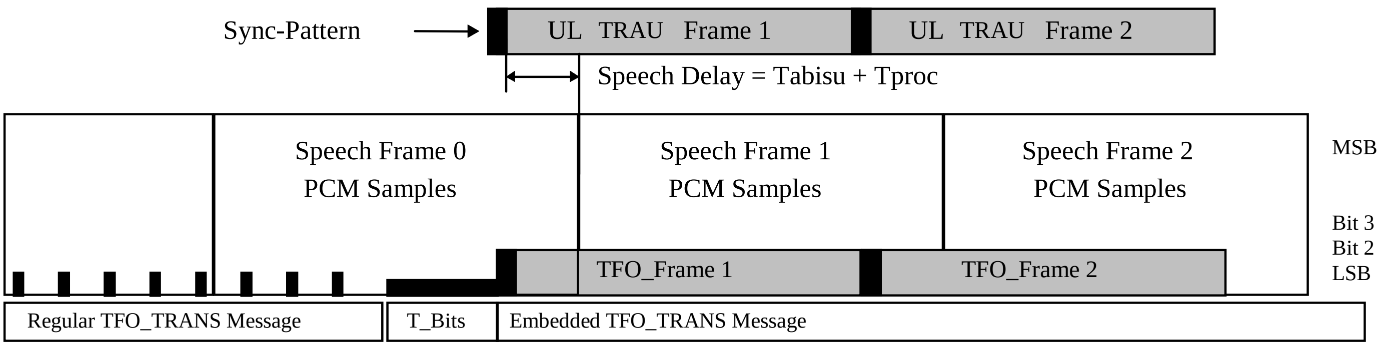

Figure 5: Time Alignment by inserting T_Bits and embedding one TFO_TRANS Message

7.3 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 *(Tbfh)* 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 (Tbfh).* 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). *Tbfh 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.*

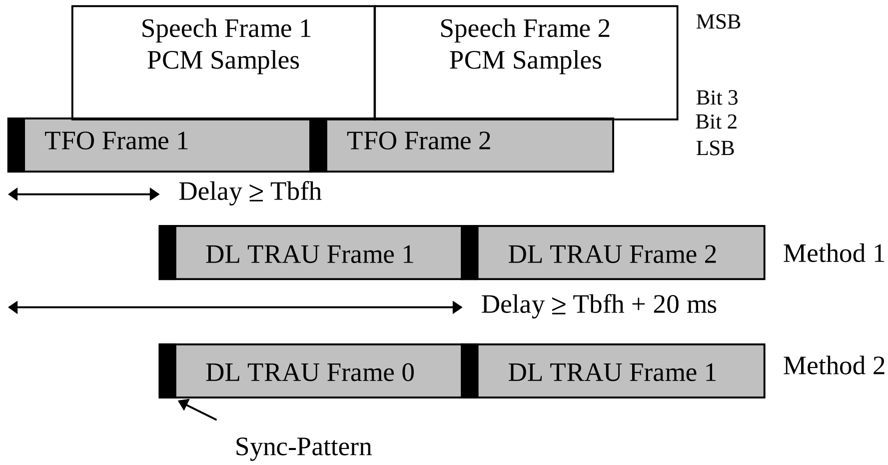

Figure 6: Downlink Time Alignment of TFO Frames

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.

8 Processes for Tandem Free Operation

The following sections describe the actions within the TRAU to establish and maintain Tandem Free Operation in terms of a State Machine, respectively TFO Processes, handling synchronization 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 behavior. 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.
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 quantified 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.

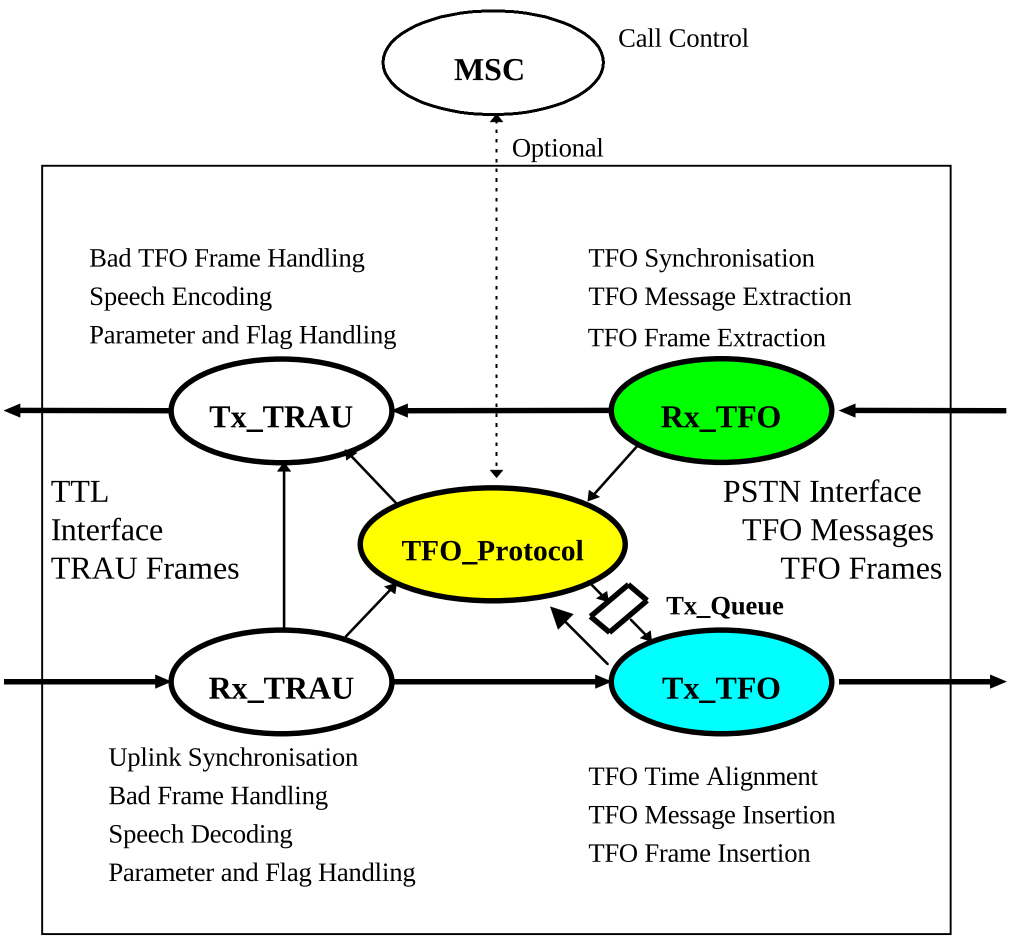

Figure 7: The TFO_TRAU consists of five Processes

8.1 Rx_TRAU Process

The Rx_TRAU Process receives Uplink TRAU Frames from the TTL Interface and synchronizes to them, i.e. checks correct synchronization 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 handed 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.

8.2 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.
Toggling between these two States is commanded by TFO_Protocol with Accept_TFO, respectively Ignore_TFO.

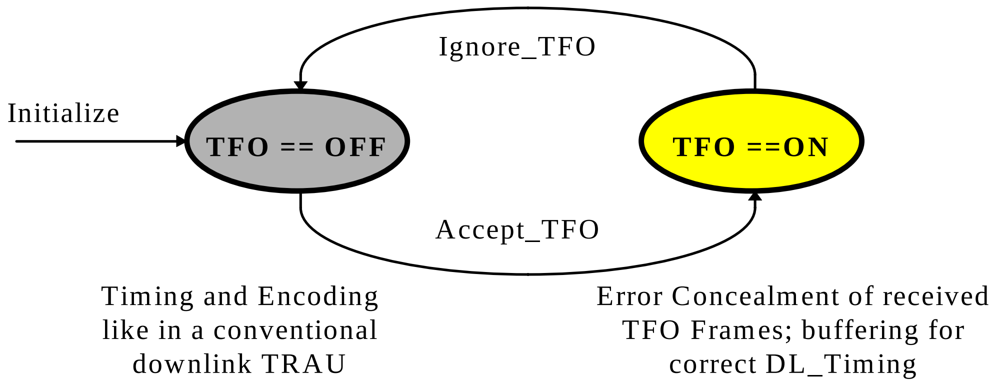

Figure 8: States of the Tx_TRAU Process

During TFO == OFF, Tx_TRAU performs all actions of a conventional downlink TRAU. 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 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 section 7.3).

8.2.1 Downlink Speech Transmission if TFO is ON

During TFO == ON, the Rx_TFO Process receives TFO Frames from the PSTN Interface. It synchronizes to them, i.e. checks correct synchronization 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 section 8.4.2), a Bad Frame Handler to derive suitable parameters for Downlink TRAU Frames. This Bad Frame Handler on the parameter level is manufacturer dependent and is not part of this recommendation.

8.2.2 Synchronization 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.

8.3 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.
Toggling between these two States is commanded by TFO_Protocol with Begin_TFO respectively Discontinue_TFO.

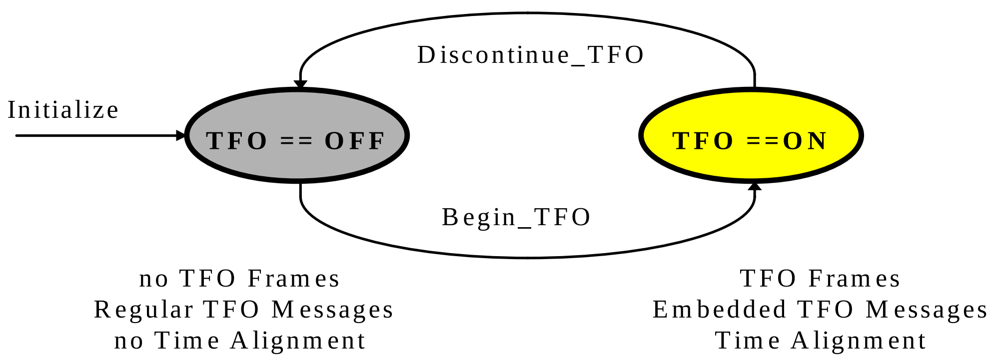

Figure 9: States of the Tx_TFO Process

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 synchronization (see section 10). The TFO_Protocol may delete and process messages from Tx_Queue by using the "Clear Tx_Queue" command at time *Tc*.

Basically, messages or commands that are already being processed by Tx_TFO at *Tc* cannot be stopped, deleted or interrupted. The TFO Protocol is designed to work properly with this default handling, but not, perhaps, at the fastest processing speed. To increase processing speed, Tx_TFO can change the content of a message during transmission as long as the header is still being transmitted (see Figure 10).

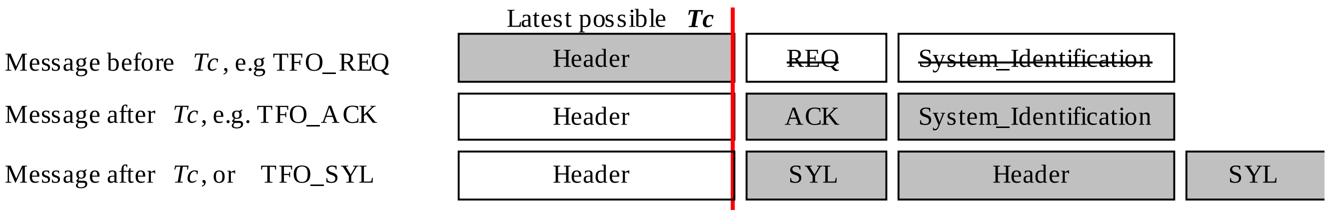

Figure 10: Example of Modification of Message Content during the Header Transmission

8.4 Rx_TFO Process

The Rx_TFO Process receives TFO Messages and TFO Frames from the PSTN Interface and synchronizes to them, i.e. checks correct sync and contents. It passes 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.

8.4.1 Search for and Monitoring of TFO Synchronization

The monitoring of TFO Frame or TFO Message synchronization 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, occasionally disturb that regular phase relation 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 synchronization 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 section 8.4.2) TFO Messages shall be reported to TFO_Protocol with a respective message. If a TFO Message has been received before and synchronization 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. synchronization is regarded as not lost, but the TFO Message is ignored.
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.
If the PCM channel is available for the first time, then a PCM_Channel_Available Message shall be sent to TFO_Protocol. Additionally, 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 TFO Frames are present), 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 synchronization.

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.

8.4.2 Errors in TFO Messages and TFO Frames

These definitions 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.

Messages 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 as long as 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 synchronization 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 synchronization 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.

Messages 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 corresponding bad frame indicator or frame loss indicator is set to 1.

8.5 bTFO_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.

8.5.1 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.

8.5.2 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.

8.5.3 Messages to and from Tx_TFO

The symbol () indicates that these Messages contain parameters, see section 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. (see section 11.1)
Tx := TFO_ACK_L ();	response only to TFO_REQ_L. (see section 11.1)
(Tx := TFO_REQ_P ();	undefined for TRAU, defined only for TCME. (see section 11.1)
Tx := TFO_TRANS ();	command IPEs to go transparent.
Tx := TFO_NORMAL;	reset IPEs into their normal operation.
Tx := TFO_FILL;	mainly to pre-synchronize IPEs.
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.

8.5.4 Messages from Rx_TFO

The symbol () indicates that these Messages contain parameters, see section 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 another, 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_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 synchronization, after a correctly received TFO Frame.
The message "Mess_Sync_Lost" is sent when no following TFO Message is found after a valid TFO message.

8.5.5 Messages from a higher level system

By default a TRAU will attempt TFO. A higher level system (e.g., MSC) can direct the TRAU to enter/exit Tandem Mode using A1-Messages.

Transcoder Control Request message, with TFO mode set to:

Tandem Mode (Inhibit_TFO) This message is sent from higher level system 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 level system, and is received prior to availability of the PCM channel, the message shall be queued until the PCM channel is available.

TFO (Allow TFO) This message is sent from a higher level system to enable TFO negotiation in the TRAU.

Transcoder Control Acknowledge message

This message is sent from the BS to the higher system level (e.g., MSC) to acknowledge whether TFO was successfully enabled or disabled in response to the mode setting request.

Interoperability issues need to be addressed when the BS and the MSC do not support these messages.

9 State Machine of the TFO_Protocol Process for CDMA

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

Initialization (• 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 another State) are handled almost instantaneously and in any case significantly shorter than the time required to complete the transmission of any one TFO Message or TFO Frame.

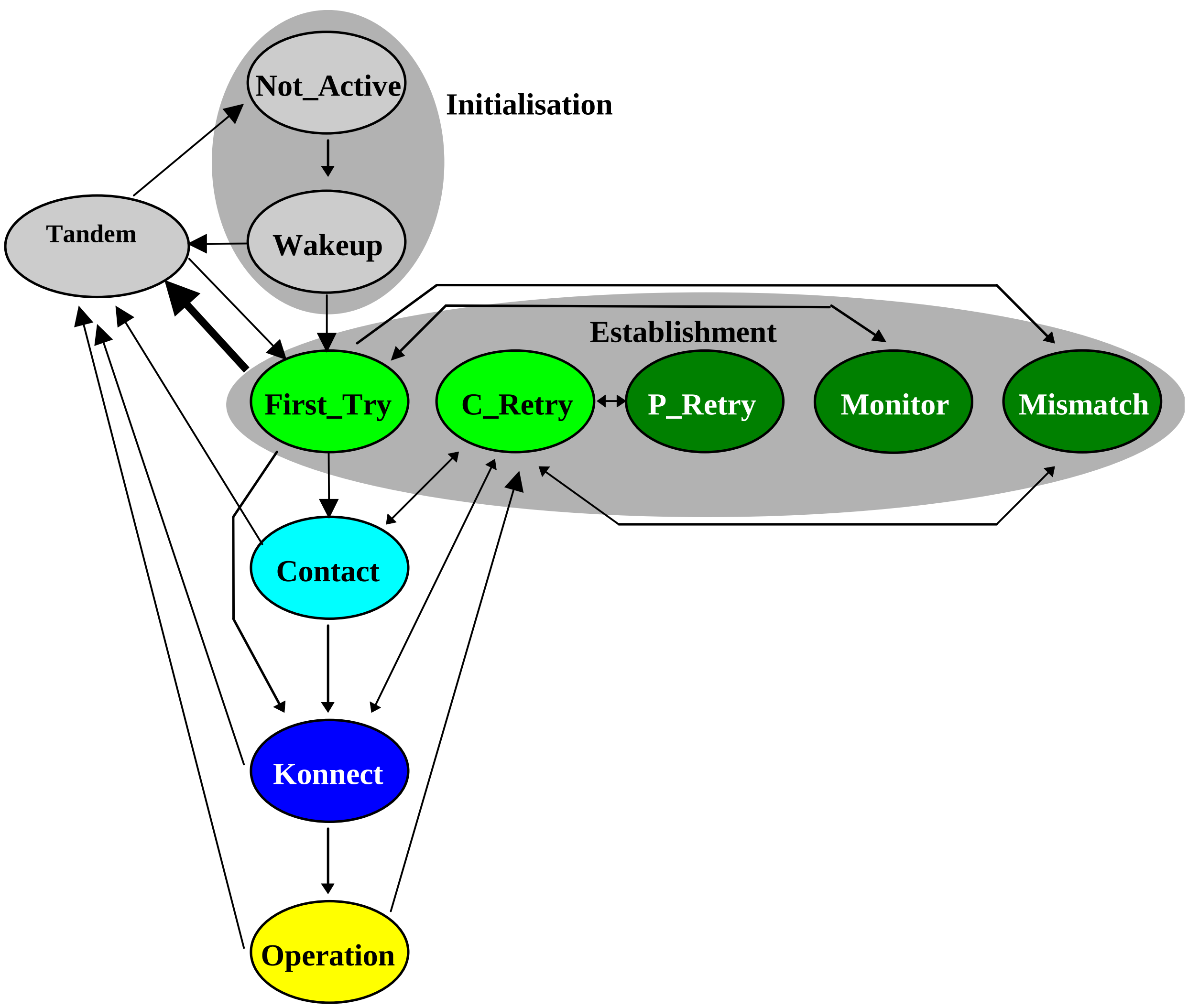

Figure 11: CDMA TFO_Protocol State Machine with most important transitions

9.1 Initialization

9.1.1 Not_Active State

The state Not_Active is system and manufacturer dependent and cannot 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".

9.1.2 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.

TFO operation begins in the Wakeup State and transitions to the First_Try State when speech processing is performed in both the uplink and downlink directions.

9.2 Establishment

The Establishment State encompasses 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 that occurs when the call is still not through connected and the TRAU receives the same outgoing messages. 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.

9.2.1 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. If a TFO Partner does not answer within this maximum time period, 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 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 also remember and compare the previously used Signature. Care has to be taken that the Signature selection contains a true random element in order to minimize the probability of two different TRAUs coincidentally selecting identical signatures.

9.2.2 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.

9.2.3 Periodic_Retry State

Entered from Continuous_Retry, TFO_Protocol periodically sends a single TFO_REQ_L to determine 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, IPEs may not be synchronized.

9.2.4 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 synchronized. It therefore transits into the Continuous_Retry State. In case of Codec Mismatch, it terminates the TFO Protocol by sending TFO_REQ_L back, optionally informs its local MSC, and transits into Mismatch.

NOTE: Since no contiguous transmission of TFO Messages is ongoing, IPEs may not be synchronized.

9.2.5 Mismatch State

From a 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 until either the distant TFO Partner changes to the Local_Used_Codec or the Local_Used_Codec is changed resulting in a match with the Distant_Used_Codec

NOTE: Since no contiguous transmission of TFO Messages is ongoing, IPEs may not be synchronized.

9.3 Contact State

TFO_REQ messages have been received from a distant TFO Partner. The Codecs do match. The link from the distant partner is digitally transparent. Now TFO_ACK needs to be sent to check the digital 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.

9.4 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 Tandem State.

9.5 Operation State

This is the main state of TFO_Protocol. In this state, the TRAU sends and receives TFO frames and the TFO connection is fully operational.

9.6 Tandem

This State is entered when either the distant partner shows an incorrect behavior or a TFO_Inhibit command from a higher level system 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.

10 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 syntax as defined in Table 14.

Table 14: Definition of the Syntax for the State Machine Description

Event: Number:	<Received Message> <running number>	…	<Other Event> <running number>
Condition: &	[<Condition>] [<Condition>]		[<Condition>] [<Condition>]
Comment: **State:**	[<Comment>]		[<Comment>]
<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 17 to table 24, 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 16, with syntax as defined in table 15.

Table 15: 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 earlier. 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 *Tc*). Note that due to the duration time to transmit a TFO_Message completely, the TFO_Protocol Process is often already within another 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 a 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., those 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., those 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 kbps 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, "self-healing" and will handle the unlikely erroneously reported Events, too.

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

Timing: If two Events occur by coincidence at the same time, then they shall be processed in the order given by tables 17 to 24 (left to right) . TFO Messages always arrive some time before the embedding TFO Frame and, therefore, shall be handled 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 16: Defined Actions

Name	Actions	Comments
C	Clear Tx_Queue; T := DIS;	Initialize Tx_Queue and disable the timer
T1	T := 1s;	Set Timeout to 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 synchronize!"
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

Table 17: 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: State:	System activates TRAU	PSTN-Interface gets active; occurs only at beginning	Loopback (LB), or distant handoff (HO)? wrong Signature?	Loopback (LB), or distant handoff (HO)?
NAC: Not_Active .	C;S;IT; WAK; typ. 1st 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 18: 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: . State:	Distant REQ Good Signature .	Distant ACK Good Signature .	Wrong Response Handoff? .	similar to ACK As response to loc ACK_?	one or more 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 for 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 for 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 for 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 for 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 for 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: Konnect .	C;DT;REQ;T1; COR; IPEs transparent!	NoAc; KON; typical: wait	NoAc; KON; .	NoAc; KON; typical: wait	AT;L;T2; 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 19: In Call Modification and Handoff

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 recognized 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; MIS; Direct info.	C;F;REQ; COR; rare case, test	C;F;REQ; COR; rare case, test
CON: Contact .	---------- ---------- .	C;REQ; COR; .	C;F;REQ; COR; rare case, test	C;F;REQ; COR; rare case, test
KON: Konnect .	---------- ---------- .	C;DT;REQ; COR; .	C;DT;REQ; COR; .	C;DT;REQ; COR; .
OPE: Operation .	---------- ---------- .	C;DT;IT;REQ; COR; .	C;DT;IT;REQ; COR; .	C;DT;IT;REQ; COR; .
TAN: Tandem .	NoAc; TAN; .	NoAc; TAN; .	NoAc; TAN; .	NoAc; TAN; .

Table 20: Special Matching TFO Messages

Event:	**TFO_REQ_L**	**TFO_REQ_L**	**TFO_ACK_L**	**TFO_ACK_L**	**TFO_REQ_P**	**TFO_REQ_P**
Number:	9	10	11	12	13	14
Condition: &	Duc==Luc Dsig==Lsig	Duc==Luc Dsig!=Lsig	Duc==Luc Dsig==Lsig	Duc==Luc Dsig!=Lsig	. Dsig==Lsig	. Dsig!=Lsig
Comment: State:	Only sent in MIS/OPE/PER HO? Loop?	Only sent in MIS/OPE/PER Codec_List	Only sent in MIS HO?	. HO?	sent by TCME only embedded	sent by TCME only embedded
NAC: Not_Active .	---------- ---------- .	---------- ---------- .	---------- ---------- .	---------- ---------- .	---------- ---------- .	---------- ---------- .
WAK: Wakeup .	---------- ---------- .	---------- ---------- .	---------- ---------- .	---------- ---------- .	---------- ---------- .	---------- ---------- .
FIT: First_Try .	NoAc; FIT; ignore	NoAc; FIT; ignore	NoAc; FIT; ignore	NoAc; FIT; ignore	---------- ---------- .	---------- ---------- .
COR: Continuous Retry	NoAc; COR; ignore	NoAc; COR; ignore	NoAc; COR; ignore	NoAc; COR; ignore	---------- ---------- .	---------- ---------- .
PER: Periodic Retry	C;F;S;REQ; COR; start again	C;F;REQ; COR; start again	C;F;S;REQ; COR; test	C;F;REQ; COR; test	---------- ---------- .	---------- ---------- .
MON: Monitor .	C;F;S;REQ; FIT; test	C;F;REQ; FIT; test	C;F;S;REQ; FIT; test	C;F;REQ; FIT; test	---------- ---------- .	---------- ---------- .
MIS: Mismatch .	C;F;S;REQ; COR; test	C;F;REQ; COR; test	C;F;S;REQ; COR; test	C;F;REQ; COR; test	S;LA; MIS; acknowledge	LA; MIS; acknowledge
CON: Contact .	C;S;REQ; COR; safe way!	C;REQ; COR; safe way!	C;S;REQ; COR; safe way!	C;REQ; COR; safe way!	---------- ---------- .	---------- ---------- .
KON: Konnect .	C;DT;S;REQ;T1 COR; safe way!	C;DT;REQ;T1; COR; safe way!	C;DT;S;REQ;T1; COR; safe way!	C;DT;REQ;T1; COR; safe way!	S;LA; KON; acknowledge	LA; KON; acknowledge
OPE: Operation .	S;L;T2; OPE; tx Codec_List	C;LA; OPE; Ack List, stop	C; OPE; Ack ok, stop	S;L;T2; OPE; exchange list	S;LA; OPE; acknowledge	LA; OPE; acknowledge
TAN: Tandem .	NoAc; TAN; .	NoAc; TAN; .	NoAc; TAN; .	NoAc; TAN; .	NoAc; TAN; .	NoAc; TAN; .

Table 21: TFO Messages with mismatching Codec Type

Event:	**TFO_REQ**	**TFO_REQ**	**TFO_ACK**	**TFO_REQ_L**	**TFO_REQ_L**	**TFO_ACK_L**
Number:	15	16	17	18	19	20
Condition: &	Duc!=Luc Dsig==Lsig	Duc!=Luc Dsig!=Lsig	Duc!=Luc Dsig==?	Duc!=Luc Dsig==Lsig	Duc!=Luc Dsig!=Lsig	Duc!=Luc Dsig==?
Comment: . State:	Mismatch Wrong Sig, HO? .	Mismatch Good Sig .	Mismatch w/wo HO .	Mismatch Codec_List Wrong Sig, HO?	Mismatch Codec_List .	Mismatch Codec_List .
NAC: Not_Active .	---------- ---------- .	---------- ---------- .	---------- ---------- .	---------- ---------- .	---------- ---------- .	---------- ---------- .
WAK: Wakeup .	---------- ---------- .	---------- ---------- .	---------- ---------- .	---------- ---------- .	---------- ---------- .	---------- ---------- .
FIT: First_Try .	C;S;L;T2; MIS; rare	C;U;L;T2; MIS; typical: Setup	C;U;L;T2; MIS; HO?	C;S;LA; MIS; rare	C;U;LA; MIS; typ: Setup	C;U;LA; MIS; HO?
COR: Continuous Retry	C;S;L;T2; MIS; .	C;U;L;T2; MIS; .	C;U;L;T2; MIS; .	C;S;LA; MIS; .	C;U;LA; MIS; .	C;U;LA; MIS; .
PER: Periodic Retry	C;F;S;L;T2; MIS; .	C;F;L;T2; MIS; .	C;F;L;T2; MIS; .	C;F;S;LA; MIS; .	C;F;LA; MIS; .	C;F;LA; MIS; .
MON: Monitor .	C;F;S;L;T2; MIS; .	C;F;L;T2; MIS; .	C;F;L;T2; MIS; .	C;F;S;LA; MIS; .	C;F;LA; MIS; .	C;F;LA; MIS; .
MIS: Mismatch .	C;S;L;T2; MIS; .	C;L;T2; MIS; .	C;L;T2; MIS; .	C;S;LA; MIS; .	C;LA; MIS; term. Protoc.	C;LA; MIS; term. Protoc.
CON: Contact .	C;S;L;T2; MIS; .	C;L;T2; MIS; .	C;L;T2; MIS; .	C;S;LA; MIS; .	C;LA; MIS; .	C;LA; MIS; .
KON: Konnect .	C;DT;S;L;T2; MIS; .	C;DT;L;T2; MIS; .	C;DT;L;T2; MIS; .	C;DT;S;LA; MIS; .	C;DT;LA; MIS; .	C;DT;LA; MIS; .
OPE: Operation .	---------- ---------- .	---------- ---------- .	---------- ---------- .	NoAc; OPE; Trans. Error?	NoAc; OPE; Trans. Error?	---------- ---------- .
TAN: Tandem .	NoAc; TAN; .	NoAc; TAN; .	NoAc; TAN; .	NoAc; TAN; .	NoAc; TAN; .	NoAc; TAN; .

Table 22: 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; MIS; HO?	NoAc; FIT; HO? Be tolerant	C;U;L;T2; MIS; HO! Mismatch!
COR: Continuous Retry	C;U;L;T2; MIS; .	NoAc; COR; Call Forward?	C;U;L;T2; MIS; .
PER: Periodic Retry	C;F;L;T2; MIS; .	NoAc; PER; Call Forward?	C;F;L;T2; MIS; .
MON: Monitor .	C;F;L;T2; MIS; .	NoAc; MON Call Forward?	C;F;L;T2; MIS; .
MIS: Mismatch .	C;L;T2; MIS; .	NoAc; MIS; Call Forward?	C;L;T2; MIS; .
CON: Contact .	C;L;T2; MIS; .	NoAc; CON; .	C;L;T2; MIS; .
KON: Konnect .	C;DT;L;T2; MIS; .	NoAc; KON; .	C;DT;L;T2; MIS; .
OPE: Operation .	NoAc; OPE; ignore?	NoAc; OPE; Hard HO?	C;DT;L;T2;IT; MIS; Hard HO into TFO
TAN: Tandem .	NoAc; TAN; .	NoAc; TAN; .	NoAc; TAN; .

Table 23: 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: State:	from MSC .	in Call Modif. stop TFO	Internal System Command to Reset TRAU	ignore, is just Filler	ignore
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 24: 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: State:	IPEs may be- come unsynch. .	Time-Out .	No Action .	Stop TFO Frames, if 3 Frames missing .	. .
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; 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; Misbehavior!	---------- ---------- .	---------- ---------- .	C;DT;REQ;T1; COR; after Timeout: N
OPE: Operation .	NoAc; OPE; typ. Final event	 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 25: System Commands

Event:	**Inhibit_TFO**	**Allow_TFO**
Number:	38	39
Condition: &	- .	-
Comment: . State:	System command Received .	System command Received
NAC: Not_Active .	-------- --------	-------- --------
WAK: Wakeup .	TAN;	-------- --------
FIT: First_Try .	C;N; TAN;	NoAc; FIT;
COR: Continuous Retry	C;N; TAN; .	NoAc; COR;
PER: Periodic Retry	C;N; TAN; .	C;F;REQ; COR;
MON: Monitor .	C;N; TAN; .	C;F;REQ; FIT;
MIS: Mismatch .	C;N; TAN; .	C;F;REQ; COR;
CON: Contact .	C;N; TAN; .	NoAc; CON;
KON: Konnect .	C;DT;N; TAN; .	NoAc; KON;
OPE: Operation .	C;DT;IT;N; TAN;	NoAc; OPE;
TAN: Tandem .	NoAc; TAN; .	C; F; REQ; FIT;

11 Codec Mismatch Resolution and Codec Optimization

11.1 CDMA TFO Operations

Codec mismatch resolution and codec optimization are not defined for CDMA TFO operations in this version of the TFO standard. The Local_Code_List in TFO messages TFO_REQ_L, TFO_ACK_L, and TFO_REQ_P contains only the Local_Used_Codec

Annex A (Normative) Tandem Free Operation Capability Description and Requirements

A.1 Introduction

It is expected that the need for Tandem Free Operation will be driven by the increasing market penetration of digital technologies, which will result in an increase in the percentage of calls that are mobile-to-mobile calls. In addition, given that the effects of tandem vocoding are greater for lower bit rate vocoders, the need for this feature becomes greater as the use of low bit rate vocoders increases.

The Tandem Free Operation (TFO) feature, also known as Vocoder Bypass (), improves the end-to-end voice quality observed in mobile-to-mobile voice calls in wireless networks.

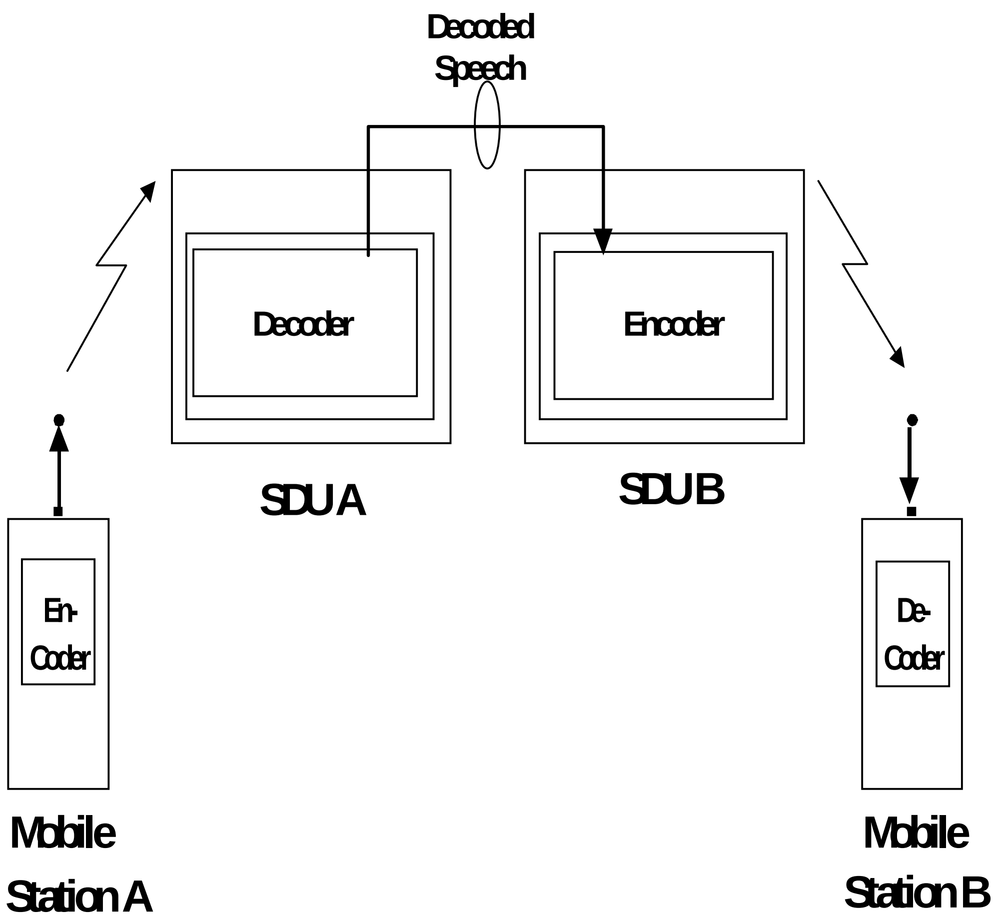

Figure A.1: Logical Diagram of the Speech Path for Digital Calls from Mobile A to Mobile B

As shown in A-1, there are four vocoders concatenated together on a mobile to mobile call connection without TFO. For example, the vocoder in mobile station A encodes the voice and sends it to SDU A. The vocoder in SDU A decodes the voice and sends it across the network to the vocoder in SDU B. The vocoder in SDU B encodes the voice and sends the encoded voice to the mobile station B where it is then decoded again. Each vocoder degrades the voice quality by introducing speech quantization errors that are inherent in any speech encoding and decoding processes. Also each vocoding process introduces additional throughput delay.

The “Tandem Free Operation” feature will enhance current operation by bypassing the vocoding process at the SDUs for mobile to mobile calls. The encoded voice received from MS A is not decoded at the SDU. Instead the encoded voice is transmitted transparently through the network. The encoded speech is routed to the appropriate SDU where it is converted back to the appropriate signaling format for routing to MS B.

A.2 Terminology

A.2.1 Acronyms

BS	Base Station
BTS	Base Transceiver System
CDMA	Code Division Multiple Access
DS0	Digital Signal Level 0
DTMF	Dual Tone Multi-Frequency
GSM	Global System for Mobile Communication
MS	Mobile Station
MSC	Mobile Switching Center
PSTN	Public Switched Telephone Network
SDU	Selection/Distribution Unit
TDMA	Time Division Multiple Access
TFO	Tandem Free Operations
VoIP	Voice over IP

A.2.2 Definitions

2G 2nd generation wireless systems.

3G 3rd generation wireless systems.

Decoder reconstruction of a speech signal from a compressed format according to an appropriate algorithm.

Encoder compression of a speech signal into a new format for bandwidth efficiency according to an appropriate algorithm.

Mobile a standalone unit communicating with the BTS over an air interface in a digital wireless communication system. In this document it is interpreted as a mobile station or a fixed wireless terminal.

SDU a unit for housing the encoder and the decoder required for digital wireless transmission.

A.3 Scope

This annex identifies the mandatory requirements and highly desirable requirements for TFO, which are used as a basis for development of a TFO standard.

A.4 Tandem Free Operation Requirements

A.4.1 Version 0.0 Mandatory Tandem Free Operation Requirements

The TFO Feature shall:

1. Be established and maintained on successful communication between local and remote vocoders. The two bypass capable vocoders must be linked directly with a digital connection with no analog and digital processes such as digital loss pad or conference bridge between them.

2. Be transparent to both the mobile and land users and it should not have any operational impact on a regular mobile to land call.

3. Support cdma2000 to cdma2000 mobile to mobile voice calls.

4. Not prevent voice calls between any 2G or 3G systems regardless if TFO is implemented.

5. Support TFO between two mobile stations that are on the same MSC.

6. Support TFO between two mobile stations that are on different MSCs within the same operator's network. This includes calls which transit other networks but terminate on MSCs within the same operator's network.

7. Support inter-MSC within two different operator’s networks using a single technology, subject to other restrictions. This includes calls, which transit other networks but terminate on MSCs within the different operator's networks.

8. Provide a solution that shall include the ability to inactivate echo cancellers while this feature is enabled, and activate echo cancellers while this feature is disabled. This feature shall not be enabled if echo canceller control is not possible.

9. Neither impact nor be impacted by soft handoff (i.e., TFO and soft handoff function independently).

10. Support TFO both before and after hard handoff when the vocoder type is the same. Support disabling of TFO after hard handoff when the vocoder types are different.

11. Support establishment of bypass mode following interruptions, such as supplementary service (e.g., 3-Way calling, etc.) and hard handoff.

12. Not preclude inband or out-of-band transmission of DTMF. This feature may be temporarily disabled to support transmission of DTMF. This requirement shall not add new DTMF generation requirements.

13. Not be restricted to mobile to mobile calls as long as conditions for TFO exist. For instance, either endpoint of a call may be a fixed wireless terminal.

14. Provide a solution that is not required to be supported across an international gateway, for example, T1 to E1, A-law to mu-law.

15. Provide a solution that shall be capable of functioning in a system that makes use of existing SS7 (ISUP) without impact to signaling and TIA/EIA-41 on the Network-Network interface.

16. Be backwards compatible to a 2G system, i.e., the solution shall function in 2G systems updated with this feature.

17. Avoid impacts to SS7 (ISUP) signaling.

18. Provide a solution that shall result in perceived voice quality equal to or better than that of the same call without the use of TFO.

19. Provide a solution that shall have minimal impact on supplementary services, e.g., Call Forwarding, Call Waiting, 3-Way Calling. However, supplementary services shall take precedence over TFO.

20. Allow Lawfully Authorized Electronic Surveillance without degradation of voice quality perceived by the mobile users.

21. Not cause further degradation to perceived voice quality during bad frames compared to the same call without the use of TFO.

22. Not result in degradation to the perceived voice quality in the presence of link impairments (e.g., jitter, insertion of gain or loss padding) between the two speech processing units compared to the same call without use of TFO. This feature may be disabled under persistent link impairment conditions.

23. Support a system that automatically attempts to use TFO when all necessary conditions are met, and the network operator enables the feature. Specifically, the call must be a mobile to mobile voice call or fixed wireless call using the same vocoder type (algorithm) and no supplementary services prevent the use of TFO.

24. Avoid negative audible impacts on the perceived voice quality during transition to and from bypass mode, e.g., during call setup, during handoff, or at the start and stop of supplementary service activities.

25. Be able to be established within 2 seconds of stable call setup, i.e., a trunk has been established between the two SDUs that contains no echo cancellers or other disruptions

26. Allow TFO mode to be established for the active call during the use of Call Waiting when the criteria for bypass mode are met. That is, when the mobile station is connected to a second call as a result of call waiting or call transfer operations and the bypass mode criteria are met, then bypass mode shall be established.

27. Be capable of allowing the network operator to prevent the establishment of TFO, even if the conditions exist for bypass mode.

28. Support the ability of the network at either end of the call to reject establishment of TFO mode.

29. Provide support for the use of TFO calls between any 2G and 3G systems when compatible TFO implementations exist in those systems.

30. Support the following network configurations (assuming that other necessary conditions for TFO are met):
    - Intra-operator, intra-MSC and inter-MSC: direct trunks between MSCs –OR– call delivery via PSTN.
    – Inter-operator: direct trunks between MSCs –or– call delivery via PSTN.

31. Not be required to work when the span types or companding (A-law / μ-law) are not in agreement.

32. Support transmission across 64 kbps clear channels.

33. Allow for TFO operation following cellular network call forwarding to another mobile. (e.g. call forwarding no answer)

34. Require encoding and decoding processes internal to the speech handlers be reactivated on the completion of a mobile to mobile connection, or on the interruption of a clear digital channel between two TFO capable speech applications such as by a digital loss, handoff, conference bridge, etc.

A.4.2 Version 1.0 Mandatory Tandem Free Operation Requirement

The TFO Feature shall:

1. Provide support for transmission of DTMF tones in tandem or TFO mode.

2. Be able to interwork with previous versions of TFO.

A.4.3 Highly Desired Tandem Free Operation Requirements

The TFO Feature may:

1. Provide support of calls other than cdma2000 to cdma2000 mobile to mobile calls, such as cdma2000 to GSM, TDMA, or Voice over IP (VoIP).

2. Minimize impacts to 2G systems to support TFO. (Example: networks using echo cancellers, gain/loss pad, or voice compression.)

3. Provide a solution that shall allow compatibility with the voice over IP (VoIP) standards.

4. Provide for negotiation towards the use of a common vocoder for the duration of the call in order to establish TFO.

5. Permit, as part of the setup negotiation process, either end of the system to transcode directly from one vocoder to another. When transcoding is employed, the system using it could identify to the other end the type of transcoding in use, thus possibly avoiding the use of two transcoders.

6. Minimize the impact on standards and functions other than those related to speech processing.

7. Support an ATM based interface to permit direct MSC to MSC and BSC to BSC transmission of vocoder frames, thereby reducing the bandwidth requirements between serving systems.

8. Define a means to allow a CDMA system to communicate between the TFO process and the MSC.

Annex B (Normative) Inband Signaling Protocol: Generic Structure

B.1 Scope

Inband Signaling 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). This equipment is termed ”In Path Equipment“ (IPEs).

Annex B defines the generic structure of these IS Messages and rules for the IS_Sender.
Annex C defines the generic rules with respect to these IS Messages for the IPEs.

Annex **B** is mandatory for TFO_TRAU equipment and informative for IPEs.
Annex **C** is informative for TFO_TRAU equipment**; Annex C is mandatory for IPEs which want maintain compatiblity with IS Messages.**

B.2 Generic Structure of Inband Signaling 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.

B.2.1 Frequency and Order of Bit Transmission

IS Messages are transferred within the Least Significant Bit (LSB) of PCM samples on 64 kbps 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 B.1.

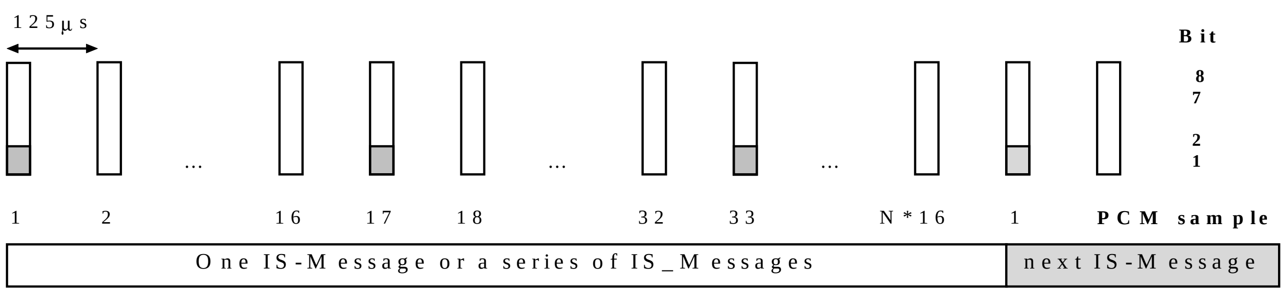

Figure B.1: Inband Signaling Structure

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 therefore 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 B.2 shows an example with two IS_Extension_Blocks.

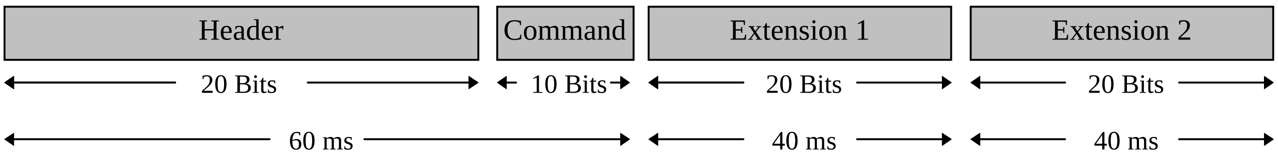

Figure B.2: Example for IS Message with two IS_Extension_Blocks

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.

B.2.2 IS_Header

The IS_Header consists of a 20-Bit long sequence, as defined in Figure B.3.

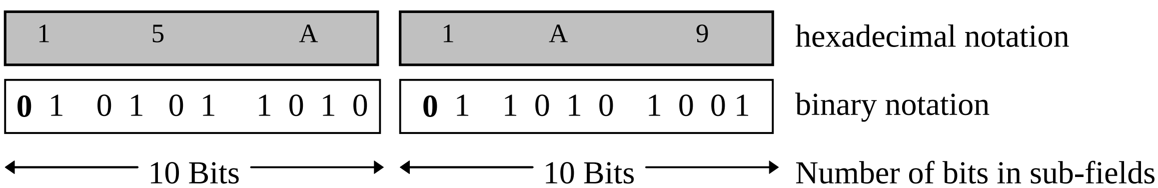

Figure B.3: Structure of the 20 bit IS_Header

B.2.3 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 B.1.

Table B.1: 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 dmin = 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 synchronization purposes, see Figure B.4.

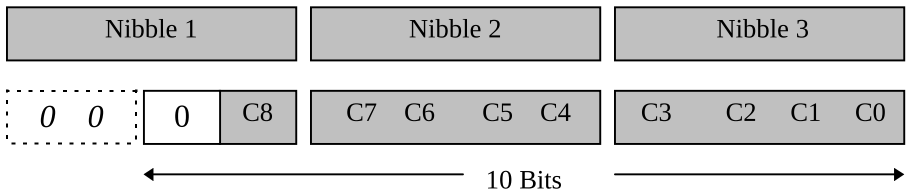

Figure B.4: General Construction of an IS_Command_Block

B.2.4 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 B.5.

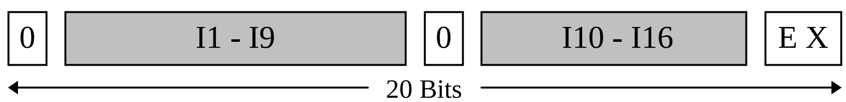

Figure B.5: General Construction of an IS_Extension_Block

The Extension_Field indicates if another 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.

B.3 Detailed Specification of IS Messages

B.3.1 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 B.6). Other IS_Extension_Blocks may follow; see Figure B.6.

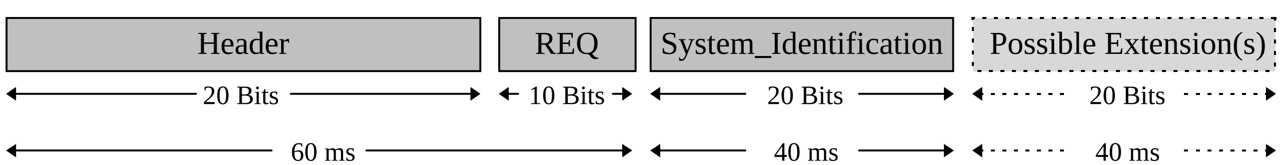

Figure B.6: General Construction of an IS_REQ Message

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).

B.3.2 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 B.6). Other IS_Extension_Blocks may follow; see Figure B.7.

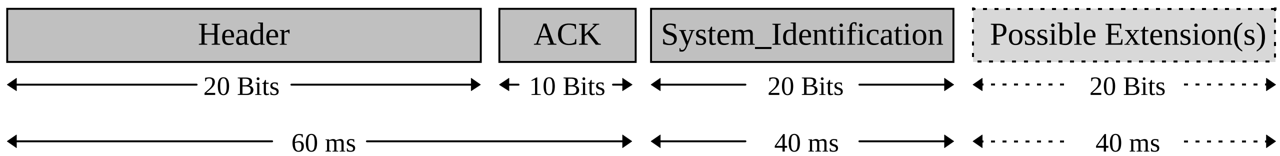

Figure B.7: General Construction of an IS_ACK Message

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

B.3.3 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 B.8.

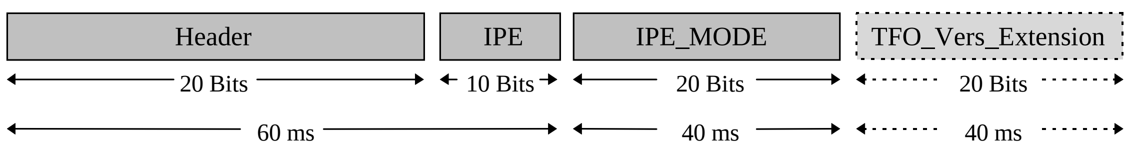

Figure B.8: General Construction of an IS_IPE Message

No specific design constraints with respect to audibility exist, since IS_IPE is typically not sent very often. Table B.2-1 defines 16 out of 32 possible IPE_Commands. The other codes are reserved for future extensions.

Table B.2-1: Defined IPE_Modes

Index	IPE_Mode	Code	MEANING / ACTION
		Hexadecimal Nibble 1 - 5	
0	Normal	0x00000	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 dmin=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 synchronization bits and set to "0", see Figure B.9. The EX field is set to "0.0" when no further IS_Extension_Block is following. The EX field is set to "1.1" when a TFO_Vers_Extension Block follows.

Table B.2-1 defines the coding in hexadecimal notation for the complete IPE_Mode_Extension_Block, with EX := 00.

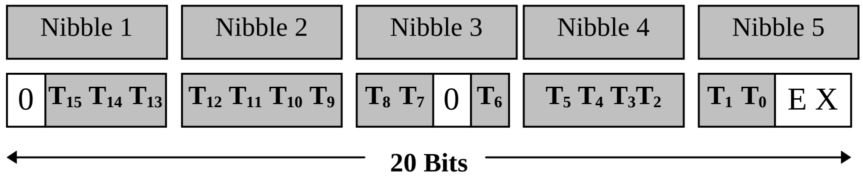

Figure B.9: IPE_Mode_Extension_Block for the IS_IPE Message

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 behavior of IPEs, when receiving such commands, is described in Annex C.
The first IS Message in a series is often "swallowed" by IPEs (see Annex C). 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.

The 20-bit version extension block shall be transmitted as part of the IS_TRANS Messages immediately before and during transmission of the first TFO frame on successful TFO negotiation and establishment. It is optionally transmitted as part of the TFO IPE messages during TFO operation. Section "CDMA TFO Version Identification" describes the TFO version number assumptions and requirements in details. This block carries two version fields. Bits 2 through 4 form the main version field, which is updated for new releases containing major revisions or feature updates. Bits 5 through 7 form the sub-version field, which is updated for new releases containing minor feature updates or corrections. Bits 8 through 10 contain a 3-bit CRC, as defined in section "Cyclic Redundancy Check", covering the two version fields. Table "TFO_Vers_Extension Block" below presents the bit mapping for the TFO version extension block.

Table B.2-2: TFO_Vers_Extension Block

Bit	Description	Comment
Bit 1	"0"	Normal IS-Message Sync Bit, Constant.
Bit 2 .. 4	Main version number	This field is incremented for major revision or feature updates.
Bit 5 .. 7	Sub-version number	This field is incremented for minor feature updates or corrections.
Bit 8 .. 10	CRC	Cover bits 2 through 7. See section "Cyclic Redundancy Check"
Bit 11	"0"	Normal IS-Message Sync Bit, Constant.
Bit 12 .. 18	Reserved	
Bit 19..20	EX EX == "0.0" EX == "1.1"	The normal 2 bits for IS-Message Extension. EX == "0.0": No other extension follows EX == "1.1": Another TFO_Vers_Extension block follows.

B.3.4 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 B.10. 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.

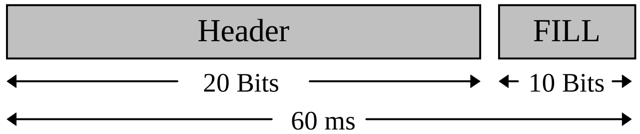

Figure B.10: Construction of the IS_FILL Message

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

B.3.5 IS_DUP Message

The IS_DUP Message may be used between IS Partners to indicate a half-duplex mode. It may be especially important in HO situations. The IS_DUP Message has no IS_Extension_Block, see Figure B.11.

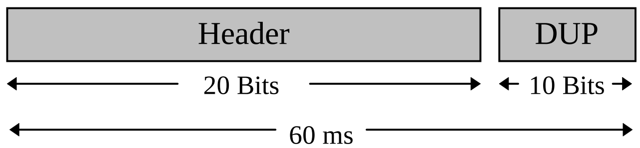

Figure B.11: Construction of the IS_DUP Message

B.3.6 IS_SYL Message

The IS_SYL Message may be used between IS Partners to indicate the loss of synchronization. It may be especially important in HO situations. The IS_SYL Message has no IS_Extension_Block, see Figure B.12.

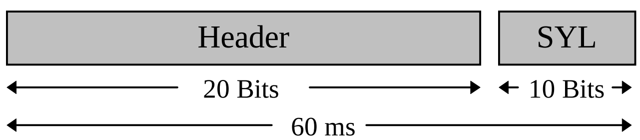

Figure B.12: Construction of the IS_SYL Message

B.4 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 160^{th} 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 B.13.

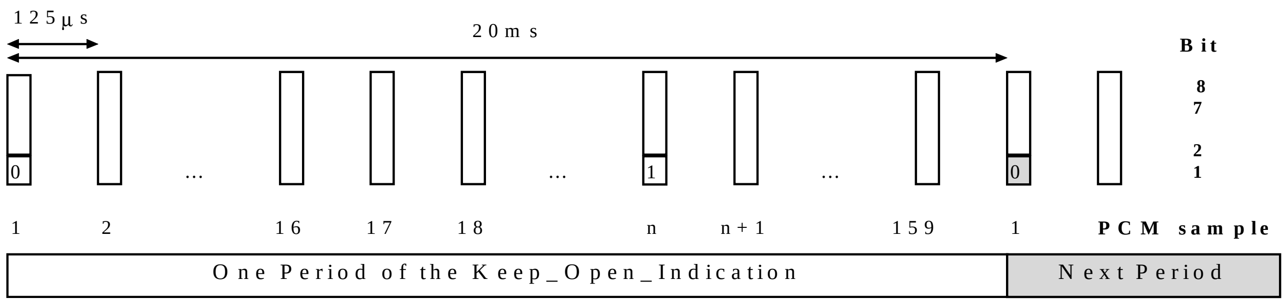

Figure B.13: Keep_Open_Indication

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.

B.5 Rules for Sending of IS Messages

IS Messages replace some bits of the PCM samples and thereforecause 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, and TRAUs (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 section C.3.2.3..

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 B.14.

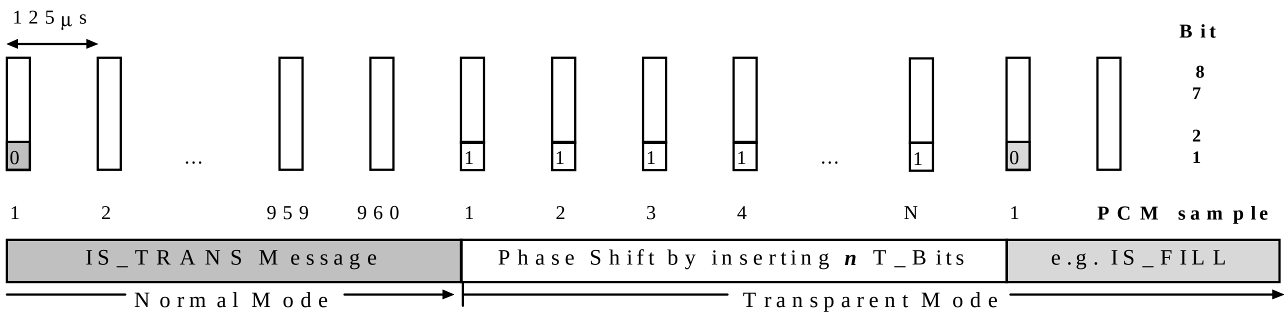

Figure B.14: Phase Shift of the 16_PCM_Sample_Grid by inserting T_Bits

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.

B.6 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 B.3 shows the defined IS_System_Identification codes.

Table B.3: Defined IS_System_Identification Codes

System	Code (in hex)
GSM: 0x53948	either 0x53948, if EX == "0.0" or 0x5394B, if EX == "1.1"
TIA/EIA-136: 0x53414	either 0x53414 if EX == "0.0" or 0x53417 if EX == "1.1"
TIA/EIA-2000: 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, TIA/EIA_2000_Identification and TIA/EIA_136_Identification, see also Figure B.15.

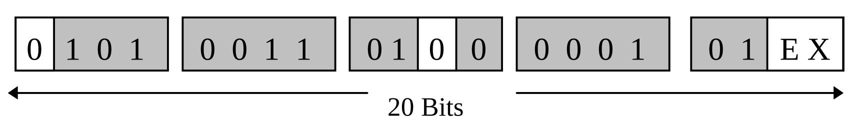

Figure B.15: IS_System_Identification for TIA/EIA-136

All other codes are reserved. Additional IS_System_Identification Codes for other systems shall be defined in a way that the audibility is minimal and the hamming distances to the already defined codes 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 dmin=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 TIA/EIA-136.
Code word 0x0279 = ***0000.0010***.0111.1001 is per definition used for TIA/EIA-2000.
The first, upper eight bits represent the systematic part (underlined), the following eight bits the redundant part of the Code words.

Please note that the ***systematic part*** is also used within the TFO Frames of GSM, TIA/EIA-136 and TIA/EIA-2000 systems for System_Identification (S1...S8), see main part of the document, section 5.

The resulting 16 bits are placed into the IS_System_Extension_Block as shown in Figure B.15 and then the whole 20 bit word is additionally EXORed with the fixed code word 0x53948 to minimize audible effects.

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

C.1 Scope

Inband Signaling 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). This equipment is termed ”In Path Equipment“ (IPEs).

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

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

Annex **B** is mandatory for TFO_TRAU equipment and informative for IPEs.

Annex **C** is informative for TFO_TRAU equipment; Annex C is mandatory for IPEs which want to maintain compatiblity with IS Messages.

C.2 Types of In Path Equipment

The term "In Path Equipment" (IPE) is used for any telecommunication equipment within the (64 kbps) 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 kbps 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;
- the tandem connection of A/μ_Law and μ/A_Law converters, or vice versa;
- links, which insert inband signaling 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.

For some tine, many of these IPEs will not be compliant with the IS Message principle described above. The IS Messages will not always pass these non-compliant IPEs in either one or both directions. 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.

C.3 IS_Compliant IPEs

C.3.1 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 Equipment (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 a system compliant way. Thus they become ***IS_Active*** Partners themselves.

C.3.2 IS Message_Transparency

When commanded into a Transparent Mode, the IPEs are fully transparent at least for the LSBs specified 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.

C.3.2.1 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.

C.3.2.2 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 C.1.

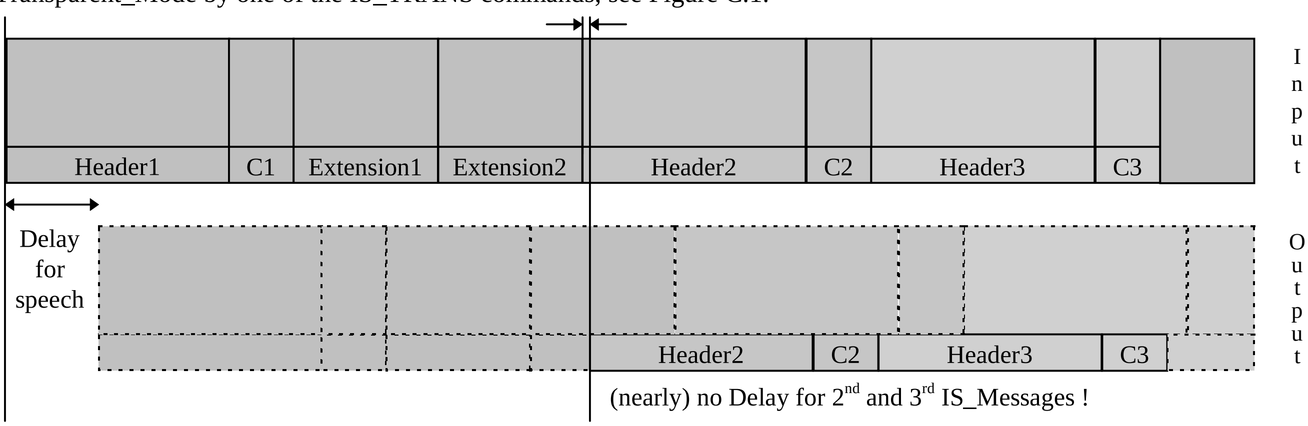

Figure C.1: Transparency and Delay for first and following IS Messages

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 C.1 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.

C.3.2.3 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 C.2, which shows an example where an isolated IS Message is travelling through an IPE.

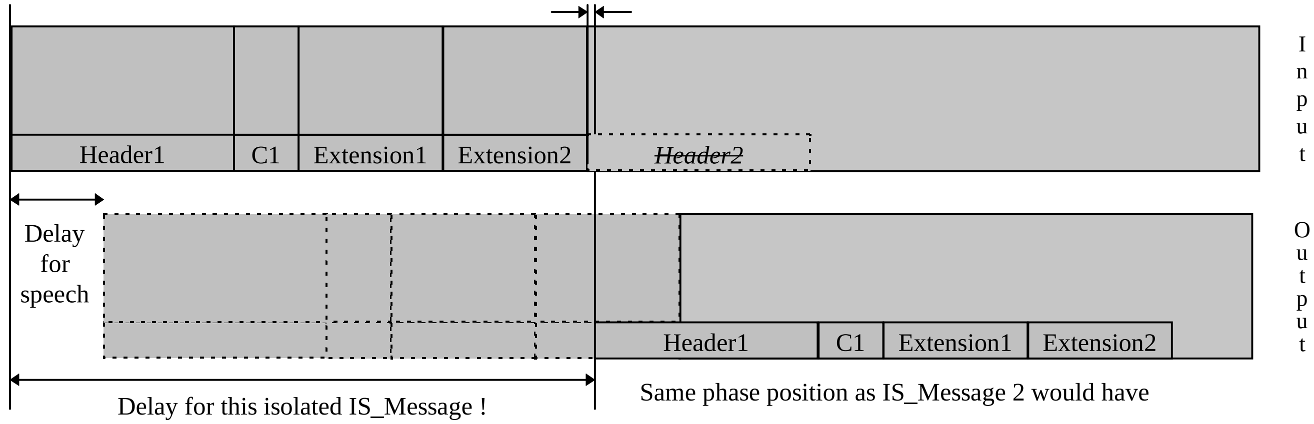

Figure C.2: Transparency and Delay for an isolated IS Message

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 the same length have no such 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.

C.3.2.4 Check if IS Message is following

The checking, whether another 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 C.2.

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.

C.3.3 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 C.3 shows a graphical representation of the State diagram of an IPE.

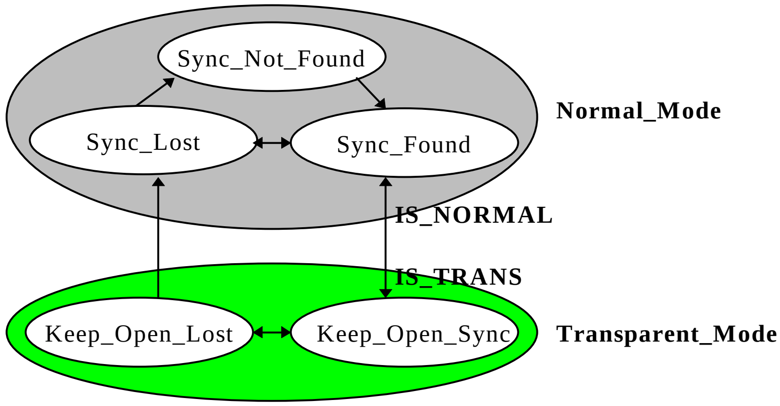

Figure C.3: Principle of a State Diagram of an IPE

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.

C.3.4 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 16^{th} 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.

C.3.5 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.

C.3.6 IPE in Sync_Lost

In Sync_Lost, an IPE shall search for IS Messages on all positions as for initial synchronization. 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 C.4. 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.

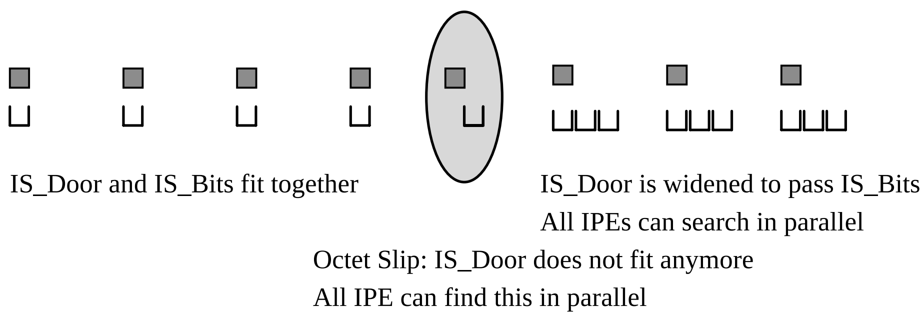

Figure C.4: Handling of octet slip for fast and parallel re-synchronization of all IPEs (optional)

C.3.7 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 another Transparent_Mode (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.

C.3.8 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.

C.4 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.

C.5 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.

C.5.1 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.

C.5.2 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 another 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).

C.6 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 synchronize 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.

C.6.1 Compliance to IS_REQ and IS_ACK Messages

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

C.6.2 Compliance to IS_NORMAL Message

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

C.6.3 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 synchronize 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.

C.6.4 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 synchronize 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 ***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 kbps channel as soon as they loose synchronization 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.

C.6.5 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.

C.6.6 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.

C.6.7 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.

Annex D History

Document history		
V1.0	March 8, 2001	Removal of TDMA text
V1.0	March 8, 2001	Addition of V1.0
V1.0	March 8, 2001	Addition of support for DTMF Tones and TFO version numbering