

RESTRICTED

DATAPATH CLOSED USER GROUP MESSAGING

ISSUE #1.0 - AUGUST 1988.

(c) Copyright 1988 Northern Telecom

**DATAPATH TECHNOLOGY
LICENSING PROGRAM
(Addendum)**



Northern Telecom makes no representations in respect to and does not warrant any of the information in this Specification, but furnishes such in good faith and to the best of its knowledge and ability. Without restricting the generality of the foregoing, Northern Telecom makes no representations or warranties as to fitness for a particular purpose, as to whether or not the use of the information in the Specification may infringe any patent or other rights of any other person. The recipient waives any claims it may have against Northern Telecom in respect of any use which the recipient makes of the information or products derived therefrom.

Northern Telecom reserves the right to alter or modify this Specification or the equipment to which it relates at any time without notice and without liability.

Proprietary

TABLE OF CONTENTS

1.0	Closed User Group Messaging	1
1.1	Bulk Data Transfer - CUG Context	1
1.1.1	CUG Message Types	2
1.2	CUG Message Description - Detailed	4
1.2.1	CUG Download Message	4
1.2.1.1	Parameter Definitions	4
1.2.2	CUG Upload Request Message	5
1.2.3	CUG Upload Indication Message	6
1.3	CUG Exchange	7
1.3.1	Overview of the CUG Exchange	7
1.3.1.1	Synchronization Phase	9
1.3.1.2	CUG Number Exchange	9
1.3.1.3	Idle Phase	10
1.3.1.4	CUG Exchange Error Conditions and Restarts	10
1.3.1.5	Public CUG Number	11
1.3.2	CUG Exchange State Diagrams	14
1.3.3	Transfer of the CUG Number to the CO	19
1.3.4	Response of the CO to the CUG Upload	20
1.3.5	Byte Formats	20

LIST OF ILLUSTRATIONS

Figure 1.	CUG Contexts and Message Types	1
Figure 2.	Downloading/Uploading CUG Data	2
Figure 3.	CUG Download Message	4
Figure 4.	CUG Upload Request Message	5
Figure 5.	CUG Upload Indication Message	6
Figure 6.	CUG Data Byte Format	7
Figure 7.	Normal CUG Exchange Byte Sequence	8
Figure 8.	Failure Conditions During the CUG Exchange .	11
Figure 9.	CUG Exchange Byte Sequence When One TE Does Not Support CUG	13
Figure 10.	Transmitter State Diagram for the CALLED TE	16
Figure 11.	Transmitter State Diagram for the CALLING TE	17
Figure 12.	Receiver State Diagram (CALLED or CALLING TE)	18
Figure 13.	Conditions Causing a TE to Request a Restart	19
Figure 14.	CUG Exchange Byte Formats	20

1.0 CLOSED USER GROUP MESSAGING

This document outlines the messages and actions associated with Datapath Closed User Group (CUG).

The purpose of these messages is to allow the COs of the two TEs in a data connection to exchange the CUG numbers of the TEs before the T-Link Handshake takes place.

The sequence of events for a CUG call is as follows:

1. As part of the call setup, the TE receives its CUG number from the CO.
2. The TE exchanges its CUG number with the far end TE over the 64 Kbps channel.
3. When the CUG number exchange is complete, the TE uploads the far end's CUG number to the CO.
4. The CO either continues with the call setup by instructing the TE to start the T-Link handshake, or takes the call down.

1.1 BULK DATA TRANSFER - CUG CONTEXT

The message types for the CUG context are summarized in the following table:

Figure 1. CUG Contexts and Message Types

CONTEXT		MESSAGE TYPE	
NAME	CODE	NAME	CODE
CUG	1	CUG DOWNLOAD	0
		CUG UPLOAD REQUEST	1
		CUG UPLOAD INDICATION	2

1.1.1 CUG Message Types

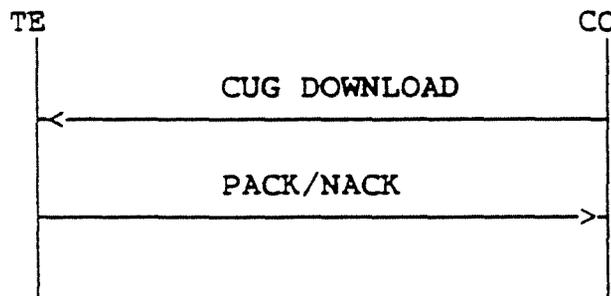
Messages with the CUG context are used to transfer a 16-bit Closed User Group (CUG) number between the TE and CO.

A CUG Download causes the TE to take the CUG number from the CO message and subsequently initiate a CUG exchange with the far end TE. A CUG Upload Indication, containing the CUG number of the far end TE, is sent to the CO when the CUG exchange is complete.

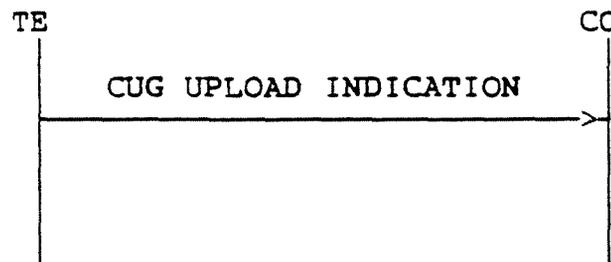
A CUG Upload Request causes the TE to send a CUG Upload Confirmation, which contains the CUG number received from the far end TE.

Figure 2. Downloading/Uploading CUG Data

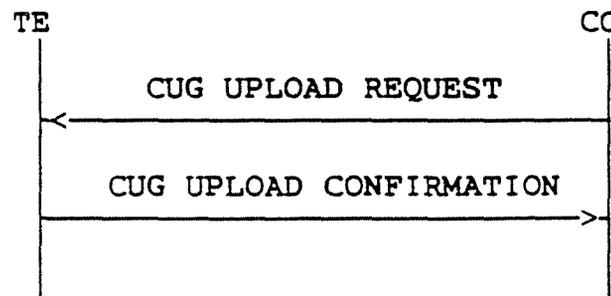
a) DOWNLOAD



b) UPLOAD



c) UPLOAD ON REQUEST

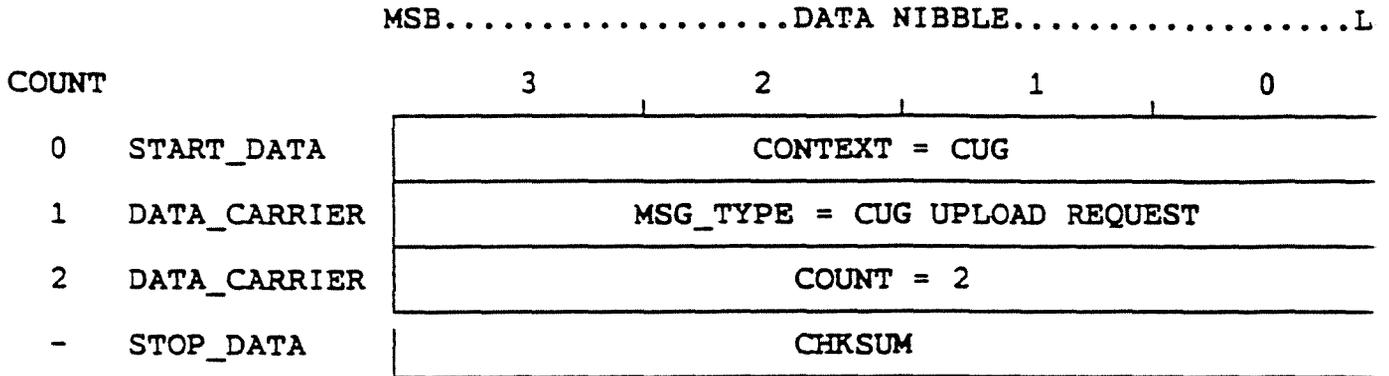


NOTE: "CUG UPLOAD CONFIRMATION" and "CUG UPLOAD INDICATION" are unique message_types in the CUG context although their message bodies are identical.

1.2.2 CUG Upload Request Message

When the CUG upload request is received, the TE uploads the far end TE's CUG number that was received through the CUG exchange. If it has not yet received the CUG number, it must not send the response.

Figure 4. CUG Upload Request Message



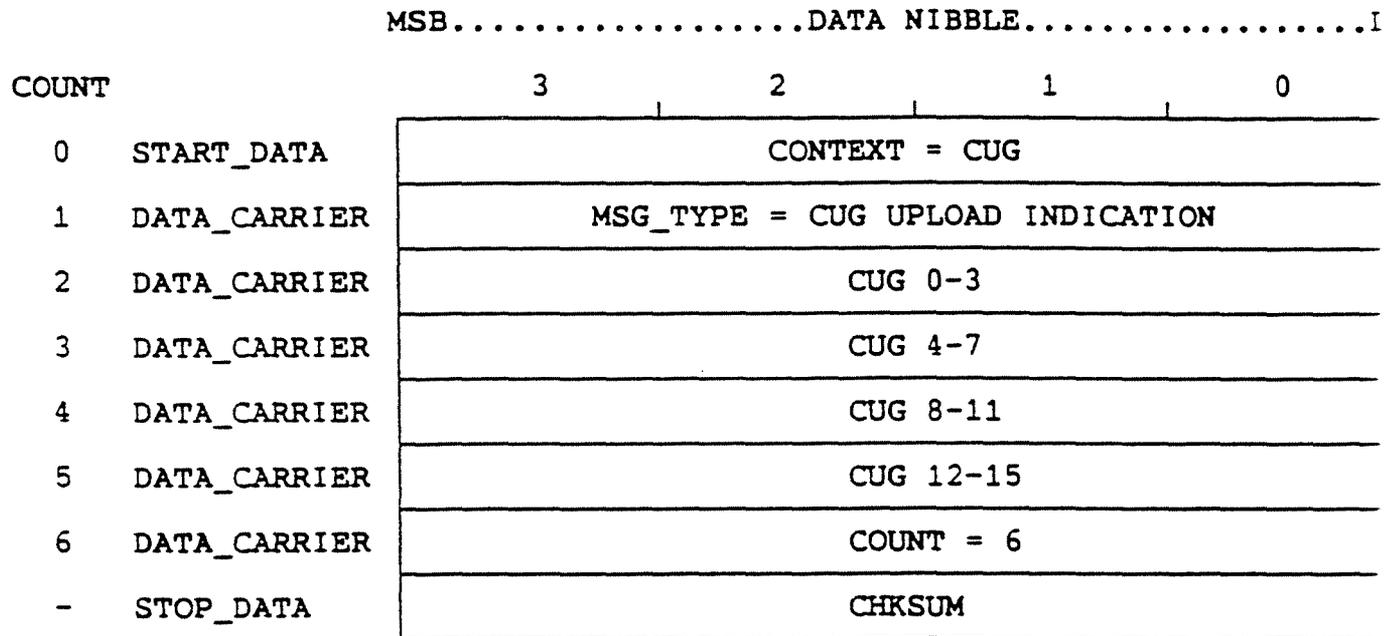
1.2.3 CUG Upload Indication Message

The CUG upload indication message is sent after a successful CUG exchange. The parameters are the nibbles of the far end TE's CUG number received through the CUG exchange.

This message is also sent in response to a CUG upload request message, if the far end CUG number has been received.

For proper interaction with the CO, this reply must be sent within 8 seconds of receiving the CUG download or CUG upload request message.

Figure 5. CUG Upload Indication Message



1.3 CUG EXCHANGE

1.3.1 Overview of the CUG Exchange

When a TE that supports CUG is involved in a call setup, the CO sends the TE its 16-bit CUG number using Bulk Data Transfer on the 8 kbps FDHP signalling channel. The TE breaks the CUG number into a low order and a high order byte, CUG1 and CUG2 respectively. The TE encodes the CUG1 and CUG2 bytes into the T-Link asynchronous data byte format: DL1 and DH1 for the low and high nibbles respectively of CUG1, and DL2 and DH2 for the low and high nibbles respectively of CUG2.

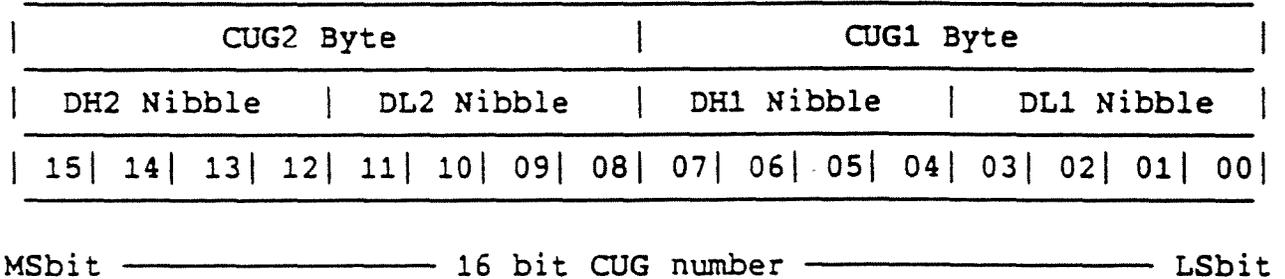


Figure 6. CUG Data Byte Format

Having encoded the CUG numbers, the TE attempts to exchange these data bytes with the far end TE. The CUG exchange is similar to the T-link parameter exchange and has three stages:

- Synchronization Phase
- CUG Number Exchange Phase
- Idle Phase

The end to end flow of messages is shown in Figure 7 on page 8.

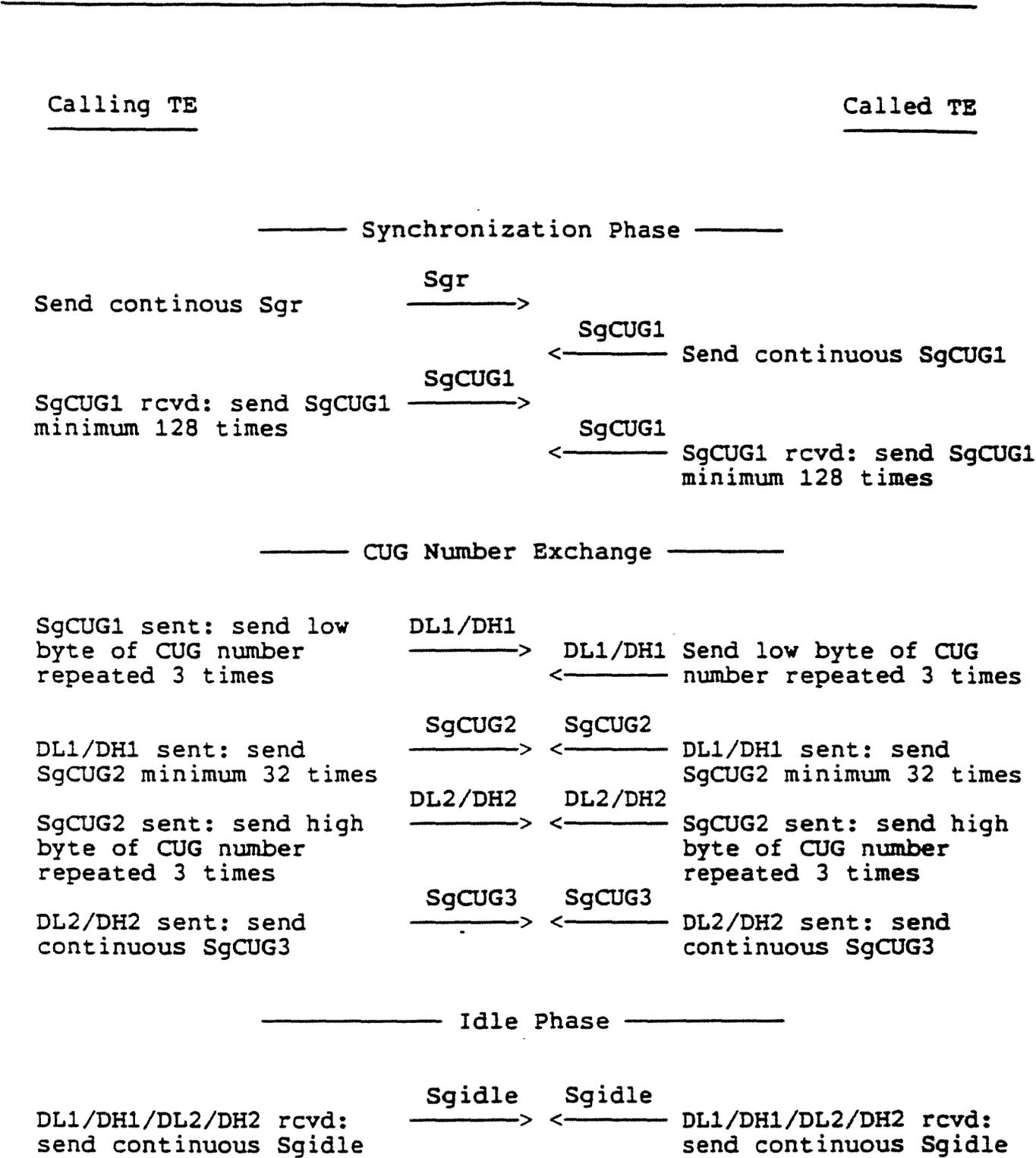


Figure 7. Normal CUG Exchange Byte Sequence

1.3.1.1 Synchronization Phase

The Synchronization phase is used to determine if both TEs in the connection support CUG.

To begin the CUG exchange, the calling TE sends continuous Sgr bytes on the 64 kbps channel and waits to receive SgCUG1 bytes from the called TE. On detection of SgCUG1 bytes from the called TE, the calling TE sends a minimum of 128 SgCUG1 bytes before passing to the CUG Number Exchange phase.

The called TE begins the CUG exchange by sending continuous SgCUG1 bytes. On detection of SgCUG1 bytes from the called TE, the called TE sends a minimum of 128 more SgCUG1 bytes before passing to the CUG Number Exchange phase.

If either the called or the calling TE detects Sgvi bytes, the TE must assume that the other TE does not support CUG. The TE must enter the Idle phase of the CUG exchange and upload the public CUG number of 1 (one).

During the Synchronization phase, it is recommended that the TE use the receipt of at least 16 contiguous signalling bytes of the same value as a means of their detection.

A time limit on the length of the CUG exchange for the called TE is also recommended. If the called TE has not detected SgCUG1 bytes from the calling TE after 4 seconds, it may assume that the calling TE does not support CUG. The called TE would then enter the Idle phase of the CUG exchange and upload the public CUG number of 1 (one).

1.3.1.2 CUG Number Exchange

The CUG Number Exchange phase is identical for both the calling and the called TE.

When the TE has passed the Synchronization phase, it sends the encoded low byte of its CUG number (DL1/DH1) repeated three times. It then sends a minimum of 32 SgCUG2 bytes to the far end, followed by the encoded high byte of its CUG number (DL2/DH2) repeated three times. Finally, continuous SgCUG3 bytes are sent until it has received both bytes of the far end's CUG number. On receipt of the far end's CUG number, the TE exits the CUG number exchange phase.

During the CUG Number Exchange phase, it is recommended that the TE use the receipt of 4 contiguous signalling bytes of the same value as a means of their detection.

1.3.1.3 Idle Phase

After the TE has completed sending and receiving the CUG numbers, the TE sends continuous Sgidle bytes on the 64 Kbps channel and the CUG exchange is complete. The TE then uploads the far end's CUG number to the CO using the CUG Upload Indication bulk data transfer message. The TE continues to send continuous Sgidle bytes on the 64 Kbps channel while waiting for further messaging from the CO to either start the T-Link handshake or take down the call. While in the Idle phase, the TE does not process the bytes received from the far end.

1.3.1.4 CUG Exchange Error Conditions and Restarts

While transmitting the encoded CUG number and associated signalling bytes, the TE must also monitor the received bytes and check for error conditions. If an error condition is detected, the CUG exchange is restarted by having the TE send continuous Sgr bytes and assume the role of the calling TE. If Sgr bytes are detected, the TE restarts the CUG exchange as the called party.

While monitoring for error conditions and restarts, it is recommended that the TE use the same algorithm as is used for the detection of valid signalling bytes (that is, the receipt of 16 contiguous bytes in the Synchronization phase, and the receipt of 4 contiguous bytes in the CUG Number Exchange phase).

While waiting for SgCUG1 bytes in the Synchronization phase, if SgCUG2 or SgCUG3 bytes are detected, the CUG exchange must be restarted. If Sgvi bytes are detected, indicating that the far end does not support CUG, the Idle phase is entered and the public CUG number of 1 is uploaded to the CO.

While waiting for SgCUG2 bytes in the CUG Number Exchange phase, if any signalling byte other than SgCUG1 or SgCUG2 is detected, the CUG exchange must be restarted.

Upon the detection of Sgr bytes during the CUG exchange, the TE must restart the CUG exchange as the called party. The TE begins transmitting SgCUG1 bytes to indicate that it has processed the restart request.

If both TEs restart the CUG exchange at the same time, both will be sending and receiving Sgr bytes. If the TE receives Sgr bytes

while sending Sgr bytes, it must restart the handshake as the called party to avoid lockup.

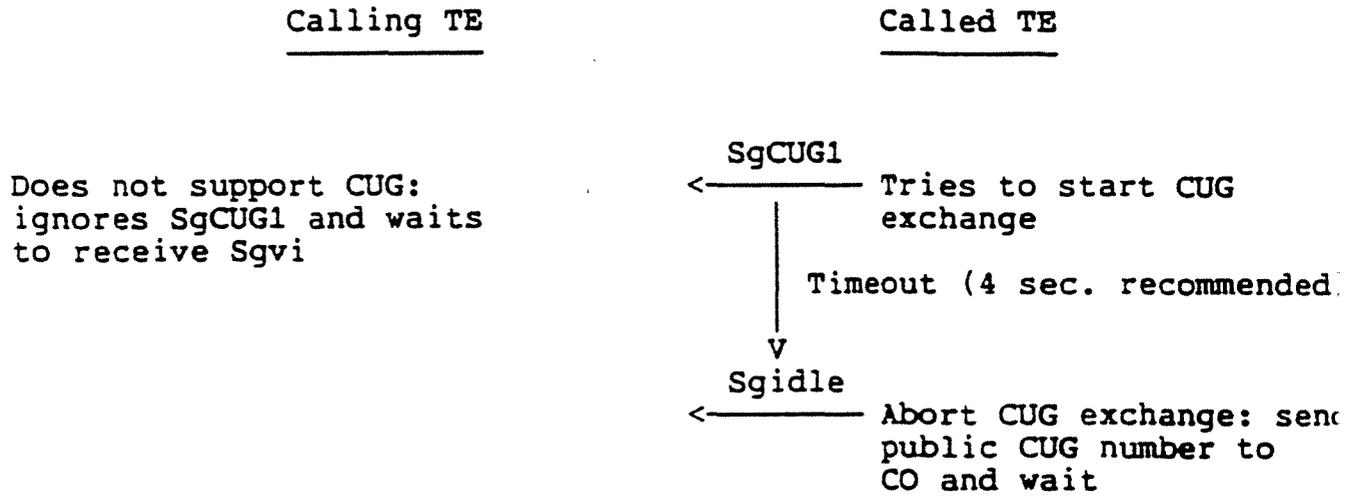
A summary of possible failure conditions during the CUG exchange and the expected responses is shown in Figure 8.

Condition	Effect	TE Response
Called TE does not support CUG	Calling TE detects received Sgvi	Calling TE sends a public CUG number to the CO
Calling TE does not support CUG	Called TE times out waiting for SgCUG1	Called TE sends a public CUG number to the CO
DL1/DH1 or DL2/DH2 lost in channel from called to calling TE	Calling TE receives SgCUG2, SgCUG3 or Sgidle bytes before the expected data	Calling TE restarts the CUG exchange
DL1/DH1 or DL2/DH2 lost in channel from calling to called TE	Called TE receives SgCUG2, SgCUG3 or Sgidle bytes before the expecting data	Called TE restarts the CUG exchange
Both TE's request a restart at same time	Sgr bytes both sent and received by TEs	Both ends send SgCUG1 bytes and act as the called party

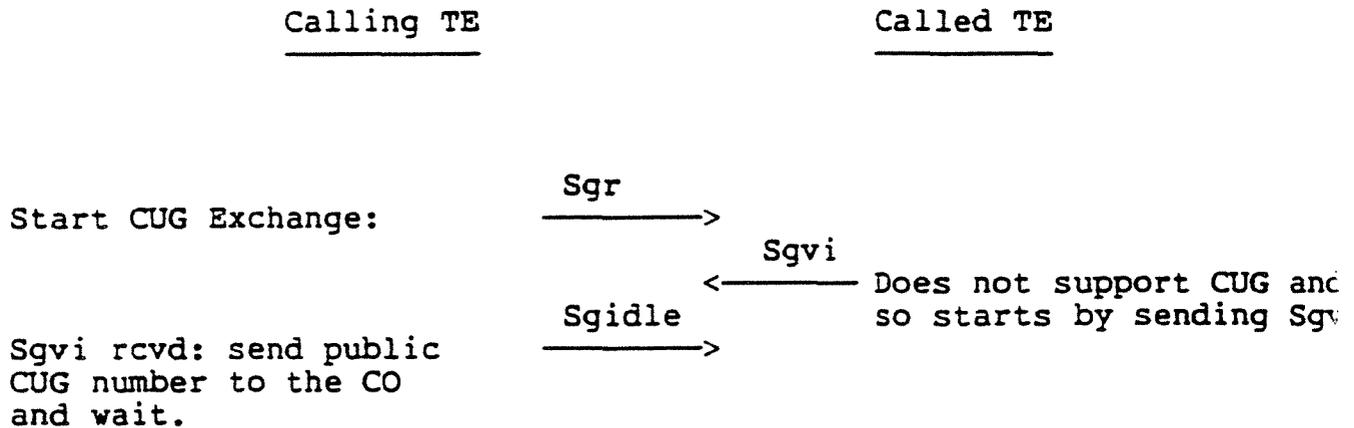
Figure 8. Failure Conditions During the CUG Exchange

1.3.1.5 Public CUG Number

The CUG exchange allows for devices that do not support CUG by defaulting to a CUG number of 1 (Public CUG Number) if the far end does not respond to the CUG exchange Synchronization phase. If the TE detects Sgvi bytes or the CUG exchange times out, the default public CUG number of 1 is uploaded to the CO. This is illustrated in Figure 9 on page 13.



a) Calling TE Does Not Support CUG



b) Called TE Does Not Support CUG

Figure 9. CUG Exchange Byte Sequence When One TE Does Not Support CUG

1.3.2 CUG Exchange State Diagrams

During the CUG exchange, the transmit portion goes through either six or seven states depending on whether it is calling TE (originator) or called TE (terminator):

1. Send Sgr bytes as time fill - calling TE only
2. Send SgCUG1 bytes (minimum 128)
3. Send the low byte of the CUG number (CUG1) three times
DL1/DH1/DL1/DH1/DL1/DH1
4. Send SgCUG2 bytes (minimum 32)
5. Send the high byte of the CUG number (CUG2) three times
DL2/DH2/DL2/DH2/DL2/DH2
6. Send SgCUG3 bytes as time fill until far end CUG received
7. Send Sgidle bytes

Figure 10 on page 16 and Figure 11 on page 17 show these states and the transition conditions for the called and calling TEs respectively.

The receive states are the same for both the calling and called TEs:

1. Wait to receive a minimum of 16 contiguous SgCUG1 bytes
2. Wait to receive the low byte of the CUG number (CUG1)
3. Wait to receive a minimum of 4 contiguous SgCUG2 bytes
4. Wait to receive the high byte of the CUG number (CUG2)
5. Wait for transmitter to finish sending CUG number to far end TE
6. Upload far end CUG number and wait for CO messaging

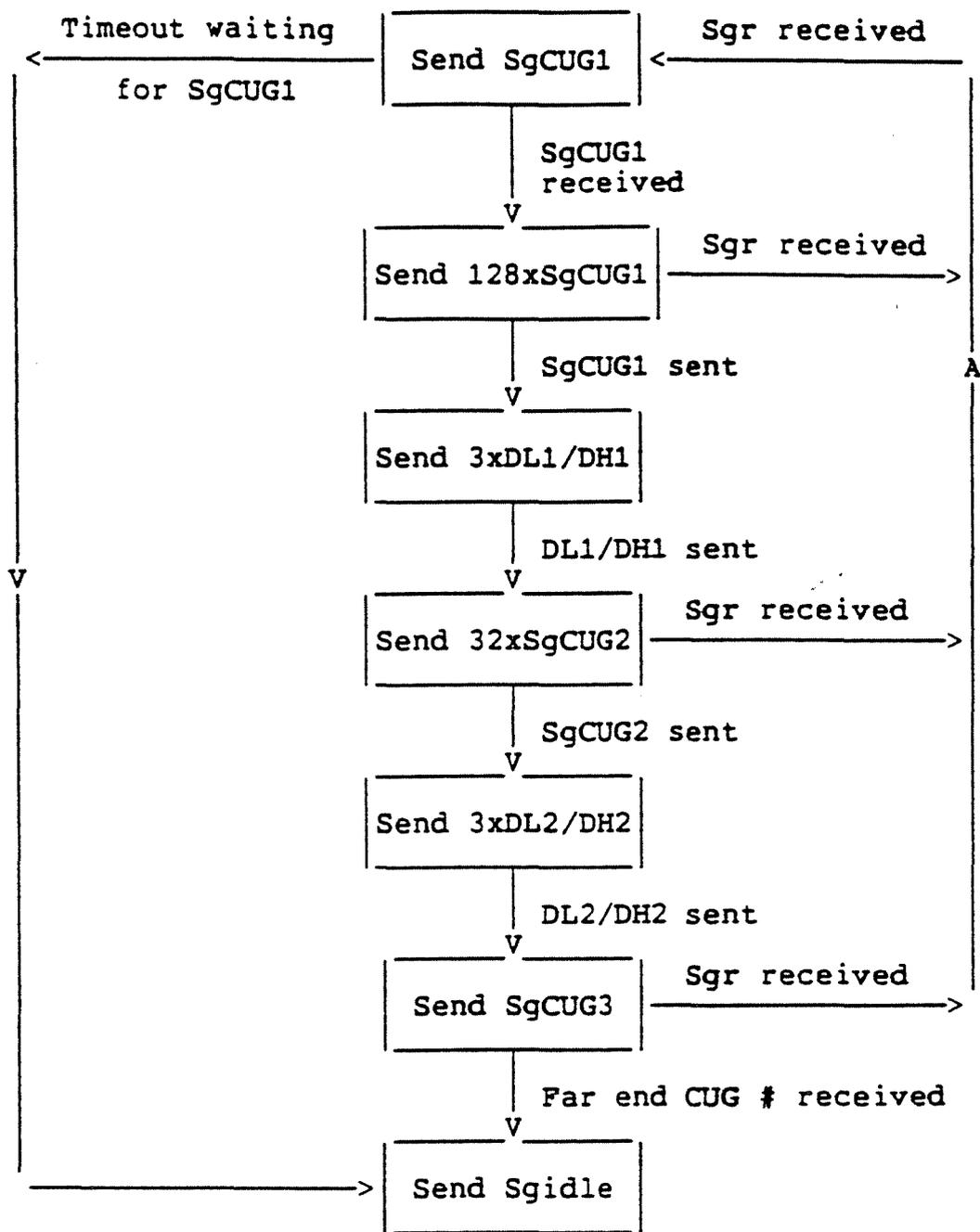
The receiver state diagram is shown in Figure 12 on page 18.

The receiver also detects and acts on failure conditions. If either TE receives Sgvi bytes at the start of the call or the called TE does not receive a reply to its SgCUG1 before the timeout period, the TE must assume that the far end does not support

CUG. In this case, the CUG exchange goes to the Idle phase and the TE uploads the public CUG of 1 to the CO.

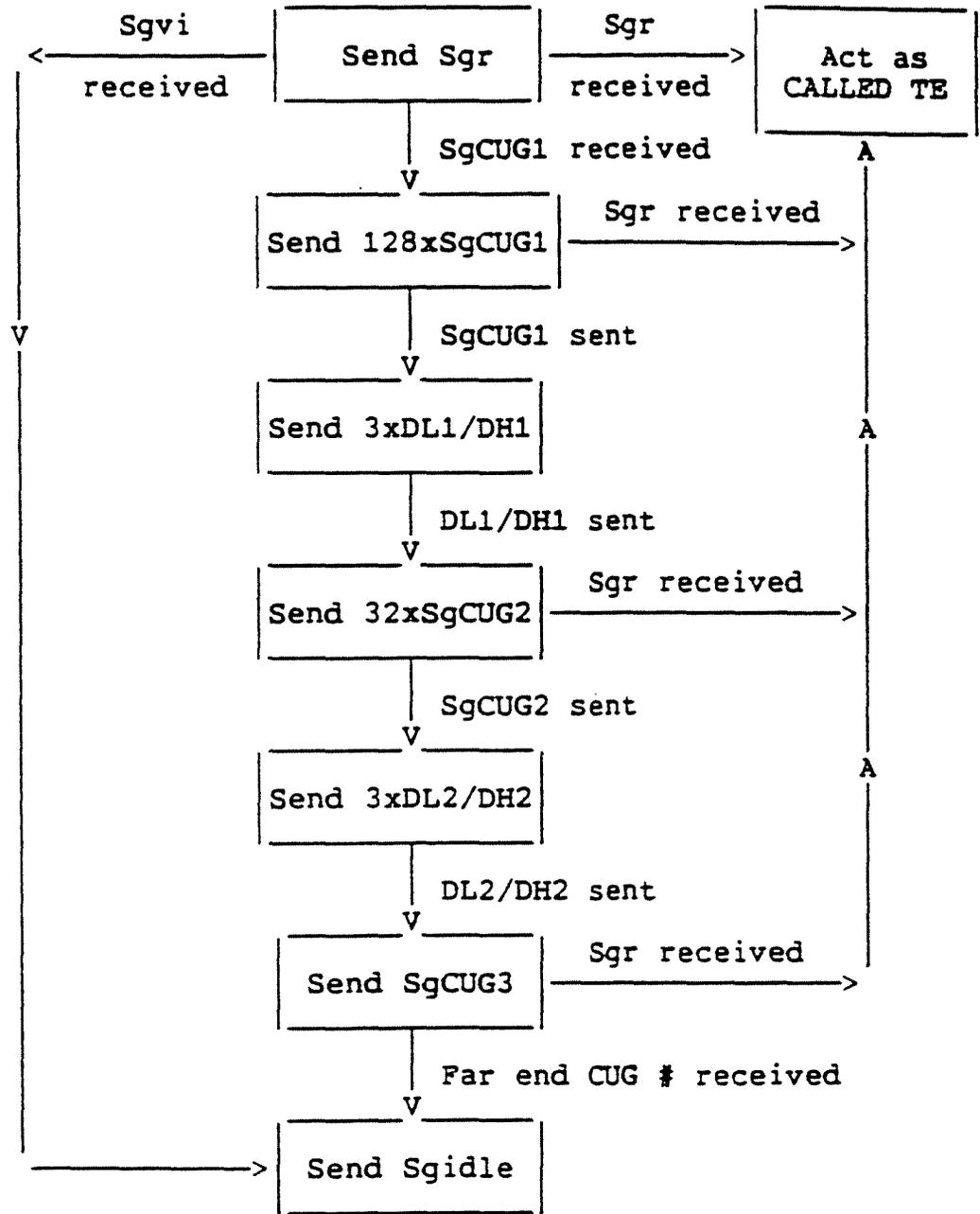
In each state, the receiver checks for restart requests (Sgr bytes) and out of sequence signalling bytes. If either occurs, the TE begins the CUG exchange over. If Sgr is received, the TE acts as the called party in the restart and sends SgCUG1. Out of sequence bytes cause the TE to request a restart itself by sending Sgr bytes to the far end TE and acting as the calling party. If both ends request a restart at the same time, both will receive Sgr bytes and restart as the called party. This is acceptable since two called parties can successfully exchange parameters.

Figure 13 on page 19 lists the out of sequence signalling byte values that cause the TE to request a restart.



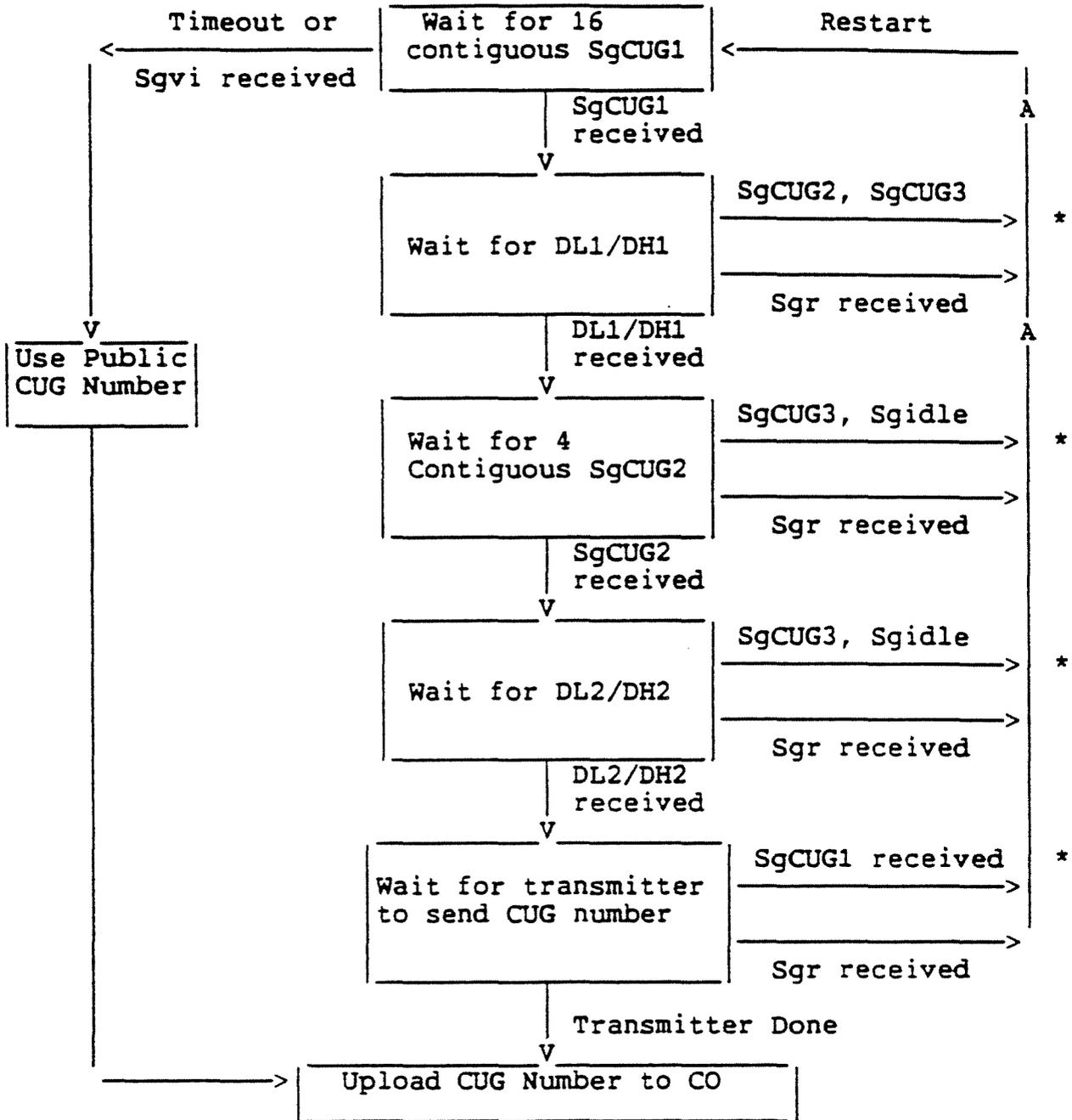
Notes: This state diagram also applies to TE's receiving a restart request

Figure 10. Transmitter State Diagram for the CALLED TE



Notes: This state diagram also applies to TE's sending a restart request.

Figure 11. Transmitter State Diagram for the CALLING TE



Notes: If signalling bytes are received out of sequence (marked by a * in the diagram) the receiver will force a restart with the transmitter acting as the calling party.

If Sgr (restart request) is received the receiver will force a restart with the transmitter acting as the called party.

Figure 12. Receiver State Diagram (CALLED or CALLING TE)

Waiting For:	Other Acceptable Bytes (Ignored by Receiver)	Unacceptable Bytes (Cause a Restart)
SgCUG1	anything else	SgCUG2, SgCUG3, Sgr
CUG1	SgCUG1	anything else
SgCUG2	SgCUG1	SgCUG2, SgCUG3, Sgr
CUG2	SgCUG2, SgCUG1	anything else
Transmitter done	anything else	Sgr, SgCUG1

Notes: At any state in the call (including waiting for SgCUG1) the receipt of Sgr bytes will cause the TE to act as the called party and start the CUG exchange over by sending SgCUG1 bytes

If a TE requests a restart it will send continuous Sgr bytes and act as the calling party when SgCUG1 bytes are received in reply.

Figure 13. Conditions Causing a TE to Request a Restart

1.3.3 Transfer of the CUG Number to the CO

After the TE to TE CUG exchange is complete, the received CUG number (or the default of 1 if the far end TE does not support CUG) is automatically uploaded to the CO.

The TE will also retransmit the received CUG number in response to a CUG upload request from the CO. Upload requests received before the end of the CUG exchange are ignored.

1.3.4 Response of the CO to the CUG Upload

When the CO receives the CUG upload from the TE, it sends a message to the TE to either start the T-Link handshake (70H - Data Alert On) or take down the call (48H - Call processing reset). In either case, the CUG exchange is completely terminated, the far end CUG number is erased, and any subsequent CUG upload request is ignored.

1.3.5 Byte Formats

The CUG exchange requires the use of three new signalling bytes that are not part of T-Link. The formats of these bytes are displayed in Figure 14 on page 20.

The formats of the Sgr, Sgvi, DL and DH bytes follow the patterns defined for T-Link.

