AM-TR-NPL-000005

Technical Interface Specifications Datapath Network Access Interface

To:	All Interested Parties
Priority:	N/A
Effective Date:	
Issue Date:	Issue #1, December 1985
Expires On:	N/A
Training Time:	N/A
Related Documents:	N/A
Canceled Documents:	N/A
Issuing Department:	N/A
Distribution:	N/A
Business Unit:	N/A

Table of Contents

1. Introduction 3
  1.1. General 3
  1.2. Terminology 3
  1.3. Applicable Document 5
2. Interface Overview 5
3. Interface: Physical and Electrical Connection Requirements 5
  3.1. Physical Connection Requirements 6
    3.1.1. Telco Provided Interface 7
  3.2. Power Requirements 7
  3.3. Sealing Current 7
  3.4. Electrical Connection Requirements 7
    3.4.1. AC and DC Termination Requirements 7
    3.4.2. Transmitter Characteristics 8
    3.4.3. Scrambler Characteristics 9
    3.4.4. Synchronization 11
    3.4.5. Logical Characteristics 13
    3.4.6. Receiver Characteristics 14
    3.4.7. Transmission Limitations 14
4. Protocol Overview 15
5. Link Level Protocol 16
  5.1. Transport Protocol (FDHP) 16
    5.1.1. General 16
    5.1.2. FDHP Basic Structure 17
    5.1.3. Transmit and Receive States 17
    5.1.4. State Diagrams 17
    5.1.5. Typical Message Transfer 22
  5.2. Link Level Message Protocol 23
    5.2.1. Basic Structure 23

5.2.2. Central Office (CO) to Terminal Equipment (TE) 28

5.2.3. Terminal Equipment (TE) to Central Office (CO) Commands 29

5.3. Translation Table 35

5.4. CO and TE Command Inter-Action 36

6. Data Formats On The 64 Kbits/s Channel 40

6.1. Basic 56 Kbits/s End-To-End Data Channel 40

6.2. Recommended Data Channel Protocol 41

7. Network Maintenance 41

7.1. General 41

7.2. User Initiated Tests 41

7.2.1. TE Self Diagnostics 41

7.2.2. Local Loopback 41

7.2.3. Far End Loopback 41

7.3. CO Initiated Loopbacks 42

ATTACHMENT 1 - End-to-End Data Channel Protocol (T-Link/MHP) 43

ATTACHMENT 2 - Clock Generation and Recovery 88

TECHNICAL REFERENCE NOTICE

This Technical Reference is published by Ameritech to inform the industry of Ameritech's view of the Technical Interface Specifications for Northern Telecom's Datapath Network Access Interface.

Ameritech reserves the right to revise this document for any reason, including but not limited to, conformity with standards promulgated by various agencies, utilization of advances in the state of the technical arts, or the reflection of changes in the design of any equipment, techniques or procedures described or referred to herein. Liability to anyone arising out of use or reliance upon any information set forth herein is expressed or implied, are made with respect to the accuracy or utility of any information set forth herein.

This document is not to be construed as a suggestion to any manufacturer to modify or change any of its products, nor does this document represent any commitment by Ameritech, or any Ameritech Operating Company to purchase any product, whether or not it provides the described characteristics.

Nothing contained herein shall be construed as conferring in implication, estoppel or otherwise any license or right under any patent, whether or not the use of any information herein necessarily employs an invention of any existing or later issued patent.

Nothing contained herein is intended as a recommendation of any product to anyone.

Document may be ordered from Ameritech by contacting the Document Order Center at (847) 248-4324.

1. Introduction

1.1. General

This document is intended as a disclosure document that defines the performance and compatibility requirements for terminal equipment that will be directly connected to the interface for Datapath service. Datapath service will be provided from a Northern Telecom DMS-100 (DMS is a trademark of Northern Telcom, Limited) Family Switch within some or all of the companies within the Ameritech Region. This service is planned to provide the user with a circuit switched data transmission service. Due to the current digital transmission network restriction, it is planned that the maximum data rate that will be provided is 56 kbits/s synchronous/asynchronous data end-to-end for interoffice applications. Datapath does have the capability to transmit 64 kbits end-to-end on intraoffice calls. The loop facility will provide a full duplex 72 kbits/s channel with 8 kbits/s used for a signaling channel for messages between the terminal equipment (TE) and the central office (CO). The remaining 64 kbits/s will be the data channel.

The data channel is used for PCM encoded call progress tones for the origination end and ringing at the terminating end during call set-up and for data once the end-to-end connection has been established.

FCC rules for the connection to this service have not yet been promulgated. Until such time as they are, this document will act as the technical requirement for all connection CPE to Datapath. This interim plan will be superceded by Part 68 and Part 15 Network Protection Criteria, when issued. Specifications pertinent to the interoperability with the network will remain in this document.

Another technology exists for switched digital service, AT&T's CSDC, with a different interface and end-to-end signaling features. A prudent vendor should be aware of its interface characteristics. This can be found in Telcordia (formally Bellcore) TR880-22135-84-01.

1.2. Terminology

The following terms and abbreviations are used within this specification:

AMI	Alternate Mark Inversion
Byte	A group of eight bits
Checksum or CHSum	The nibble that is transmitted when the RACK state is indicated. The value of Checksum is determined by summing the four data nibbles and using the four LSB of the sum as the Checksum.

CHO	This is the transmitter state indicated when data is contained within the byte.
DLC	Data Line Card
DMS Machine	DMS-100 Family Northern Telecom Digital Multiplex System
DTE	Date Terminal Equipment
DTR	Data Terminal Ready
FDHP	Full Duplex Handshaking Protocol
I	Interface
Idle	This is the transmitter state indicated when there is no data transmitted in the byte.
Inband Protocol	The protocol used on the 64 kbits/s data channel
LSB	Least Significant Bit
Message	Any of the messages as defined in the TE/DLC Message Protocol Specification (contained within a 16 bit envelope)
MSB	Most Significant Bit
NACK	Negative ACKnowledgement is the receiver state transmitted when the received Checksum does not agree with the receiver's calculated Checksum or when the received state is not the expected received state.
NCTE	Network Channel Terminating Equipment
Nibble	A group of four bits (half a byte)
No-op	No-operation
PACK	Positive ACKnowledgement is the receiver state transmitted when the received Checksum agrees with the receiver's calculated Checksum.
RACK	Reqest ACKnowledgement is the transmitter state that is transmitted when the receiver is requested to acknowledge receiving the data nibbles either correctly or incorrectly. The Checksum transmitted with the RACK is what the receiver uses to perform the validity test on the received data.
RI	Ring Indication
RTS	Request to Send
TCM	Time Compression Multiplexing
TE	Terminal Equipment

1.3. Applicable Document

TR-880-22135-84-01 - Bell Communications Research, Inc. Technical Reference, July 1984 Issue 1, Circuit Switched Digital Capability Network Interface Specification (Appendix A, Cable Characteristics 1 Hz to 5 MHz)

2. Interface Overview

Access to the Datapath service is through a jack on the customer premises. This jack is provided with a two wire tip-and-ring interface.

The functions and capabilities of the network are accessed through the proper physical and electrical interfacing and the interface protocol of the signal channel.

Table 2-1 summarizes the interface characteristics. Detailed information on the interface compatibility and performance requirements is given in Sections 3, 4, and 5.

SUMMARY OF INTERFACE CHARACTERISTICS Table 2-1
TCM transmission over 2-wire loop
Line transmission rate: 160 kbits/s
Modulation scheme: 50% BIPOLAR, (Alternate Mark Inversion)
Transmit power: 10 dBm average
Maximum loop range: 5.4 km (Nominal)
Maximum insertion loss: 45 dB
Clock recovery from received burst
Overvoltage Protection
DC termination of 2 kohms for 6 mA + 1 mA Sealing Current
AC Termination of 135 ohms (5 kHz to 160 kHz)

3. Interface: Physical and Electrical Connection Requirements

The requirements detailed here give the physical layer requirements necessary to provide Datapath service. Datapath provides a DMS machine with a circuit switched digital data capability.

At the two wire interface that is defined here Time Compression Multiplexing (TCM) is used to transfer bidirectional digital signals at an effective data rate of 72 kbits/s full duplex. The 72 kbits/s is configured as an 8 kbits/s signal channel plus a 64 kbits/s data channel. The TCM is used to transfer the signaling plus data information across this interface by alternately time interleaving busts of data in the transmit and receive directions (sometimes called Ping-Pong).

The line rate for TCM transmission is 160 kbits/s. The TCM is based on a 1 millisecond time frame with each frame containing two bursts of data. The first burst of data is sent from the (master) DLC (data line card) in the DMS machine to the TE (slave) i.e. DLC is in the transmit mode, the TE is in the receive mode.

During the latter half of the frame, the second burst will be transmitted. For this burst the transmit and receive directions will be interchanged and the TE (slave) will transmit to the DLC (master).

The format of each of these bursts is identical.

The TCM burst that is transmitted from the DLC (master) will be clocked according to a master clock on the DLC side of the interface. The master clock controls the transmission of bursts at 1 millisecond intervals. The TCM burst that is transmitted from the TE (slave) will be timed with timing information derived from the received DLC TCM burst.

The mechanical, electrical and logical connection requirements are given here to enable correct interworking between the equipment on either side of the interface.

3.1. Physical Connection Requirements

The interface consists of a single pair of conductors (Tip and Ring) connected to the Terminal Equipment on one side and to the data line card in the DMS machine via the access line on the other side as shown in Figure 3-1.

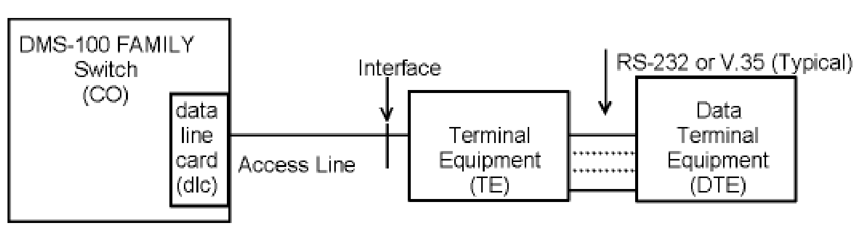
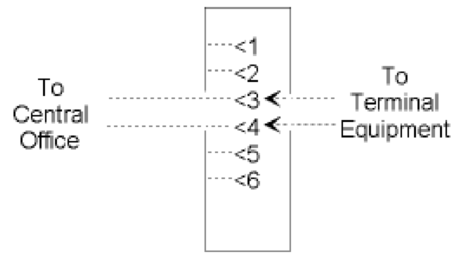


Figure 3-1. Datapath System Physical Configuration

3.1.1. Telco Provided Interface

The physical connection of the TE to the network is made using a miniature 6-position jack. The jack used is as shown in FCC Rules and Regulation, Part 68, Subpart F, Figures 68,500 (b) (1) and 68.500 (b) (2). This connection configuration is shown in Figure 3-2 below:



3.2. Power Requirements

The power for the terminal equipment shall be provided locally by the customer.

3.3. Sealing Current

The DLC provides a sealing current of 6 mA ± mA over the access line through the interface. The sealing current is to ensure that continuity of the two-wire metallic loop is maintained. CPE must provide a DC parameter to the sealing current.

3.4. Electrical Connection Requirements

3.4.1. AC and DC Termination Requirements

3.4.1.1. DC Characteristics

The Terminal Equipment shall present a nominal DC resistance of 2 kohms across the interface.

3.4.1.2. AC Characteristics

At the interface the Terminal Equipment shall present a balanced termination with an impedance of 135 ohms ± 10 percent over a frequency range of 5 kHz.

3.4.2. Transmitter Characteristics

The line driver in the TE must be enabled only during the transmit period of the TCM frame. The line signal must be Alternate Mark Inversion (bipolar), Return to Zero, 50% duty cycle. To reduce crosstalk on the loop facility, the transmitted signal should be shaped by a first order Low Pass filter. The corner frequency of the filter must be at approximately 260 kHz.

3.4.2.1. Transmit Pulse Characteristics

The transmit pulse characteristics given in Table 3-1 below represent the Bipolar signal prior to being passed through the first order filter defined above.

Table 3-1

Pulse Type	Bipolar Return-to-Zero
Duty Type	50%
Pulse Repetition Period	6.25 microseconds ± 50 nanoceconds
Pulse Height at NI (Into 135 ohms)	2.4 v ± .15v
Pulse Width (W+, W–)	3.125 microseconds ± 100 nanoseconds

Pulse Overshoot	h ± less than or = 0.01A h – less than or = 0.01A
Maximum Dynamic Pulse Jitter	± nanoseconds

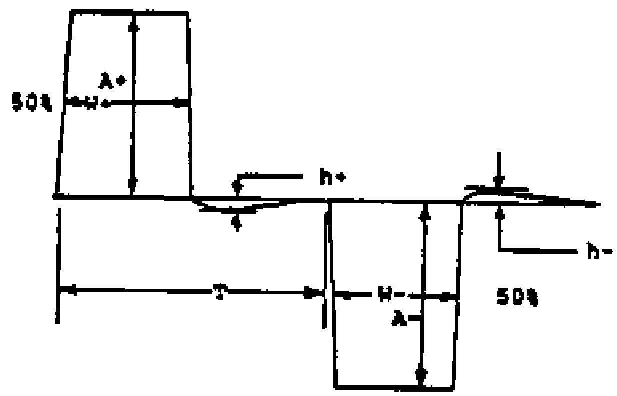


Figure 3-3
Transmit Pulse Parameters

3.4.2.2. Transmitted Signal Power

The average power is a TCM burst transmitted by the TE into a 135 ohms termination at the interface shall be a maximum of 10 dBm.

The peak power in a TCM burst transmitted into 135 ohms shall be a maximum of 16 dBm.

3.4.3. Scrambler Characteristics

A scrambler and descrambler shall be provided to randomize the data to ensure there are no spectral peaks in the transmitted bursts. Also to provide sufficient density of "1" bits in the bursts to facilitate timing recovery.

The scrambler and descrambler shall be frame synchronized. This means that with the exception of the start and stop bits, that frame the TCM bursts, the 72 bits shall be exclusive ORed with a pseudo-random bit pattern. This applies to both the transmitted data and the received date. Prior to the data being coded into bipolar form and being transmitted, it is fed into one of the inputs of a two input exclusive OR gate with the bit pattern fed into the second input coming from the pseudo-random generator. The bit pattern the generator shall use is given in Table 3-2. The start and stop bits are not passed through the scrambler or descrambler.

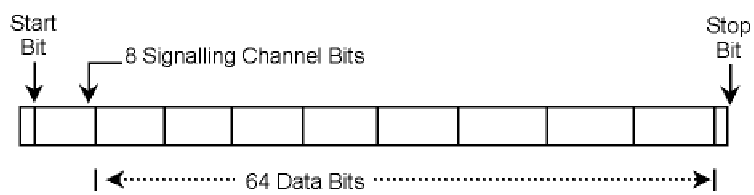
Table 3-2 Psuedo Random Generator Output				
Bit Count	Value		Bit Count	Value
0	1		36	1
1	0		37	0
2	1		38	0
3	1		39	1
4	0		40	1
5	1		41	0
6	1		42	1
7	0		43	1
8	0		44	1
9	0		45	0
10	1		46	1
11	0		47	0
12	1		48	0
13	0		49	0
14	0		50	1
15	0		51	0
16	0		52	0
17	0		53	0
18	1		54	1
19	1		55	1
20	1		56	0
21	0		57	1
22	1		58	0
23	1		59	1
24	0		60	1
25	1		61	0
26	0		62	0
27	0		63	1

28	1		64	1
29	1		65	1
30	1		66	0
31	1		67	0
32	0		68	1
33	0		69	1
34	0		70	0
35	0		71	0

3.4.4. Synchronization

3.4.4.1. Transmit Bursts (CO to TE)

The CO shall control the burst sequence based on a 1 msec frame period. The CO transmits a burst of 74 bits at 160 kbits/s as indicated in Figure 3-4.



Line Rate 160 kbits/s (T-6.25 micro seconds)

TCM Frame 1 ms

Burst Length 74 bits

Figure 3-4. Loop Burst Format

3.4.4.2. TE Synchronization

The transmit clock of the TE should be frequency locked to a clock derived from the received TCM bursts.

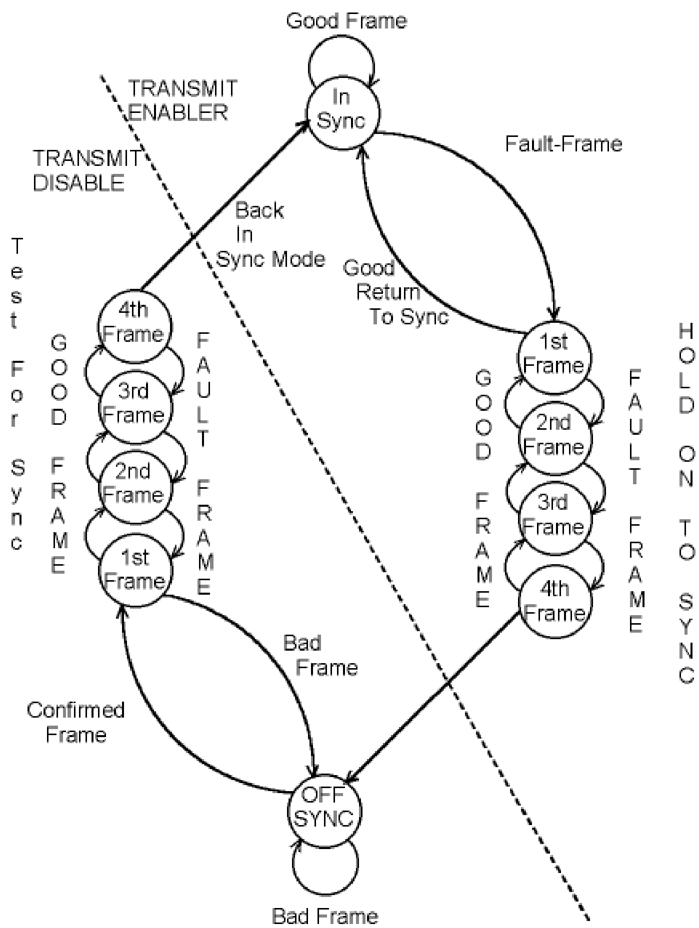
Timing the transmission and reception of data to and from the user device (DTE) can be accomplished in a number of ways. One recommended solution is described in Appendix B.

3.4.4.3. Sync Loss and Recovery

The status of the interface in terms of synchronization is defined by the four possible states indicated in Figure 3-5. The states are: in sync, hold on to sync, off sync and test for sync. Sync is determined by detaching the start and stop bits framing the data. A four step filter with a maximum count of four and a minimum count of zero is used for determining sync. For each good frame received the count is increased by one and for each bad frame received the count is decreased by one. When the count reaches four then the TCM system is considered in sync. While the TCM system is in sync, the four step filter system continues to keep count. Should the count drop to zero then the TCM system is considered out of sync. The definition of a good frame is if the period between the start and stop bits is .5375 ms (86 bit count). That is the time from the end of the received burst to the start of the next received burst. This indicates the full 1 ms frame can be accounted for (74 x 6.25 microsec. + .5375 ms = 1 ms).

On the TE side of the interface, if sync is lost the TE shall stop transmitting any data. The receive window is then opened so that any data on the loop can be received. The TE shall then start to hunt for the Start and Stop framing bits to reestablish sync.

On the CO side of the interface if sync is lost the receive enable window will move to the end of the receive window to ensure that the stop bit is received. The CO will then begin to hunt for the Start and Stop bits received to reposition the receive enable window. When the sync is lost the CO continues to transmit the framing bits as well as random data to the TE.



3.4.5. Logical Characteristics

3.4.5.1. Frame Structure

The CO, which acts as the master, operates on a frame rate of 1 ms. During each frame the CO transmits 64 bits of data, 8 bits of supervision or control signaling and 2 framing bits as shown in Figure 3-4. The same sequence of bits is received in the receive half of the 1 ms frame as shown in Figure 3-6.

For the received and transmitted bursts, the time required for each is 462.5 microsec or a total time of 0.925 ms. Therefore, the sum of the time delay introduced by the TE from the end of the received burst to the start of its transmit burst plus the round trip delay introduced by the subscriber's loop shall not exceed 75 microseconds. Hence, the rationale for the maximum one way delay introduced by the subscriber loop that can be tolerated is 37.5 microseconds assuming no delay is introduced by the TE. The maximum one way delay is reduced by a time interval equal to one bit time (6.25 microsecond) to provide a guard time at both the TE and the CO. The guard time is provided to allow for the TE and the CO circuitry to switch from the receive mode to the transmit mode.

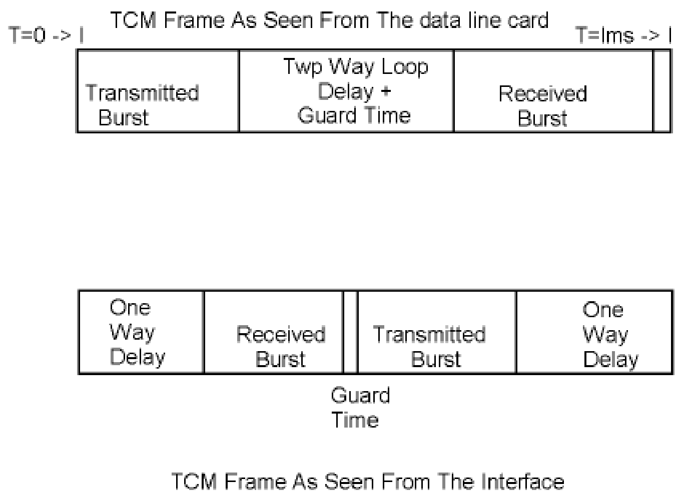


Figure 3-6. TCM Frame Transmission & Reception

3.4.6. Receiver Characteristics

The function of the line receiver is to recover data from the received signal. It shall be enabled only during the receive period of the TCM frame. The signal received at the interface will be generated by a driver on the DLC meeting the requirements given in Section 3.4.2 and modified by the cable facilities.

3.4.7. Transmission Limitations

3.4.7.1. General

The line code in both directions is bipolar Alternate Mark Inversion (AMI). The pulses have a nominal 50 percent duty cycle. The line transmission rate is 160 kbit/s (DLC to TE) with a receive rate of 160 kbit/s + 2 percent jitter (TE to DLC).

The maximum allowable access line attenuation at the Nyquist frequency (80 kHz) is 45 dB.

The maximum available one way access line delay is 31.25 microseconds.

Table 3-3. Loop Length Limits

(excluding other loop impairments)

Cable Gauge (AWG)	Max. Length for 45 dB Loss at 21 Degrees C, 80 kHz		Max. Length for 31.25 Micro Sec. Delay		Limiting Factor (°)
	km	kft	km	kft	
26	4.0°	13.1°	5.5°	17.9°	Attn.
24			5.7°	18.5°	Delay
22					Delay

3.4.7.2. Bridge Tap Limitations

The transmission limitations specified in Section 3.4.7.1 are based on access lines with no bridge tap. Those access lines that do have bridge taps, in addition to meeting the maximum loss requirements including the losses caused by the bridged taps, are required to meet the following requirements:

A. If all the bridged taps are less than 1 kft then only the loss constraint applies.

B. If one or more bridged tap exceeds 1 kft then the sum of all bridged taps shall be less than or equal to 2.5 kft.

4. Protocol Overview

The two wire Datapath interface to the network provides a 72 kbps full duplex transmission capability comprised of an 8 kbps signaling channel and a 64 kbps data channel. The kbps signaling channel allows communication between this switch an the terminal equipment (TE) using the Link Level protocol for network access, setup and special feature activation. This link level protocol must be used for network access and is described, in detail, in this document (Sections 4.2 and 5).

The 64 kbps data channel provides a network transparent end to the 56 kbps full duplex transmission path (8 kpbs of the 64 kbps, first described, is reserved to fill the network requirement of the 8th bit equal to 1). Within the 56 kbps data channel any protocol may be used for end-to-end transmission provided the terminal equipment at either end of the datapath call are equipped with the same or compatible protocol firmware. Although this network access and interface specification cannot define, as a requirement, the protocol to be used for end-to-end compatibility, included in Appendix A is a detailed description of the protocol used by Northern Telecom in their Datapath Data Units. This protocol, entitled KHP protocol, must be used by any terminal equipment (data unit) designed to be compatible with Northern Telecom Data Units or designed to access Datapath's modem pooling feature. Another technology exists for switched digital service, AT&T's CSDC, with different end-to-end signaling features. A prudent vendor should be aware of its interface characteristics. This can be found in Telcordia (formally Bellcore) TR 880-22135-84-01.

5. Link Level Protocol

The link level protocol described herein is a requirement for network access. This is the protocol structure used in the signaling channel for communication between the network and the terminal equipment. This protocol is comprised of two layers; the transport layer or FDHP (Full Duplex Handshaking Protocol) and the link level layer or TE/DLC message protocol.

5.1. Transport Protocol (FDHP)

5.1.1. General

The messages passed over the 8 kbit/s signaling channel are 16 bits long. The messages are defined by the TE/DLC message protocol specified in Section 5.2. The transport protocol that is used to send and receive the messages is called Full Duplex Handshaking Protocol (FDHP). FDHP is a byte oriented envelope protocol that provides error detection via checksum, error correction via re-transmission, and flow control. To transmit a single message (2 bytes) plus the checksum twice, will require six signaling bytes. Both the data and handshaking control information are designed to communicate in a full duplex mode. Every byte transmitted contains 3 types of information:

1. Transmit State (2 bits)
2. Receive State (2 bits)
3. Data Nibble (4 bits)

The checksum is determined by serial addition of each of the four data nibbles that form a message. Each data nibble is added to the sum as it is received or transmitted with the four least significant bits being used as the checksum.

5.1.2. FDHP Basic Structure

Full Duplex Handshaking Protocol is a byte-oriented envelope protocol. The byte format is shown in Figure 4-1 below.

MSB LSB

TxS1	TxS0	RxS1	RxS0	D3	D2	D1	D0

BYTE FORMAT
Figure 5-1

5.1.3. Transmit and Receive States

The transmit and receive states are indicated using the four MSB of the byte. The possible states are defined in Figure 5-2, Byte Contents, shown below:

BYTE CONTENTS

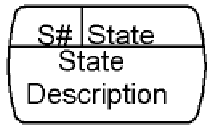
Figure 5-2

Transmit States	TxS1	TxS0	RxS1	RxS0	D3 D2 D1 D0
IDLE (no data)	0	0	X	X	X
CH0 (data on channel 0)	0	1	X	X	data nibble
RACK (request acknowledge)	1	0	X	X	checksum nibble
Receive States					
CTS (clear to send)	X	X	1	1	X
PACK (positive ACK)	X	X	1	0	X
NACK (negative ACK)	X	X	0	0	X
Note: X = Don’t Care					

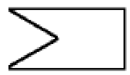
5.1.4. State Diagrams

The transmitter and receiver processes are represented by the states and the transitions between them. An input causes the process to leave a state and the flow of the state diagram varies depending on the input received. Whenever the state number reached is one that has occurred previously in the state diagram the flow of the state diagram reverts to the earliest occurrence of the state defined.

A number of definitions are given below to clarify the symbols used in the state diagrams that follow:



This symbol indicates a state. The numbers (S#) in the upper left corner indicates the state number and the state that is transmitted is indicated to the right of the S#.



Symbol for Received Inputs

Decision Symbol

Connector Symbol

Internal Task Symbol

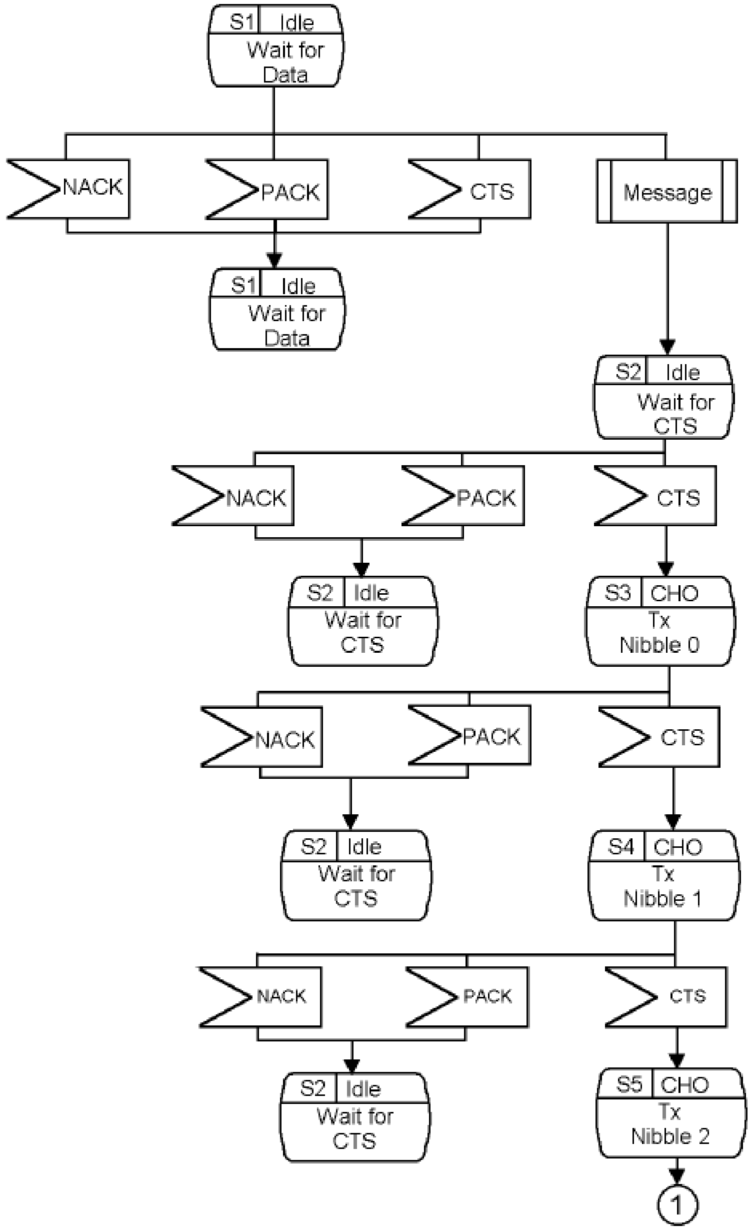


Diagram 5-2. Transmit State Diagram

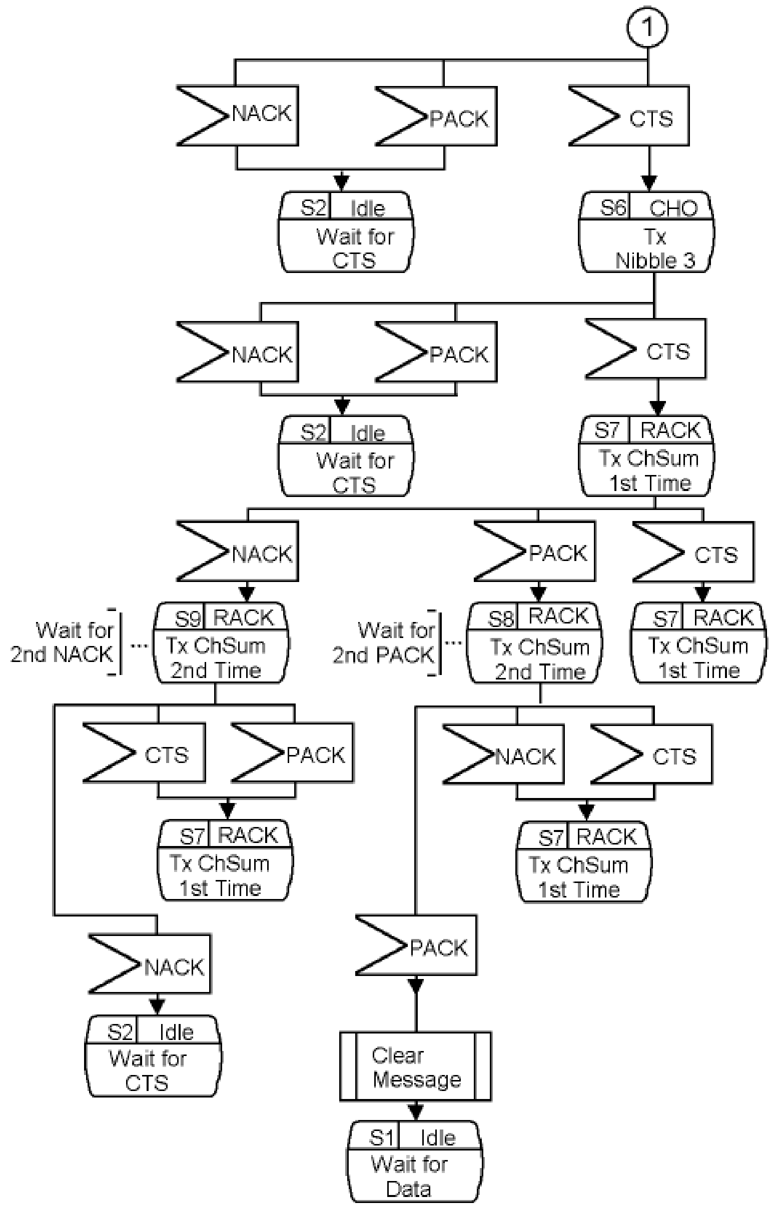


Diagram 5-2. Transmit State Diagram
(continued)

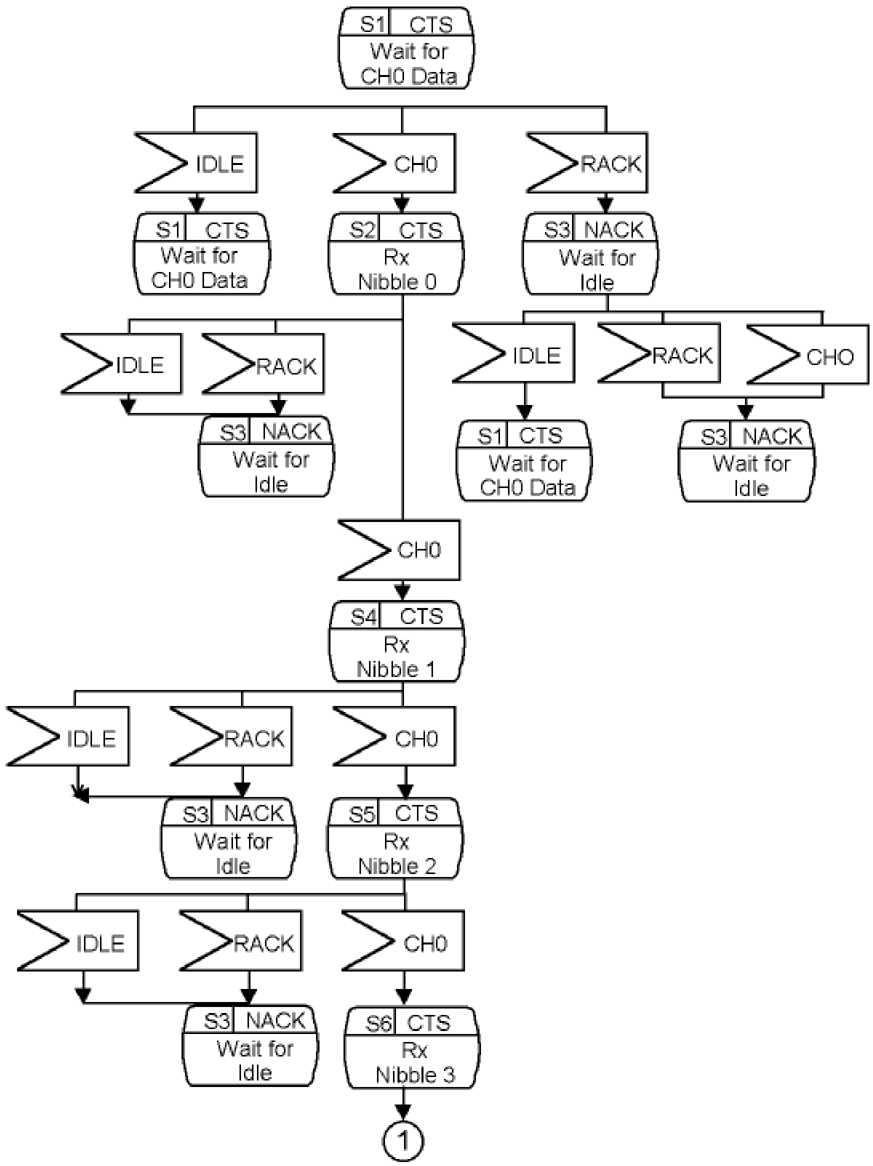


Diagram 5-3. Receiver State Diagram

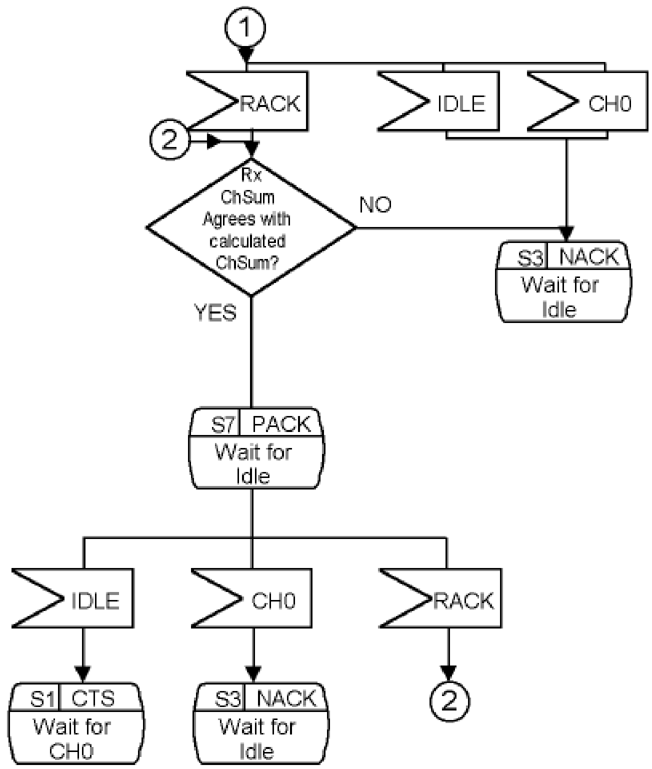


Diagram 5-3. Receiver State Diagram
(continued)

5.1.5. Typical Message Transfer

The following description is typical of a message transfer when the received checksum is correct and when there is no interruption in the message transmission introduced by a request from the receiver. The receiver can stop the transmission of the message at any time by entering the BACK state. If this does occur during the transmission of a message, the transmitter is required to re-transmit the complete message after the receiver re-enters the CTS state.

TRANSMITTER					RECEIVER		
Tx-state	Rx-state	Date			Tx-state	Rx-state	Data
				<···	X	PACK/NACK	X
IDLE	X	X	···>				
				<···	X	CTS	X
CHO	X	nibble0	···>				
				<···	X	CTS	X
CHO	X	nibble1	···>				
				<···	X	CTS	X
CHO	X	nibble2	···>				
				<···	X	CTS	X
CHO	X	nibble3	···>				
				<···	X	CTS	X
RACK	X	checksum	···>				
				<···	X	PACK	X
RACK	X	checksum	···>				
				<···	X	PACK	X
IDLE	X	X	···>				
				<···	X	CTS	X

Table 5-1. Typical Message Transfer

5.2. Link Level Message Protocol

5.2.1. Basic Structure

Datapath service provides a 64 kbits/s full duplex data channel that has an associated 8 kbits/s full duplex signal channel. The signal channel is used for call set-up as well as several other control functions outlined in the signal channel message protocol as defined below. The eight bit control commands contained within a sixteen bit envelope have been developed to handle call processing. The message protocol for Datapath service has several additional commands that are defined to provide the necessary communication between the Terminal Equipment and the CO. Both incoming and outgoing messages on the signal channel consist of 16 bits contained within a two byte structure as shown in Figure 5-3.

Figure 5-3. Basic Message Structure

This structure only represents the format of the message and does not represent how the message will be transported over the signal channel. The transport protocol for messages is defined by the Full Duplex Handshaking Protocol (FDHP) (See Page 5-1).

Byte 1	The first byte of the message to be passed over the interface. This is the high order byte.
Byte 0	The second byte of the message to be passed over the interface. This is the low order byte.
Address	The four MSB of byte 1 are reserved for the address. For Datapath service, the address is used to enable the DMS Machine to direct a message to either the DLC or to the TE. The DLC is assigned address 8 Hex (1000). The TE is assigned address 0 Hex (0000). Hence, the received messages from the CO relevant to the TE will always have address 0 Hex. Also, any messages transmitted by the TE will normally have address 0 Hex, thereby indicating to the CO that the message originated from the TE rather than the DLC.
Info	The information nibble of byte 1 is currently not used and should be set to logic level 0.
Command	These eight bits form the actual commands that are acted on by the TE or the CO.
Alert ON	This command alerts the terminal equipment of an incoming call and that the CO is transmitting a PCM u-law encoded ringing signal over the data channel.
Alert OFF	This command alert the TE that the ringing signal is no longer on the Data Channel.

Progress Tone Alert (ON)	This command alerts the TE to monitor the data channel for PCM - law encoded voiceband signals. When the TE is originating a call, this command monitors call progress tones and when the TE is answering a call it monitors voice band signals. The transmission from the voice-on state to the voice-off state may be triggered by receiving either the OFF command, or a reset command, if the ON command is followed by a Data Alert ON command. The detection of valid data by the TE turns off the progress tone alert.
Progress Tone Alert (OFF)	This command is not used at the present time but reserved for future use.
Feature Alert ON/OFF	These commands are for special feature signaling depending upon which feature was subscribed to. (Example: ring again recall)
Secondary Feature Alert ON/OFF	These commands alert the TE of a second special feature which was subscribed to.
Data Alert ON	This command alerts the TE to monitor the data channel for valid data. "Data", as defined here, includes any in-band handshaking protocol.
Data Alert OFF	This command is not used at the present time but is reserved for future use.
TE Status Request (Data Rate)	This command requests the TE to report present operating data rate of the TE. This request command will only be used if the modem pooling feature is subscribed to.
TE Status Request (loopbacks)	This command requests the TE to report the loopback status of the TE should the TE have loopback capability (Loopback capability is not a requirement of and (TE). This command can occur at any time.
TE Status Request (Near End)	This command request the TE to report status of the near end TE/DTE interface and can occur at any time. It is not a requirement for any TE to respond to this command.
TE Status Request (Far End)	This command requests the TE to report status of the far end TE/DTE interface as indicated by the in-band protocol should the TE employ the KHP protocol described in Appendix A. The TE is not required to respond to this command.
Loopbacks Off	This command requests a reset of any loopback previously established by the CO (Loopback capability is not a requirement of any TE).
Data Channel Loopback	This command requests the TE to place the 64 kbps data channel within the TE in a loopback state (Loopback capability is not a requirement of any TE).
TE/DTE Loopback	This command shall request a loopback of the Data Channel at the local TE/DTE interface (Loopback capability is not a requirement of any TE).

Far End DTE/TE Loop-back	This command shall request the local TE to initiate an in-band request to the far end TE to establish a loopback at the DTE/TE interface. This command assumes that both TEs use the in-band protocol described in Appendix A. (Loopback capability is a requirement of any TE).
Message Echo Start/Stop	This command requests the TE to loopback all subsequent commands in the signal channel to the DLC. While in the Echo mode no incoming commands will be acted upon by the TE until the echo message stop command is received. TE keyboard inputs are acted upon. (Loopback capability is not a requirement of any TE).
Under Test Alert (On/Off)	This command alerts the TE that the CO is seizing the TE or access line for testing and may not be used for calls.
Call Processing Reset	This command is used when a TE to TE connection is taken down. It is a single command that combines the functions of the Alert Off, the Call Progress Alert Off, and the Feature Alert Off commands.
Soft Reset	This command is sent by the CO to indicate that the switch has reset itself. This command shall perform all three set functions defined for the Call Processing Reset Command in addition to signaling the TE that all previously requested special features cancelled.
Hard Reset	This command is sent by the CO after loop maintenance procedures are completed to insure that the TE is left in a known state. This command shall perform all the reset functions defined by the soft reset in addition to resetting any loopback state and the FDHP transmitter/receiver.
Feature Indicator	These commands send feature status messages to the TE. The status messages can be one of four states: ON, OFF, Primary Indicator and Secondary Indicator. An example of a primary versus a secondary indicator is of a lamp flashing at 120 vs. 60 interruptions per minute. The last three bits of these command bytes are left open so that they can be mapped to the different features available (See Table 5-2 on feature command codes and Section 5.3).

RI Cycle On	This command indicates to the TE that an incoming call exists. It can be used as an alternate means of signaling should the TE not wish to use the PCM encoded ringing signal.
Ri Cycle Off	This command turns off the ringing indicator described in RI Cycle On.

Table 5-1
CO to TE Command List

Function	Command Code								HEX
	M7	M6	M5	M4	M3	M2	M1	M0	
Alert On	0	1	1	0	1	1	1	1	6F
Alert Off	0	0	0	0	1	1	1	1	0F
Progress Tone Alert On	0	1	1	0	1	1	0	0	6C
Progress Tone Alert Off	0	0	0	0	1	1	1	1	0C
Feature Alert On	0	1	1	0	1	1	1	0	6E
Feature Alert Off	0	0	0	0	1	1	1	0	0E
Secondary Feature Alert On	0	1	1	0	1	0	1	0	6A
Secondary Feature Alert Off	0	0	0	0	1	0	1	0	0A
Data Alert On	0	1	1	1	0	0	0	0	70
Data Alert Off	0	0	0	1	0	0	0	0	10
TE Status Request (Data Rate)	0	0	0	1	0	0	0	1	11
TE Status Request (Loopback)	0	0	1	1	0	0	0	1	31
TE Status Request (Far End)	0	1	1	1	0	0	0	1	71
TE Status Request (Near End)	0	1	0	1	0	0	0	1	51
Loopbacks Off	0	0	0	1	0	1	0	0	14
Data Channel Loopback	0	0	1	1	0	1	0	0	34
TE DTE Loopback	0	1	1	1	0	1	0	0	74
Message Echo Stop	0	0	0	0	1	0	0	1	09
Message Echo Start	0	1	1	0	1	0	0	1	69
Under Test Alert (On)	0	1	1	1	0	1	1	1	--
Under Test Alert (Off)	0	0	0	1	0	1	1	1	17
Call Processing Reset	0	1	0	0	1	0	0	0	48
Soft Reset	0	0	0	0	1	0	0	0	08
Hard Reset	0	1	1	0	1	0	0	0	68
RI Cycle On	0	1	1	1	0	1	0	1	75
RI Cycle Off	0	0	0	1	0	1	0	1	15
Feature Indicator On	0	1	1	0	0	D	D	D	0X
Feature Indicator Off	0	0	0	0	0	D	D	D	6X
Primary Indicator	0	0	1	0	0	D	D	D	2X
Secondary Indicator	0	1	0	0	0	D	D	D	4X

NOTE: "DDD" indicates the binary code associated with a special subscription feature. For example, the indicator message for the Auto Dial Feature would be "01100010 62" (See Section 5-4).

5.2.2. Central Office (CO) to Terminal Equipment (TE)

See Table 5-1 for the actual command codes)

Table 5-1
CO to TE Command List

Function	Command Code								HEX
	M7	M6	M5	M4	M3	M2	M1	M0	
Alert On	0	1	1	0	1	1	1	1	6F
Alert Off	0	0	0	0	1	1	1	1	0F
Progress Tone Alert On	0	1	1	0	1	1	0	0	6C
Progress Tone Alert Off	0	0	0	0	1	1	1	1	0C
Feature Alert On	0	1	1	0	1	1	1	0	6E
Feature Alert Off	0	0	0	0	1	1	1	0	0E
Secondary Feature Alert On	0	1	1	0	1	0	1	0	6A
Secondary Feature Alert Off	0	0	0	0	1	0	1	0	0A
Data Alert On	0	1	1	1	0	0	0	0	70
Data Alert Off	0	0	0	1	0	0	0	0	10
TE Status Request (Data Rate)	0	0	0	1	0	0	0	1	11
TE Status Request (Loopback)	0	0	1	1	0	0	0	1	31
TE Status Request (Far End)	0	1	1	1	0	0	0	1	71
TE Status Request (Near End)	0	1	0	1	0	0	0	1	51
Loopbacks Off	0	0	0	1	0	1	0	0	14
Data Channel Loopback	0	0	1	1	0	1	0	0	34
TE DTE Loopback	0	1	1	1	0	1	0	0	74
Message Echo Stop	0	0	0	0	1	0	0	1	09
Message Echo Start	0	1	1	0	1	0	0	1	69
Under Test Alert (On)	0	1	1	1	0	1	1	1	--
Under Test Alert (Off)	0	0	0	1	0	1	1	1	17
Call Processing Reset	0	1	0	0	1	0	0	0	48
Soft Reset	0	0	0	0	1	0	0	0	08
Hard Reset	0	1	1	0	1	0	0	0	68
RI Cycle On	0	1	1	1	0	1	0	1	75
RI Cycle Off	0	0	0	1	0	1	0	1	15
Feature Indicator On	0	1	1	0	0	D	D	D	0X
Feature Indicator Off	0	0	0	0	0	D	D	D	6X
Primary Indicator	0	0	1	0	0	D	D	D	2X
Secondary Indicator	0	1	0	0	0	D	D	D	4X

NOTE: "DDD" indicates the binary code associated with a special subscription feature. For example, the indicator message for the Auto Dial Feature would be "01100010 62" (See Section 5-4).

Table 5-2
TE to CO Command List

Function	M7	M6	M5	M4	M3	M2	M1	M0	HEX
	Command Code								
DN Message	0	0	0	0	0	0	0	0	00
Release Message	0	0	0	0	1	1	1	1	00
Local DTR On	0	0	1	0	0	0	1	1	23
Local DTR Off	0	0	1	0	0	0	1	0	22
Resource	0	0	0	0	0	0	0	1*	01
Auto Dial	0	0	0	0	0	0	1	0*	02
Speed Call	0	0	0	0	0	0	1	1*	03
Ring Again	0	0	0	0	0	1	1	0*	06
Originate	0	0	1	0	0	1	1	0	26
Answer	0	0	1	0	0	1	1	1	27
Inband Sync Lost	0	0	1	0	0	1	0	0	24
Inband Sync Found	0	0	1	0	0	1	0	1	25
Data Rate Status	0	1	0	0	X	X	X	X	4X
Loopback Status	0	1	0	1	X	X	X	X	5X
Near End Status	0	1	1	0	X	X	X	X	6X
Far End Status	0	1	1	1	X	X	X	X	7X
Dial Pad Messages ‘1’	0	0	0	0	1	0	0	0	08
‘2’	0	0	0	0	1	0	0	1	09
‘3’	0	0	0	0	1	0	1	0	0A
‘4’	0	0	0	0	1	1	0	0	0C
‘5’	0	0	0	0	1	1	0	1	0D
‘6’	0	0	0	0	1	1	1	0	0E
‘7’	0	0	0	1	0	0	0	0	10
‘8’	0	0	0	1	0	0	0	1	11
‘9’	0	0	0	1	0	0	1	0	12
‘0’	0	0	0	1	0	1	0	1	15
‘*’	0	0	0	1	0	1	0	0	14
‘#’	0	0	0	1	0	1	1	0	16

NOTE: The “*” indicates the LSB codes to be used for CO-TE indicator messages described in Table 5-1. The “X” indicates LSB codes used for indicator status (See 5.3.1 Command Descriptions).

5.2.3. Terminal Equipment (TE) to Central Office (CO) Commands

(See Table 5-2 for TE to CO Command Codes.)

5.2.3.1. Command Descriptions

DN Message	This command is sent to the CO to indicate a request for service has been made i.e. the equivalent of taking a telephone set off-hook.
Release Message	This command is sent to the CO to indicate that a call is terminated, i.e. the equivalent of placing a set on-hook.
Dial Pad Messages	These commands are sent to the CO and serve the same purpose as the keys on a standard dial pad of a phone, i.e. for network addressing.
Local DTR ON/OFF	These commands indicate the status of the DTR lead of the TE/DTE interface. A call cannot be originated or answered unless the DTR ON message has been sent. The only two exceptions to this rate are when the originate and answer commands are used. If during an established call a DTR OFF message is sent, the CO will break down the call.
Resource	This command requests the modem pooling resources, if available, from the CO. In addition, this command can be used, in the future, to make additional features and resources available for the TE.
Auto Dial	This command initiates the CO to place a call to a preset directory number.
Speed Call	This command is an indication to the CO that the following dial pad entries are for access to a programmed directory number in a list of frequently-called numbers.
Ring Again	This command may be used when the originating party reaches a busy directory number to instruct the CO to monitor the called Directory Number. When the called DN is no longer busy, the CO alerts the originator using the Feature Alert-ON and the Feature Alert-Off sequence.

NOTE: The above commands on Northern Telecom's TE (Data Unit) are associate with specific keys on the TE.

Inband Sync-Found	The command is sent to the CO when the TE to TE connection has been made and the TE has detected that valid data is being received as per the inband protocol. At the present time this is only used where the modem pooling feature is provided by the local Ameritech Operation Company (AOC(|). It may be used in the future for other purposes.
Inband Sync-Lost	At the present time this command is only used where the modem pooling feature is provided by the Telco. It may be used in the feature for other purposes.
Originate	This command provides the same function as the DN command for originating calls. Using the originate command also automatically indicates to the CO that the local DTR is on.
Answer	The command accomplishes the same function as the DN command for answering. It is primarily used for an auto answer capability. Before this command is sent, the TE must receive the RI Cycle ON command. Sending the Answer command automatically indicates to the CO that the DTR is on.
Data Rate Status	This command may be sent when the TE receives the TE Status Request-Data Rate command from the CO. This command may be used by the CO to perform diagnostics, using its own TE resident in the CO, which can be configured to be compatible with the customer's data rate adaption, if a common protocol is used.

NOTE: The following command messages are optional and are necessary only if the TE includes these status capabilities.

Loopback Status	This command may be sent in response to receiving a TE Status Request-Loopbacks command from the CO.

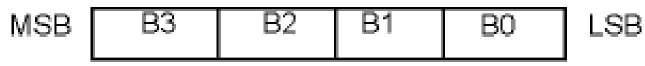


Figure 5-4. Loopback Nibble

Loopback Status (Cont'd)	The four LSB of this command, as shown in Figure 5-2, indicate the loopback status of both the near and far end TE as follows: Bit "B3" shall be set to 1 if the local TE is requesting that the far end TE/DTE interface be looped back. This can occur either as a result of a switch setting on the local TE or due to a previous command set by the CO, otherwise this bit shall be 0. Bit "B2" shall be set to 1 if the local TE/DTE interface is looped back either as a result of switch setting on the local TE or due to a previous command sent by the CO, otherwise this bit shall be 0. Bit "B1" shall be set to 1 if the far end TE has placed the local TE/DTE interface in a loopback state using the in-band protocol. This bit shall be 0 when this condition does not exist. Bit "B0" shall be set to 1 if the local 64 kbits/s data channel is looped back within the TE, otherwise it shall be 0.
Near End Status	This command may be sent in response to receiving a TE status Request-Near End command from the CO.

Figure 5-5. Status Nibble

Near End Status (Cont'd)	The four LSB of this command indicate the near end status of the TE/DTE interface as follows: The "RI" bit shall indicate the status of the RI lead of the TE/DTE interface. The "CTS" bit shall indicate the status of the CTS lead of the TE/DTE interface. The "RTS" bit shall indicate the status of the RTS lead of the TE/DTE interface. The "DTR" bit shall indicate the DTR lead status of the TE/DTE interface.
Far End Status	This command may be sent in response to receiving a TE Status Request-Far End command from the CO.

MSB	O	O	RTS	DTR	LSB

Figure 5-6. Status Nibble

Far End Status (Cont'd)	The four LSB of this command are used in the manner indicated in Figure 5-4. The command determines the far end TE/DTE interface status. The two MSB of the nibble as shown in Figure 5-4 are not in use at the present time and therefore shall be 0. The two LSB of this command are used in the same manner as were the two LSB of the previous command except their status reflect the far end TE/DTE status.
No-op Codes	Commands that do not match any of the commands defined in this specification shall be treated as no-op codes, i.e. they are acknowledged by no action taken.

5.3. Translation Table

The translation table, Table 5-3, enables the Ameritech Operating Company (AOC) to specify which features are going to be assigned for a specific customer installation. Key numbers 1, 2, 3, 4, and 7, are the assigning codes used by the AOC. Key 1 is always enabled in the customer's feature profile contained in the CO software since it forms part of the basic service offering. It is shown here to indicate the relationship between this key command and its related indicator. The other four keys are classified as optional, since they may or may not be subscribed to by a customer that is provided with the service. This means that the customer profile in CO software can be set to select any combination of the four optional features.

Table 5-3

Translation Codes

Key No.	Command Code - TE to CO (See Table 5-2)	Indicator Code - 'DDD' (See Table 5-1)	Ind. No
1	00 - DN	000	1
2	01 - Resource	001	2
3	02 - Auto Dial	010	3
4	03 - Speed Call	011	4
7	06 - Ring Again	110	7

As an example, if the key number 7 feature were enable for a particular customer's profile, then the TE sending the "Ring Again" command would activate this feature under the proper circumstances. Also the change of the "Ring Again" feature would be indicated to the TE using the indictor commands as follows:

(Key 7 - DDD = 110)

Feature Alert - On = 0110 0110

Feature Alert - Off = 0000 0110

Primary Indicator = 0010 0110

Secondary Indicator = 0100 0110

5.4. CO and TE Command Inter-Action

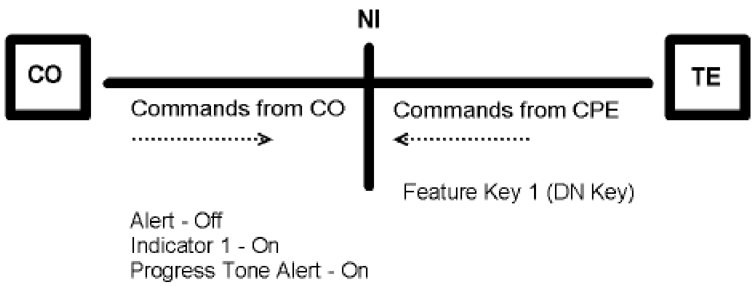
The CO functions on the basis of stimulus signaling from the TE received in the form of commands passed over the 8 kbps signal channel. The CO in turn expects the TE to react to the commands sent to the TE as per the command protocol. The following sequences are provided as an illustration of the CO and TE interaction across the Network Interface.

Command Sequence 1

Situation: *DN Key is pressed*

Assumptions:

1. TF is Idle
2. Square brackets, [ ], indicates u-law encoded voiceband signal on the 64 kbit/s TE receive channel.



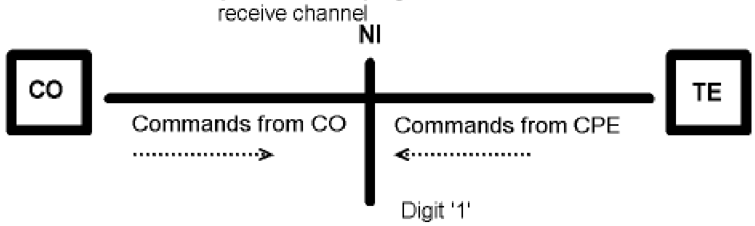
[Call Progress Tones]

Command Sequence 2

Situation: *To Dial Ext. '1305'*

Assumptions:

1. Diat Tone is present
2. Indicator 1 is On
3. Square brackets, [ ], indicates voiceband (u-law encoded) signal on the 64 kbit. TE receive channel



Digit '1'
[Dial Tone Stops]
Digit '3'
Digit '0'
Digit '5'
Dial Alert - On

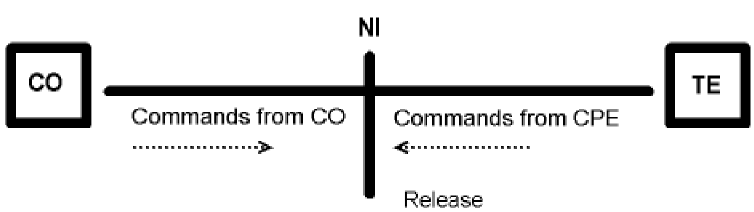
[Call Progress Tones]

Command Sequence 3

Situation: *Aborting Call*

Assumptions:

1. Call may be at any stage
2. Indicator 1 is On



Release

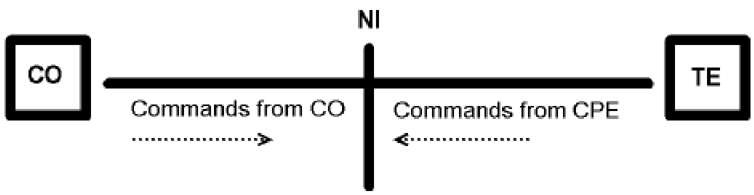
Call Processing Reset
Indicator 1 - Off

Command Sequence 4

Situation: *Auto Answer*

Assumptions:

1. TE is idle
2. DTR is On
3. Indicator 1 - Off
4. TE is set to auto answer calls



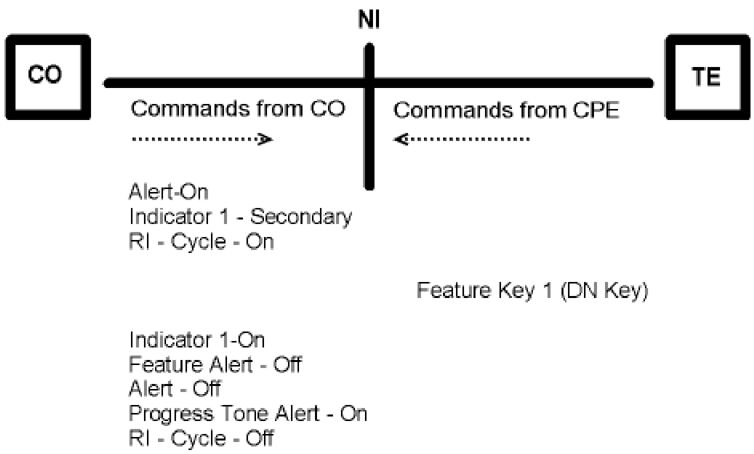
Alert-On
Indicator 1 - Secondary
RI cycle - On
[u-Law Encoded Ringing On 64 kbit Data Channel]
Answer
[Turn off Ringing]
Indicator 1 - On
Feature Alert - Off
Alert - Off
Progress Tone Alert - On
Data Alert - On
RI cycle - Off

Command Sequence 5

Situation: *Manual Answer*

Assumptions:

1. TE is idle
2. Indicator 1 - Off
3. Local DTR - On



6. Data Formats On The 64 Kbits/s Channel

6.1. Basic 56 Kbits/s End-To-End Data Channel

The 64 kbits/s full duplex data channel between the TE and the CO is reduced to a maximum end-to-end capacity of 56 kbits/s. This is due to the constraint that each byte shall be transmitted with 68 set to one to assure at least one “1” per byte. Therefore, the general format for all bytes, once an end-to-end connection has been established is:

MSB							LSB
X	X	X	X	X	X	X	1
b1							b8

Where “x” indicates the data bits.

The general format for all bytes received is the same as shown above with the exception that b8 may be received as either “1” or “0”.

6.2. Recommended Data Channel Protocol

To be compatible with Northern Telecom's TE or the modem pooling feature, it is a requirement that the end-to-end protocol, as defined in Appendix A (T-Link), be used for data transfer over the 56 kbps data channel. However, it is not recommended that any end-to-end protocol be included in any new TE design. Instead, the TE should be capable of data transfer at 56 kbps without an end-to-end protocol and have the ability to recognize and implement the T-Link protocol, as defined in Appendix A, should it reach an NTI TE when setting up a call.

The benefit of using the above recommendation is that it would not limit compatibility of any TE to one specific vendor's equipment.

7. Network Maintenance

7.1. General

Datapath service will have the capability for both automatic routine testing that may be initiated when a fault is suspected. To conduct some of these tests it is necessary that the TE be equipped with certain loopback features. Although these loopback features cannot be listed as interface requirements, it is highly recommended that they be included in all TE to eliminate the cost to the customer of an unnecessary field visit by the local operating company.

7.2. User Initiated Tests

7.2.1. TE Self Diagnostics

The provision of some form of TE self diagnostics is recommended as a means of enabling the user or installation personnel to verify the TE's satisfactory operation.

7.2.2. Local Loopback

When activated this feature is required to place the local TE/DTE interface in a loopback state in both directions, i.e. towards the DTE and towards the DLC. This allows the customer to verify the operation of his DTE. The status of this feature shall be indicated by the Loopback Status command (See the TE to CO commands).

7.2.3. Far End Loopback

When activated this feature tells the local TE to request the far end TE to loopback at its TE/DTE interface. The user can then check the integrity of the end-to-end data channel and verify the operation of the data encoding/decoding in the TE by examining the looped back data at the local TE/DTE interface. The status of this feature shall be indicated by the Loopback Status command (See TE to CO commands).

7.3. CO Initiated Loopbacks

The TE shall be capable of responding to the various loopback commands that can be initiated by the CO using the messages specified in Section 5.2. These loopbacks include loopback of the 64 kbits/s data channel, the signal channel and either the local or far end TE/DTE interfaces.

ATTACHMENT 1 - End-to-End Data Channel Protocol (T-Link/MHP)

1.1. Introduction

The terms “T-Link” and “MHP” message handshaking protocol) are used interchangeably throughout this document. They are two different names for the same end-to-end handshaking protocol described herein.

The terms “shall” and “must” are used in this recommendation on the basis that the recommendation is being implemented. The use of these terms indicates those specific characteristics that are required to be met to ensure the absolute minimum acceptable compatibility between various implementations of this protocol.

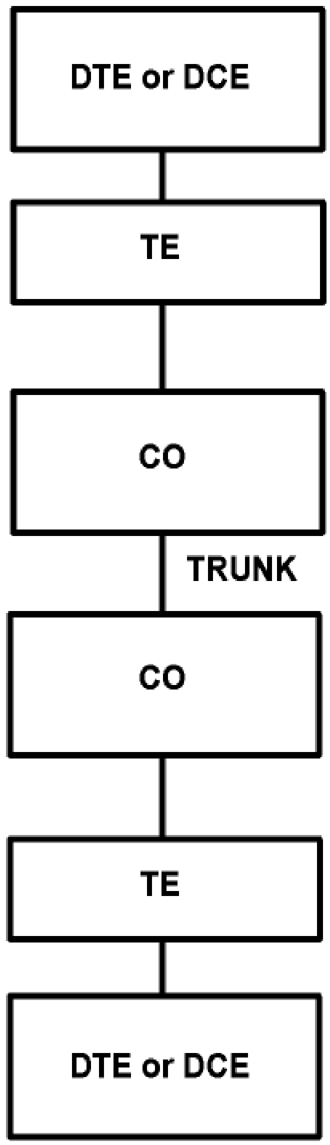
T-Link is a full duplex byte-oriented end-to-end protocol designed to transfer either synchronous or asynchronous data over a 64 kbps digital circuit at rates up to 56 kbps. There is a continuous flow of messages over the circuit and so it must be dedicated to T-Link for the duration of end-to-end communications once a connection has been established.

At low user data rates (less than or equal to 9600 bps) the protocol transmits each piece of data several times to allow error correction by voting on the received data. This method is well suited to systems where errors occur individually or in small groups and is much easier to implement than retransmission type error correction schemes.

T-Link treats the 64 kbps channel as if it consists of 8000 eight bit bytes per second. Figure 1.1 shows, in block diagram form, the hardware components in a hypothetical circuit using T-Link. At each end of the circuit there is the user’s terminal equipment (TE) that may have either a

TE/DTE or a TE/DCE interface beyond the two wire interface to the TE. For the purposes of this recommendation the TE/DTE or TE/DCE is regarded as the end of the circuit.

Figure 1-1. Sample Configuration
Using T-Link



1.2. An Example of T-Link

Before giving a detailed description of T-Link it is useful to look at the components of a typical call to give an overview of how T-Link works. Figure 2.1 illustrates the steps involved and also references the sections of the specification where further information on the various phases of a call can be found.

Before T-Link can be used it is necessary to establish an end-to-end 64 kbps digital connection (State 1 in Figure 2.1).

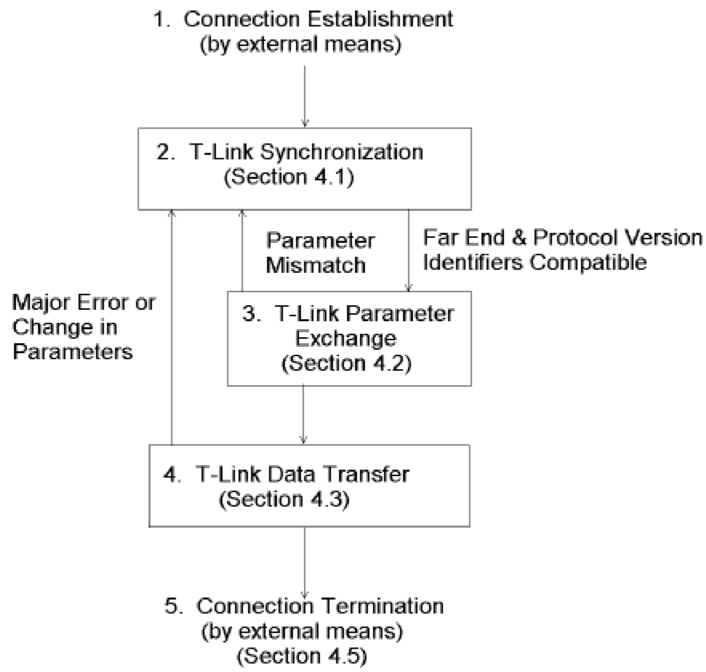
A switched connection may be set up in response to a user request, in which case a signaling mechanism between the end of the circuit and the switch must be established external to T-Link. This issue of establishing the connection will not be considered further in this specification.

Once an end-to-end connection exists the next phase is to synchronize the data transmitting and receiving circuitry at both ends of the connection (State 1 in Figure 2.1). This is done by repeatedly sending a signaling byte (referred to as Sgvi) from the called TE (call it A) to indicate a request to synchronize. Both TE's scan, the received T-Link bytes for Sgvi's when a T-Link call is not in progress and when the string of Sgvi's from "TE A" is received at the calling end (TE B), it goes into its synchronization mode and responds by sending Sgvi's back to "A". Following the Sgvi, the two TE's exchange a 6 byte data frame containing an Encoded T-Link protocol version identifier with "TE A" sending first and "TE B" responding. This ensures that both ends know the version of T-Link in use at the other end. If the T-Link versions at both ends are compatible the calls proceed and if they are not compatible the TE may either terminate the call and break the connection or repeat the synchronization process.

After synchronization the two ends shall send each other a series of five (or more) parameter messages which indicate the mode in which they would like to operate (State 3). The parameters defined such things as whether the data being sent will be synchronous or asynchronous and the data rate. If both ends send compatible parameters the call proceeds, otherwise the conflict must be resolved by having one TE adapt its parameters to match the other or by breaking the connection if adaption is not possible. The called TE is expected to adapt after the first parameter exchange (if necessary) and if it can not it may request that the call setup procedure be restarted and the parameter exchange be repeated with the calling TE expected to adapt.

If the call proceeds beyond the exchange of parameters the transmission of data can begin (State 4). For transmission rates less than 48 kbps signaling messages are initially sent to act as time fill and to indicate the levels on the control leads between the user's data equipment and the TE.

Figure 2-1. Stages in a T-Link Call



To allow a delay before the actual start of transmission either end can turn off the clear to send (CTS) signal at the remote end to disable it from sending. Transmission rates of 48 kbps and higher go from parameter passing directly to sending data. However, transmission by the far end can be delayed initially by sending continuous marks (all ones). Note that T-Link is a transparent protocol and does not modify any higher levels of protocol that may be in use end-to-end. Thus if different protocols are implemented in the user devices T-Link allows a flow of bits end-to-end but an exchange of information may not be possible.

The data formats will vary depending on the data rate and whether synchronous or asynchronous transmission is used end-to-end. Asynchronous data words are encoded into 2 byte pairs (4 data bits each) with signaling bytes used for time fill and to delineate each word as well as indicating the levels of control leads to and from the user's device. Synchronous data at less than 56 kbps is sent six bits at a time (in one byte message) and 56 kbps data is sent in 7 bits of a message byte.

Signaling bytes are also used for time fill with synchronous data, except at 48 and 56 kbps where every byte must contain user data. Data transmitted at up to 9.6 kbps is transmitted multiple times (3 times for asynchronous, 4 times for synchronous) to allow bit errors to be detected and removed. Higher rate data is transmitted only once.

There is no explicit end of transmission indication in T-Link. Data will continue to flow end-to-end until the connection is broken, reinitialized or until actions by the end users, at a higher protocol level, disable transmission via the control leads between the user and the TE (State 5). As was the case for connection establishment an external mechanism must be provided if it is desired to break a switched connection.

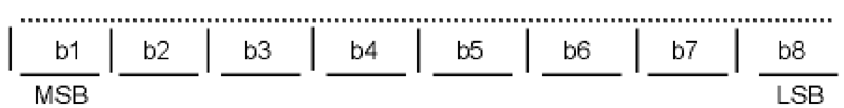
The following sections specify more details on the format and sequence of messages used by T-Link. Section 1.3 defines the byte-oriented message format used to transfer data and signaling information between the two ends. The specifics of parameter exchange and data discussed in Section 1.4.

Appendix B discusses data clock generation.

1.3. Byte Formats

1.3.1. Overall Characteristics

T-Link is a byte-oriented protocol with all signalling information and data expressed as bytes using the format:



where bit 1, the most significant bit is transmitted first and bit 8, the least significant bit, is sent last. It should be noted, however, that for historical reasons the actual data being transported will use a different numbering terminology (discussed more fully in Section 1.3.4). To avoid confusion, bits used by T-Link and sent over the channel will be called “b1” through “b8” while data bits will be referred to using the prefix “d” (e.g. d0, d1, d2....).

Within T-Link data and signaling bytes are distinguished by the value of “b7”, it is zero for bytes containing user or parameter data and one for bytes used to signal between the two ends of the circuit. The one exception to this is for 56 kbps data transmission that requires 7 bits of data per byte 8000 times a second, so once handshaking completes all bytes are treated as data bytes and “b7” contains a data bit. The general format of a T-Link byte as sent is therefore:

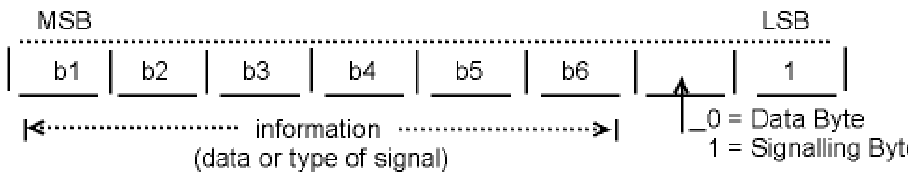


Table 3.1 summarizes the uses of the other bits in a T-Link byte and the remainder of this section defines the various formats in more detail. There are four main byte types: general selector bytes (Sg) which are used for time fill and to transport the interface control leads end-to-end, asynchronous data bytes (DH and DL) and synchronous data bytes (Ds). Each byte type has its own distinct format allowing the byte type to be uniquely identified in the absence of errors. Codes not shown in Table 3.1 are reserved and shall not be transmitted. If one of the unused byte formats is received, the protocol need take no action other than counting it as another byte in cases where the number of received bytes is significant. Some of the unused bytes may be defined in the future for optional protocol expansions.

It should be remembered in looking at Table 3.1 and the following sections that although bit "b8" is shown as a "1" and should be transmitted that way it may be overwritten in the channel and be received with a value of zero.

Table 3.1 Summary of T-Link Byte Formats

Bit Values b1 MSB	b2	b3	b4	b5	b6	b7	b8 LSB	Abbr.	Meaning	See Section
0	1	0	1	0	1	1	1	Sgvi	Protocol Version Identifier Follows	3.2.1
0	0	0	0	0	1	1	1	Sgp0	Parameter 0 Follows	3.2.2
0	0	0	1	0	1	1	1	Sgp1	Parameter 1 Follows	3.2.3
0	0	1	0	0	1	1	1	Sgp2	Parameter 2 Follows	3.2.4
0	0	1	1	0	1	1	1	Sgp3	Parameter 3 Follows	3.2.5
0	1	0	0	0	1	1	1	Sgp4	Parameter 4 Follows	3.2.6
0	1	1	1	0	1	1	1	Sgss	Secondary Signaling Follows	3.2.7
0	1	1	0	0	1	1	1	Sgr	Call Restart Request	3.2.8
x	x	x	x	x	0	1	1	Sd	EIA/CCITT Control Signals and Time Fill	3.3
x	x	x	x	x	x	x	1	Ds7	56 Kbps Synchronous Data	3.4.1
x	x	x	x	x	x	0	1	Ds6	48 Kbp or Less Synchronous Data	3.4.2
x	x	x	x	0	1	0	1	DL	Low Nibble or Asynchronous Data	3.5
x	x	x	x	1	1	0	1	DH	High Nibble of Asynchronous	3.5

Note: x's for bit values indicate variable contents (0 or 1).

1.3.2. General Selector Byte Formats

The general selector byte (Sg) is normally used during call set-up, but may be used in other applications. The eight valid "Sg" messages, each pointing to a unique interpretation, are given in Table 3.1. The setting of "b6" and "b7" to "1" for a transmitted T-Link byte is required to indicate that the byte is an "Sg" message. Bit patterns for "bits" "b1" to "b5", other than those defined in Table 3.1 for "Sg" bytes, are reserved codes and shall not be transmitted. In addition reception logic shall be designed such that receipt of a reserved code will not cause any action.

1.3.2.1. Protocol Version Identifier Message

	bi							b8
Sgvi	0	1	0	1	0	1	1	1
DLvi	d3	d2	d1	d0	0	1	0	1
DNvi	d7	d6	d5	d4	1	1	0	1
	MSB							LSB

The protocol version identifier message informs the far end of the connection which version of the protocol is being used by the sender. It also serves to request the initiation of the parameter exchange sequence to begin communication end-to-end.

There shall be a minimum of sixteen (16) Sgvi messages before the arrival of the DL-DH pairs containing the Version Identifier Message information. The receiving equipment shall vote upon the repeated data using the methods described in Section 1.4.3.1.

To start a T-Link data call (after the connection is established) the called TE must send continuous Sgvi's (to the called TE). After receiving Sgvi, the called TE sends the actual protocol version identifier, encoded as two bytes, each containing one nibble (4 bits) of the byte version number. The low nibble is encoded in a DHvi byte and the high nibble in a DHvi byte, with the two bytes, then alternately transmitted three times (LD, DH, DL, DH, DL, DH) to allow transmission errors to be corrected. The receipt of this 6 byte pattern by the calling TE, causes it to stop sending Sgvi's and send its protocol version identifier to the called TE using the same encoding method.

The exchange of Sgvi bytes and protocol version identifiers between the two ends ensures that a two-way connection exists before any information is transmitted. Sgvi's from the called TE trigger Sgvi's from the calling TE which in turn triggers the transmission of the version identifiers, first by the called TE and then by the calling TE's.

To conform to standard data communications usage, the 8 bits making up the protocol version identifier are numbered "d0" through "d7", with "d0" the least significant bit and "d7" the most significant bit. The versions of T-Link that a TE is compatible with are indicated by setting or clearing the eight bits, with bit "d0" representing Version 1, bit "d1" representing Version 2 and so on up to bit "d7" which represents Version 8. If the TE supports a given version it will set the associated bit to "1" otherwise the bit will be zero. A TE can indicate support for multiple versions by setting several bits.

At present only one version of T-Link is in use (Issue 1) and so bit "d0" is set and bits "d1" to "d7" cleared. If during parameter exchange a TE receives a version identifier with which it is incompatible, it can signal this fact to the far end TE by sending an all zero version identifier instead of its actual value. This ensures that both ends recognize that a version mismatch exists.

Example: The version identifier data bytes for a TE that support T-Link Versions 1, 2 and 4 would be:

	bi							b8
DLvi	1	0	1	1	0	1	0	1
DHvi	0	0	0	0	1	1	0	1
	MSB							LSB

This is a hypothetical example since only Version 1 has been defined at present.

1.3.2.2. Parameter 0 Message

	bi							b8
Sgp0	0	0	0	0	0	1	1	1
DLp0	d3	d2	d1	d0	0	1	0	1
DNp0	d7	d6	d5	d4	1	1	0	1
	MSB							LSB

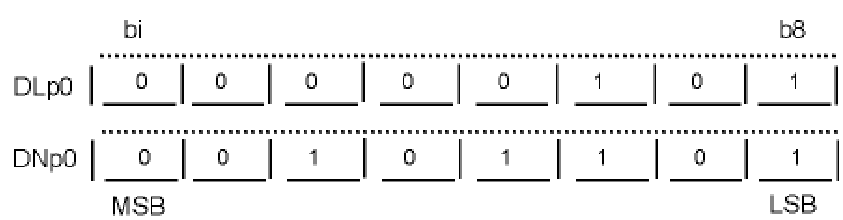
Following the completion of the version identifier message the mandatory parameters must be sent in order, possibly with optional parameters interspersed. The Sgp0 byte signals the start of the first mandatory parameter and must be sent a minimum of sixteen times, to notify the far end that the parameter is coming, followed by the eight bit parameter 0 value. It is recommended that Sgp0 (and all the other parameter identifiers) actually be sent 32 or more times to allow for errors and byte slips in the channel which may corrupt some of the bytes. Maintaining the integrity of the parameter identifiers is vitally important to the exchange of parameters and well worth the minimal delay introduced by sending the Sg bytes a few more times.

Parameter 0 data is encoded as two nibbles with the least significant 4 bits in byte DLp0 and the most significant 4 bits in DHp0. The two bytes are alternately transmitted three times to allow error correction (DL, DH, DL, DH, DL DH).

The parameter data bits have the following meanings:

d0 to d4	Reserved: transmitted as zeros and ignored on receipt.
d5	indicates how the TE is configured to interface to the user device. 0: the sending TE is connected to a DTE (e.g. terminal) and acts as a DCE. 1: the sending TE is connected to a DCE (e.g. a modem) and acts as a DTE.
d6	Reserved: transmitted as zero and ignored on receipt
d7	indicates whether the user device at the sending end is synchronous or asynchronous. 0: asynchronous transmission 1: synchronous transmission

Example: an asynchronous DTE (d5=1, d7=0) will transmit the following data bytes:



The receiving terminal equipment shall vote upon the repeated data using methods described in Section 1.4.3.1

1.3.2.3. Parameter 1 Message

	bi							b8
Sgpl	0	0	0	1	0	1	1	1
DLp1	d3	d2	d1	d0	0	1	0	1
DNpl	d7	d6	d5	d4	1	1	0	1
	MSB							LSB

Parameter 1 must follow the parameter 0 message, possibly after an optical parameter. Again the Sgpl byte at least 16 times (32 or more recommended) to prepare the far end followed by the eight bit parameter encoded with its high and low nibbles in DHpl and DLpl. To allow error correction the two encoded nibbles are alternately sent three times (DL, DH, DL, DH, DL, DH).

The data bits are interpreted differently depending on whether the sender is asynchronous or synchronous (d7 in parameter 0 is “0” or “1” respectively). If the sender is asynchronous the bits in parameter 1 are used as follows:

d0 to d3	Reserved: transmitted as zeros and ignored on receipt
d4	indicates whether the TE receiving this parameter is to generate a parity bit, based on the values of the received user data, and then add this parity to the data, before sending it to the user device. Both TE’s in a circuit must make the same choice to either enable or disable parity.
0:	parity generation disabled - the TE receiving this parameter should not generate a parity bit. The received data may or may not carry a parity bit (as data). For example, if the user data is seven bits plus parity, it could be treated as 8 bit characters with parity disabled (d4 - 0, d7 = 1) in which case the parity bit generated by one user device will be transported transparently to the other user device. In this case, it is the responsibility of the user to ensure the devices, at both ends, have compatible parity selections.
1:	parity enable - the TE receiving this parameter should generate a parity bit from the received user data. In this case, the received data will not carry an embedded parity bit. For example, if the data is seven bits plus parity, the parity bit will be stripped off at the sending end to form a seven bit character (d7=0), regenerated at the receiver and then added to the data to form an 8 bit character that is sent to the user device. The actual type of parity calculated (bit d5) may differ at the two TE’s.

d5	indicates the type of parity. If parity is disabled this bit is transmitted as zero and ignored on receipt. How this bit is used in implementation dependent and some TE's may allow different parity selections at the two ends of a connection. 0: even parity 1: odd parity
d6	indicates the mode of transmission (from the DTE) 0: half duplex 1: 8 bit words

NOTE: The actual connection between the TE's remains full duplex in either case.

d7	indicates the length of a data word. This choice may be subsequently overridden by bits "d6" and "d7" of parameter 3, if operation with 5 or 6 bit data words is desired. This length excludes the parity bit, if parity is enabled, since the parity bit is not transported across the circuit. 0: 7 bit words 1: 8 bit words

For a synchronous sender parameter 1 is interpreted as:

d0 to d5	Reserved; transmitted as zeros and ignored on receipt
d6	indicates the mode of transmission 0: half duplex 1: full duplex
d7	indicates the transmit data clocking source (from the user to the TE). Appendix B contains further discussion of timing considerations for synchronous data. 0: the TE intends to accept a user-provided external clock to time the transmitted data. Data flow will not be synchronized to the network in this case. 1: the TE intends to use the receiver clock to also time the transmitted data. This will synchronize the data flow to the network clock (64 kbps).

Example: An asynchronous terminal operating half duplex with odd parity and 7 bit words (d4=1, d5=1, d6=0, d7=0) would have the following parameter data sent:

	b1							b8
DLp1	0	0	0	0	0	1	0	1
DNp1	0	0	1	1	1	1	0	1
	MSB							LSB

1.3.2.4. Parameter 2 Message

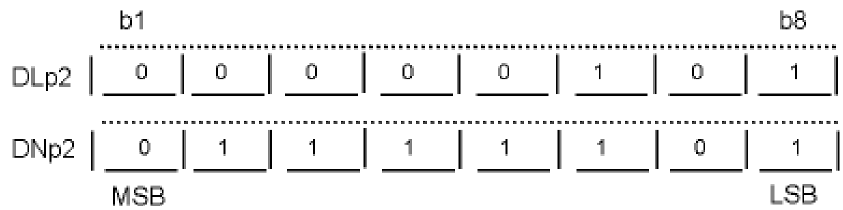
	b1							b8
Sgp2	0	0	1	0	0	1	1	1
DLp2	d3	d2	d1	d0	0	1	0	1
DNp2	d7	d6	d5	d4	1	1	0	1
	MSB							LSB

The parameter 2 message follows parameter 1 (an optional parameter may be inserted between them) and as stated previously, it starts with at least 16 transmissions of the Sgp2 byte (32 or more repetitions are suggested). This is followed by three transmissions of DLp2 and DHp2, the encoded low and high nibbles, respectively, of the 8 bit parameter 2 value (DL, DH, DL, DH, DL, DH).

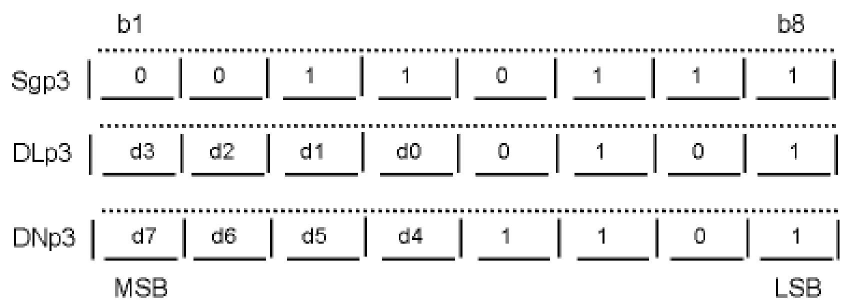
With the exception of bit 4 (d4), the bits of parameter 2 have the same meaning for synchronous and asynchronous transmission:

d0 to d3	Reserved; transmitted as zeros and ignored on receipt
d4	for asynchronous operation (d7 of parameter 0 is "0") this indicates the number of stop bits. This choice can be overridden by bit "d4" of parameter 3 to obtain 1.5 stop bits. 0: 1 bit stop time 1: 2 bit stop time - for synchronous operation (d7 of parameter 0 is "1") this bit is set to zero on transmission and should be ignored on receipt.
d5	indicates whether data being passed to the sending station will be echoed back by an intermediate device (generally the TE) as well as being sent to the far end TE. This is useful when data transmission is half duplex. 0: data will be echoed back by an intermediate device at the end sending this parameter, generally by the TE. 1: data will NOT be echoed by an intermediate device at the end sending this parameter
d6	indicates if the sender is setup to auto-answer a modem if this is the terminating device. This information is not used by T-Link but is passed to the user device in case it is required in a specific application. Note that this is an issue between the TE and its associated DCE and does not affect the answering of a T-Link call since the call must be answered before the parameter exchange can begin. If the TE is not connected to a modem (DCE) this bit should be set to zero. 0: manual assistance will be required to answer a call to a modem at the sender's end 1: the modem at the sender's end can auto-answer a modem.
d7	is used to start and stop a loopback at the receiving end. During a loopback all bytes (data and signaling) received by the looped back end will be sent back to the other end. The loopback, if requested, takes effect at the end of the call setup, that is after all parameters have been received and accepted. To clear a loopback, it is necessary to set up a new call with this bit turned off during the parameter exchange. If the TE is requesting a loopback in its transmitted parameter 2, it must ignore received loopback requests to avoid a lockup state with both TE's looped. 0: turn loopback OFF 1: turn loopback ON

Example: for an asynchronous modem i.e. in parameter 0 d5 = 0 and d7 = 0) using 2 stop bits, with no echo. capable of auto-answering and without a loopback at the far end (d4=1, d5=1, d6=1, d7=0) parameter 2 would be sent as:



1.3.2.5. Parameter 3 Message



The parameter 3 message comes after the completion of parameter message 2 with an optional parameter message permitted between them. It consists of at least 16 transmissions of Sgp3 (more are preferred) followed by three alternate repetitions of the parameter data bytes DLp3 and DHp3 containing the low and high nibbles of the parameter 3 value. The data is sent in the pattern DL, DH, DL, DH, DL, DH.

If the sending end operates synchronously (bit d7 of parameter 0 is “1”) then all the bits of parameter 3 are reserved and should be set to zero on transmission and ignored on reception. The bits must still be sent however to maintain a universal core for the parameter exchange.

For asynchronous operation at the sending end (bit d7 of parameter 0 equal to “0”) the bits are used as follows:

d0 to d3	Reserved; set to zero on transmission and ignored when received
d4	allows the use of 1.5 stop bits for asynchronous data to and from the device terminating the sending end. 0: use the number of stop bits specified by bit d4 of parameter 2 1: use 1.5 stop bits
d5	Reserved: send as zero, ignore no receipt
d6, d7	allows shorter data word lengths than defined by bit d7 of parameter 1. As before, if parity generation is enabled (d4 = 1 in parameter 1) the length excludes parity bits.

d7	d6	
0	0	- use word length in parameter 1 (7 to 8 bits).
0	1	- use 6 bit data words,
1	0	- use 5 bit data words,
1	1	- illegal, should never be transmitted and only be received due to errors on the end-to-end connection.

Example: for asynchronous operation with 1-5 stop bits and 6 bit data words (d4=1, d6=1, d7=0) the parameter 3 bytes are:

	b1							b8
DLp3	0	0	0	0	0	1	0	1
DNp3	0	1	0	1	1	1	0	1
	MSB							LSB

In most cases the parameter 3 bits will be all zero since 1.5 stop bits and 5 or 6 bits are not frequently used.

1.3.2.6. Parameter 4 Message

	b1							b8
Sgp4	0	1	0	0	0	1	1	1
DLp4	d3	d2	d1	d0	0	1	0	1
DNp4	d7	d6	d5	d4	1	1	0	1
	MSB							LSB

The final parameter message is for parameter 4 which follows the parameter 3 message. This parameter message may be preceded by an optional parameter message but not followed by one. The parameter 4 message consists of at least 16 transmissions of Sgp4 (32 transmissions or more are suggested) followed by 3 repetitions of the two data bytes each containing a nibble of the parameter 4 value, the low 4 bits are in DLp4 and the high 4 bits in DHp4. The data bytes are sent in the format DL, DH, DL, DH, DL, DH to allow error correction.

The parameter 4 value indicates the data rate requested by the sender for use between itself and the user's DCE or DTE. The speed selection depends on the values of d4 to d7 (d0 to d3 are reserved and must be set to zero on transmission and ignored on receipt) and whether the sender wants asynchronous (d7 of parameter 0 is a "1") operation. Table 3.2 lists the options of synchronous and asynchronous data rates.

Example: setting d4=0, d5=1, d6=1, and d7=1 selects 7200 bps asynchronous or 48000 bps synchronous operation. The corresponding data bytes are:

	b1							b8
DLp4	0	0	0	0	0	1	0	1
DNp4	1	1	1	0	1	1	0	1
	MSB							LSB

d7	d6	d5	d4	Async. Data Rate (bits/sec.)	Sync. Data Rate (bits/sec.)
0	0	0	0	19200	Reserved
0	0	0	1	50	Reserved
0	0	1	0	75	Reserved
0	0	1	1	110	Reserved
0	1	0	0	134.5	1200
0	1	0	1	150	2400
0	1	1	0	300	3600
0	1	1	1	600	4800
1	0	0	0	1200	7200
1	0	0	1	1800	9600
1	0	1	0	2000	14400
1	0	1	1	2400	19200
1	1	0	0	3600	38400
1	1	0	1	4800	40800
1	1	1	0	7200	48000
1	1	1	1	9600	56000

1.3.2.7. Secondary Signaling Message

	b1							b8
Sgss	0	1	1	1	0	1	1	1
DLss	d3	d2	d1	d0	0	1	0	1
DNss	d7	d6	d5	d4	1	1	0	1
	MSB							LSB

This is an optional message used to indicate the levels or user interface control leads used by EIA RS-449, EIA RS-232C and CCITT V.24 but not supported by Sd (see Section 1.3.3). If used it may be inserted between any two of the six messages used during initial handshaking (version identifier or parameters 0 through 4) but not before the version identifier. Sgss messages may also be sent during data mode under certain circumstances. The secondary signaling message is sent in the same fashion as the other 3 byte signaling messages with a minimum of 16 transmissions of Sgss (more are recommended) followed by 3 repetitions of the two encoded signaling data nibbles (DL, DH, DL, DH, DL, DH). The length of the message means that it can only be sent during data mode for data rates of 1200 bps and slower since at higher rates there are not enough bytes between user data bytes. It will be necessary to repeat the parameter exchange to send an Sgss message at rates over 2400 bps.

For asynchronous transmission it may be possible to buffer the user data and send it after the secondary signaling message in which case its use with higher data rates may be possible.

The function of signaling bits d0 to d6 are shown in Table 3-3 (bit d7 is unused). Also shown is the direction of the signals, for a DTE signals labeled “in” will be sent by the TE to the user device based on the received parameter values from the far and TE while the transmitted bit values (TE to TE) will reflect the levels of the signals received on the DTE leads labeled “out”. The reverse is true for a DCE. As a result some of the bits may have no meaning on transmission since no corresponding input exists and others can be ignored on receipt since there is no output for that signal. If a secondary signaling line is not used the corresponding bits should be set in a neutral state. (Neutral is a state that will not interfere with the other signals and data and may be off for some leads (e.g. test mode) and on for others (e.g. secondary clear to send).

T-Link simply provides a means to transport the signals shown in Table 3-3 end-to-end and any actions taken on receipt of these signals is implementation dependent. Specifically, the reaction to loopback and test signals may be to simply pass the signals to the user device or the TE may act on them in some way.

Table 3-3. Use of the Bits in the Secondary Signaling Message				
SignallingBit	NAME OF THE SIGNAL	RS232C LEAD [3]	RS449 LEAD [5]	CCITTT LEAD [4]
d0	Secondary Request to Send (Out)	SCA	SRS	120
d1	Secondary Clear to Send (In)	SCB	SCS	121
d2	Local Loopback (Out)	-	LL	141
d3	Remote Loopback (Out)	-	RL	140
d4	Test Mode (In)	-	TM	142
d5	Terminal In Service (Out)	-	IS	-
d6	Secondary Receive Ready (In)	SCF	SRR	122

Notes:

1. Dashes in the columns for RS232C and CCITT (V.24) leads indicate that the function is not defined for that interface.

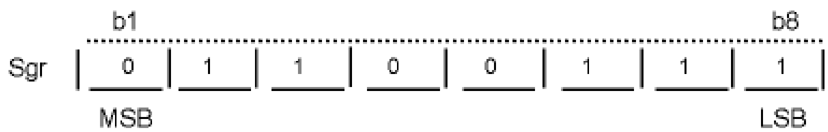
2. A value of one for signaling lead represents on the ON condition (active) and a zero represents the OFF condition (inactive).

3. Unused signals should be shown in the OFF condition (O).

4. The directions in brackets after the names of the signals indicate which way the signal flows relative to a DTE. The directions will be reversed fro a DCE.

5. Bit d7 is unused (transmitted as zero and ignored on receipt).

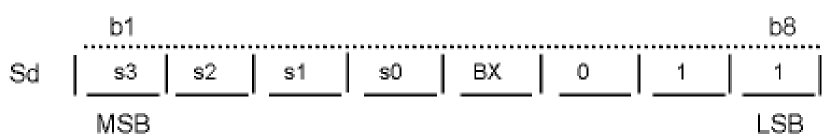
1.3.2.8. Call Restart Request



The call restart byte, Sgr, is sent by the TE when it finds itself in a condition where it is impossible to continue with the call. There are three situations in which a restart is useful: signaling a failure to adapt during profile adaption, signaling a change in the operating parameters of the TE that affects end-to-end data flow and recovery from a channel or TE failure. The effect of Sgr is to force the receiving TE to send continuous Sgvi bytes and repeat the T-Link synchronization and parameter exchange. Sgr should be sent until the TE receives back Sgvi's from the far end indicating that it has started the synchronization process. If it supports profile adaption the TE receiving Sgr will act as the adaptable party in the parameter exchange that follows.

1.3.3. EIA/CCITT Control Lead Signaling Byte

The format of the signaling byte (Sd) is shown below with b7 set = to 1 to indicate it is a signaling byte and b6 set = 0 to indicate it is an Sd type signaling byte (rather than Sg).



Unlike the signaling bytes considered up to now, control lead signaling bytes (Sd's) contain the signaling information to be transferred and are sent individually. These bytes serve two purposes, to indicate the levels on the interface control leads from end-to-end and to act as time fill between user data bytes. Time fill is needed since the full 56 kbps of channel capacity (or even the 48 kbps available if bits b7 and b8 are reserved) is not occupied with data for most speeds. Also, by varying the number of Sd's between encoded groups of data, it is possible to adjust for drift between the data rate and the network clock used to time the 64 mbps T-Link channel.

The Sd message supports the major control leads required by RS-232C (EIA) or V.24 (CCITT). If needed further control lead levels can be indicated using the secondary signaling message during handshaking (see Section 1.3.2.7). It is not the intent of T-Link to constrain the implementor to a specific interface type. If it is required for a specific application other interface methods may be used, providing they use the Sd byte and Sgss message in a manner consistent with the interfaces mentioned above.

The interpretation of the option bits in Sd (s0, s1, s2, s3 and BK) depend on whether the device terminating the connection is a DCE or a DTE. The function of these option bits is to take the levels on the EIA/CCITT control leads that are inputs to the TE from the user and transport these levels to the other end of the circuit where they are provided on corresponding EIA/CCITT outputs. The implications of this are:

1. if one end of the connection terminates in a DCE and one end terminates in a DTE the signaling bits will map inputs at one end to outputs on interface pins with the same name at the far end, for example connecting data set ready to data set ready and data terminal ready to data terminal ready (EIA pins CC and CD, CCITT pins 107 and 108/2).

2. if both ends terminate in a DCE or DTE the EIA/CCITT control leads monitored will be either an input or an output at both ends of the connection. In this case an input pin at one end is mapped to a different pin that is an output at the other end. For example, the data terminal ready input will control the data set ready output or vice versa with similar devices at both ends.

Table 3.4 shows hot the inputs and outputs are mapped into the option bits (s0, s1, s2, s3) and Figure 3.1 shows the virtual end-to-end connections that result. Note that Sd allows the levels on the control leads to change at any time after handshaking is completed.

Among the leads transported by the Sd byte, the carrier detect (CD), request to send (RTS), clear to send (CTS), data terminal ready (DTR) and data set ready (DSR) signals will be used by most DCE/DTE's and must be supported if RS-232C or V.24 type interfaces are used. The other signals, speed select, secondary transmit data and secondary receive data, may not be provided by all DCE/DTE's and are included to support specific early applications of T-Link. If not used these bits should be set to their off state ("0" except for s2 (secondary transmit/receive which should be set to one, if unused). If supported, the speed select lead may be controlled by a user device that is a DTE to select the data rate of a DCE at the other end of the circuit at the start of a call. The speed selected by the DTE must be specified in the T-Link parameter exchange and adopted by both TE's since they clock the data in and out. It is not possible to change speeds during a call using the speed select lead without restarting the call to reconfigure the TE's.

The fifth bit value in Sd, called BK, is used to signal the detection of a break condition (a long continuous space condition) from an asynchronous terminating device. The bit has no meaning for synchronous transmission and should be set permanently off (zero) if byte d7 of parameter 0 is a “1.” When the TE receives a break it sets the BK bit in the Sd byte to 1 and optionally forms an all zero data word, both of which are transported to the far end. As soon as the break condition is no longer being received (any non zero bit) the BK bit in Sd returns to one. when an Sd byte with BK=1 is received the TE starts transmitting all zeros (space) data to the terminating device until an Sd byte with BK=0 is received on the connection is reinitialized. It is necessary to signal break end-to-end since a string of consecutive zeros can not be represented in asynchronous data words, because each word contains at least one stop bit (a mark of “1” value). An all zero data (space) word may be sent, if necessary, when a break is detected for simplicity since the receiver may interpret the first few bits of a break as all zero data and only signal a break when the received signal remains low for several (typically two) word times. Thus, by the time the break is detected an all-zero word may be in the process of being sent. No data bytes need to be sent during a break condition and continuous Sd bytes may be transmitted. However, some implementations may send all-zero data and BK during a break. Note that a break only affects one direction of transmission and data may continue to flow from the TE receiving the break and to the TE sending it depending on the protocol in the user device.

Table 3-4. Use of the Bits in Sd Bytes

a) DTE Sending (bits sent) or DCE Receiving (bits received)

Signaling Bit	Name of Signal	EIA Lead	CCITT Lead
s0	Request to send (RTS)	CA(4)	105
s1	Speed Select (CH)	CH(23)	111
s2	Secondary Transmit Data (STD)	SBA(14)	118
s3	Data Terminal Ready (DTR)	CD(20)	108/2

b) DCE Sending (bits sent) or DTE Receiving (bits received)

Signaling Bit	Name of Signal	EIA Lead	CCITT Lead
s0	Carrier Detect (CD)	CF(8)	109
s1	Clear to Send (CTS)	CB(5)	106
s2	Secondary Receive Data (SRD)	SBB(16)	119
s3	Data Set Ready (DSR)	CC(6)	107

Notes:

EIA leads correspond to the terminology of ELA standard RS-232C. The number in brackets is the pin number on a DB25 connector.

CCITT leads correspond to the terminology of CCITT recommendation V.24.

CCITT leads where the bits are sent, the corresponding leads are inputs. Cases where bits are received treat the leads as outputs.

Carrier Detect is also referred to as Receive Line Signal Detect (RLSD).

The speed select signal (CH) is used to signal the data rate between a DTE and a DCE and does not affect the rate at which the TE's operate.

A 1 value for a signaling bit represents the ON condition (active) and a 0 represents an OFF (inactive).

Unused bits should be set to the OFF condition (0) except for s2 (STD/STR) which should be set to 1 if not used.

Figure 3-1. Examples of End-to-End Signaling Using the Bits in Sd

Consider the case of a host computer that is connected to TE that is the terminating end of a connection as illustrated below.

Originating End

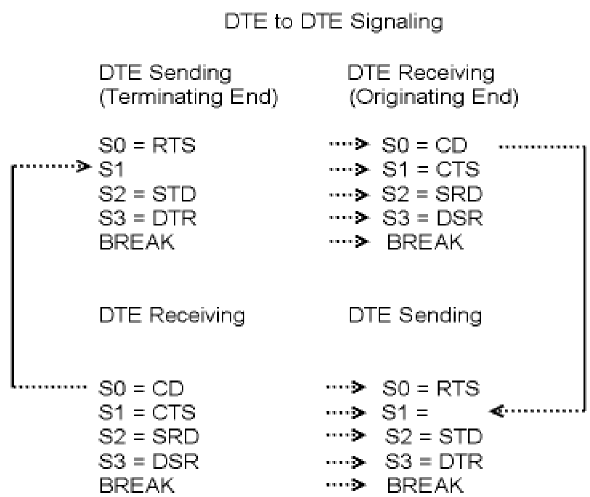
Terminating End

DTE to Host Computer

Host Computer Sending		Originating End Receiving
S0 = RTS	········>	S0 = RTS (Request to Send, EIA CA/CCITT 105
S1 = CH	········>	S1 = CH (Speed Select, EIA CH/CCITT 111)
S2 = STD	········>	S2 = STD (Secondary Transmit Data, EIA SBA/CCITT 118)
S3 = DTR	········>	S3 = DTR (Data Terminal Ready, EIA CD/EIA 108/2)
BREAK	········>	BREAK (Decoded by the TE and converted into a long spacing signal on the transmitted data lead of the DTE at the originating end.)

Host Computer Receiving		Originating End Sending
S0 = CD	<········	S0 (Carrier Detect. EIA CF/CCITT 109)
S1 = CTS	<········	S1 = CTS (Clear to Send, EIA CB/CCITT 106)
S2 = SRD	<········	S2 = SRD (Secondary Receive Data, EIA SBB/CCITT 119)
S3 = DSR	<········	S3 = DSR (Data Set Ready, EIA CC/CCITT 107)
BREAK	<········	BREAK (Detected by the TE and converted to a setting of the BK bit in the Sd byte. An all-zeros data character is also sent.)

Now consider the above case where the host computer is still the DTE at the terminating end and the originating end is connected to a terminal indicated as the DTE in the diagram above.

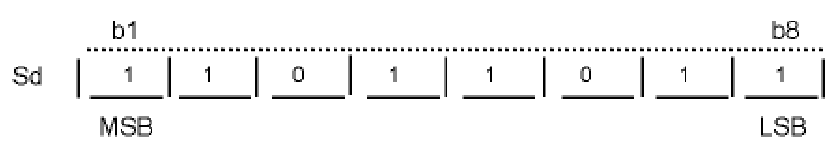


Arrows indicate the looping of a received signal to generate a transmitted signal. This should be done within the TE. In half duplex applications a delay may be inserted between transitions on the received s0 and the transmitted s1 to provide an RTS-CTS delay if needed by the DTE or DCE.

NOTE: The control lead abbreviations are expanded in Table 3.4

To reduce the risk of a bit error in a Sd byte, causing an incorrect transition of the control leads, a change in the outputs should only occur after several Sd bytes request the change.

Example: Assuming a break condition (BK=1), with active levels on the request to send, data terminal ready outputs, speed select off from a DTE terminating device (s0=1, s1=0, s3=1), and secondary transmit data inactive (s2=1), the resulting Sd byte would be:



1.3.4. Data Byte Formats

The following sections describe the byte formats used to transport data end-to-end using T-Link. It should be noted that user data is simply mapped into bytes as it is received and no effort is made to provide higher level protocol conversion. This transparency is desirable since it allows T-Link to be used as the transport mechanism for a variety of user protocols. However, it requires that the user ensure that comparable protocols are used at both ends of a connection.

There are two basic data formats, the Data Byte and the Signaling Byte. In both formats, the least significant bit is always set to a binary 1. The bit next to the least significant bit is the DATA/SIGNALING separator which identifies the two basic data formats:

xxxxxx01 = DATA BYTE (Each byte contains data information)

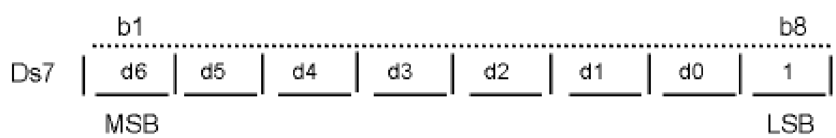
xxxxxx11 = SIGNALING BYTE (Each byte contains signaling information)

xxxxxx represents the information bits

The transmission of Synchronous User data at speeds of 56 kbits/s is a special case and is discussed in Section 3.4.1 Data is not inverted. A data MARK is a bipolar one.

1.3.4.1. 56 Kbps Synchronous Data

Synchronous data transmitted at a speed of 56 kbits/s is formatted as byte “Ds7” as shown below:



Since 56 kbps data transmission uses 7/8 of the 64 kbps transmission capacity and the least significant bit of each byte is set to 1, Ds 7 bytes contain no explicit identification. Instead, they are only sent when 56 kbps synchronous operation has been requested and accepted during handshaking and so are identified by context. The Sd messages and other signaling bytes can not be identified and are not sent during 56 kbps operation, therefore the end-to-end transfer of interface control lead information must be arranged by some other means. Further, since there is no time fill, the end-to-end data rate must be exactly 7/8 of the nominal 64 kbps channel bit rate. This implies that the user must be synchronized with the network clock.

Each Ds7 byte contains 7 bits of 56 kbps user data numbered d0 through d6. Following the convention used in data communication, d0 is the least significant data bit and it is the first bit received from and transmitted to the user interface, while d6 is the most significant bit. This is the opposite of the telecommunications convention of most significant bit (b1) first.

Once started the transmission of Ds7 will continue uninterrupted until the call is terminated by external means.

1.3.4.2. 48 Kbps or Lower Synchronous Data

Synchronous data transmitted at speeds of 48 kbits/s and below is formatted as byte “Ds6” as shown below:

	b1							b8
Ds6	d5	d4	d3	d2	d1	d0	0	1
	MSB							LSB

Synchronous transmission speeds below 48 kbps (actually 6/8 of 64 kbps) require only 6 of the 8 bits per channel byte to carry data; so one bit (b7) can be used to identify the type of byte allowing Sd bytes to be used as time fill and to indicate control lead levels. At 48 kbps all bytes are required for data but at lower speeds Ds6 and Sd bytes will be interleaved on the channel.

Each Ds6 byte contains six bits of user data numbered d0 to d5, where d0 is the least significant bit, which is transmitted/received from the user first, and d5 is the most significant bit.

1.3.5. Asynchronous Data

Asynchronous data is an 8-bit stripped of start and stop bits. This byte is separated into two 4-bit nibbles as shown:

	b1							b8
DL	d3	d2	d1	d0	0	1	0	1
DN	d7	d6	d5	d4	1	1	0	1
	MSB							LSB

Unlike synchronous data, asynchronous transmission yields inherently delimited words composed of between 5 and 8 data bits surrounded by start, stop, and optionally, parity bits. Each data work, minus start, parity and stop bits is encoded into a two byte message for transmission by T-Link with each byte containing up to a nibble (4 bits) of user data. The two bytes are referred to as DL and DH and contain, respectively, the low order and high order nibbles. The are always transmitted as a pair (DL first). The low order nibble (in DL) contains d0, the least significant bit and the one that is transmitted/received first on the user interface. The high order nibble contains the most significant bits, which may be any of d4, d5, d6 or d7, depending on the word length. Unused bit locations are set to zero on transmission.

Note that DL and DH have the same format as the data bytes sent as part of the version identifier and parameter messages. The distinction between asynchronous data, parameter data and synchronous data bytes (which could have the same values in bit b5 to b8 as DL or DH) is by context; parameter data occurs only during the parameter exchange and once the parameters are exchanged both ends know the expected data format. Sd bytes will be used to delimit DL-DH sequences, to indicate the levels on control leads, as time fill, and as a separator between data words.

1.4. Byte Sequences

This section provides details on the steps involved in transferring data using T-Link and the reasons why each step is necessary. It assumes a byte-oriented flow of data and signaling information using the terminology defined in Section 3 to name the various byte formats. The discussion follows as closely as possible the chronological sequence of events during a call from the time a circuit has been established until the exchange of data terminates. Except when noted otherwise the exchange of messages and responses to them are between the TE's not the user DTE/DCE's.

1.4.1. Synchronization and Version Verification

1.4.1.1. End-to-End Synchronization

When the circuit is established the data connection is initially assumed to be unsynchronized since neither end knows the operating parameters or even if the other TE is functional. Synchronization may also occur, or be invoked, at a later point in the data call, if one or both ends of the connection wish to change the operating parameters (i.e., removing a loopback) or to recover from a major error. To synchronize, continuous protocol version identifier bytes (Sgvi) are sent followed by encoded data (3 x DL/DH) which indicates the version of T-link used.

The end-to-end synchronization procedure is asymetric with one end originating transmission of the Sgvi bytes and the encoded data and the other end responding. At the start of a call, the called TE (answering end) is the originator and starts sending Sgvi bytes as soon as the connection is established. The calling TE (which requested the connection) may send Sgr or some other byte until it receives Sgiv's (from the called TE) which it acknowledges by sending continuous Sgvi bytes. Transmission of Sgr bytes while waiting for Sgvi is optional and other byte values may be used (Sgvi is not allowed). In applications where the TE must be able to synchronize with its own signal for maintenance purposes it is recommended that Sgr bytes be sent (see Section 1.4.4).

The receipt of Sgvi's by the called TE verifies that a two way connection exists and it then proceeds to send the encoded version identifier data (DL/DH/DL/DH/DL/DH). The calling TE in turn knows that a two way connection exists when it receives this data and will send version identifier data of its own to the called TE.

A possible criterion for determining that Sgvi's are being received, at either end, is that 14 out of the last 16 bytes received are Sgvi's. By allowing two errors in 16 received Sgvi bytes, T-link takes into account the possibility of bit errors corrupting the received signal and allows synchronization in their presence. On the other hand, requiring 14 out of 16 consecutive bytes to be Sgvi's makes it unlikely that the pattern will occur by chance during the period while the connection is being set up and may contain fragments of other calls. The sampling interval of 16 bytes arises since this is the minimum number of consecutive received Sgvi's specified in Section 3.2.1 and some implementations may send Sgvi's in bursts of 16 separated with other bytes (e.g., encoded version identifier data).

1.4.1.2. Version Identifier Byte Sequence

Once synchronization has been established (Sgvi's have been transmitted and received) the eight bit version identifier is transmitted as a six byte sequence consisting of 3 alternate repetitions of DLvi for the low nibble of the version identifier and DHvi for the high nibble (see Section 1.3.5). The format is the same as that used for asynchronous data and is encoded and decoded in the same way. The byte pair is transmitted three times to allow some bit errors to be removed by taking a majority vote among the received bytes of each type. The precise recovery technique used is implementation dependent but the receiver should detect and reject "data" recovered when an error in bit b7 of an Sgvi byte turns it into a DL byte on receipt (see Table 3.1 for a comparison of the two byte formats). Such an error could cause the receiver to attempt to recover the version identifier while Sgvi's are in fact still being received but this is fairly easy to avoid since the false DL will be followed immediately by Sgvi's rather than more data bytes.

After the version identifier is received each end compares the received version identifier to its own value and if they agree the call continues. If the version identifiers are not compatible the actions to be taken are implementation dependent but will usually involve sending an all-zero version identifier from the calling end (since it receives the version identifier from the called before sending its own) and breaking the connection by external means. The called TE may also request that the connection be broken if it receives the all-zero version identifier or it is not compatible with the version identifier received (although this should never occur in normal operation). Note that compatibility in version identifiers does not require that the calling and called

TE's use the same version identifier but only that they both support at least one common version. If they have several versions in common the numerically highest one should be used. Section 1.3.2.1 describes how the versions supported are encoded. For the current version of T-Link the version identifier is 1 (hexadecimal 01).

The minimum version identifier message that must be sent is:

```
Sgvi/Sgvi/Sgvi/.../Sgvi/DLvi/DHvi/DLvi/DHvi/DLvi/DHvi
          |<.......... 16xSgvi.........>|
```

In most cases, however, considerably more than 16 sgvi's will be sent before a response is received from the far end TE signaling, that the transmission of the data bytes can start. In implementations where the TE's may send sgvi independently it is suggested that a minimum of 128 Sgvi bytes be sent before the version identifier data (see Section 1.4.4).

1.4.1.3. Transition Into the Parameter Exchange

At the calling end, the TE may start sending parameters as soon as it has completed sending the version identifier, if the versions at the two ends are compatible. The called end must wait until it receives the version identifier from the calling TE before sending parameters in order to ensure that the two ends are compatible and also to determine the parameters to be used since this may change for different versions of T-link. either end of the connection may send a small number of Sgvi bytes after the encoded version identifier (DL/DH pairs) as time fill. Specifically, the called TE may send Sgiv's while waiting for a response to its encoded version identifier although it is preferable to send Sgp0 bytes. These Sgvi's should be ignored by the protocol.

1.4.2. Exchange of T-Link Parameters

After synchronization is established and the version identifiers have been successfully exchanged the two TE's must decide the mode in which they will exchange user data, taking into account the characteristics of the user devices (DTE or DCE). There are five mandatory parameters (parameter 0 through 4) that must be exchanged and possibly one or more optional ones (secondary signaling is the only optional parameter formally defined). After receiving the parameters a decision is made as to whether the two ends can be made compatible and the protocol either proceeds to the transfer of user data or the synchronization process is restarted. Details on the use and meaning of the six parameters can be found in Section 1.3.

1.4.2.1. Parameter Encoding

The parameters are each eight bits long and are exchanged in the form of a parameter message. Each transmitted message consists of a minimum of 16 signaling bytes identifying the parameter type (Sg, one of Sgp0, Sgp1, Sgp2, Sgp3, Sgp4 and Sgss) followed by the parameter value split into a most significant and a least significant nibble (4 bits each), encoded into a 2 byte pair and repeated 3 times. Parameter bytes are encoded and decoded in the same way as asynchronous data using DL to carry the low number of the parameter and DH to carry the low number of the parameter and DH to carry the high nibble (also known as DLp0/ DHp0,DLp1/DHp1...depending on the parameter number). A complete parameter message will contain at least 22 bytes and have the format:

```
Sg/Sg/.../Sg/DL/DH/DL/DH/DL/DH
|<·······16xSg······>|
```

Normally the message transmitted will contain more than 16 Sg bytes to reduce the risk of error (the transmission of 32 or more Sg's is suggested).

A small number of Sg bytes may be transmitted after the parameter message with the identification for the parameter just sent. The presence of these bytes is not necessary and if present they will be ignored. The 16 Sg bytes transmitted immediately preceding the encoded data must all contain the correct parameter number.

1.4.2.2. Parameter Decoding

At the receiving end it is necessary to first determine the parameter number and then decode the parameter nibble to form a byte of parameter information. The parameter number can be calculated from the received information in a number of ways, for example, looking for an Sg byte that occurs three times in a row or in 14 out of 16 received bytes or by voting among a group of received bytes. The precise method used can be chosen to suit the implementor of the system. The TE should, however, examine the bytes in small groups and update the estimate of the parameter number as more bytes are received. This is because some bytes with the previous parameter encoding may be sent after the parameter may vary (although it must always be more than 16). The parameter identifier is sent 16 or more times to allow very reliable decoding and also to allow time for the TE to adapt its operation to the received parameters as they are received rather than at the end of the parameter exchange. It is not required that adaption occur as the parameters are received, however, and the time at which this is done may vary between implementations.

Once the parameter number is assigned the parameter value is calculated by taking a majority vote over the 3 DL an DH bytes received to obtain the low and high nibbles of the parameter respectively. Again, it is important that the recovery process detect the condition where an error in bit b7 causes a parameter identifier (e.g., Sgp0) to be received as a DL byte and not falsely decode this and the following bytes as data (i.e., false start protection).

1.4.2.3. Parameter Sequencing

The parameter exchange must contain a minimum of 6 messages (including the version identifier which must always precede the parameters) transmitted in the following order:

1. version identifier
2. parameter 0
3. parameter 1
4. parameter 2
5. parameter 3
6. parameter 4

If it is used, the secondary signaling message can be inserted between any two of the six mandatory messages. Additional parameters may also be defined in the future and can be introduced in a similar manner. Parameters 0 through 4 must always be exchanged even if they are not required in a specific application, in which case they can be ignored when received. Once the complete protocol version identifier message has been received the transmission of the parameters may proceed independently of the received messages. It is not necessary for TE to receive one parameter (from the far end) before sending the next although such “handshaking” implementations are permissible providing steps are taken to avoid lockup situations if a message is lost in transmission.

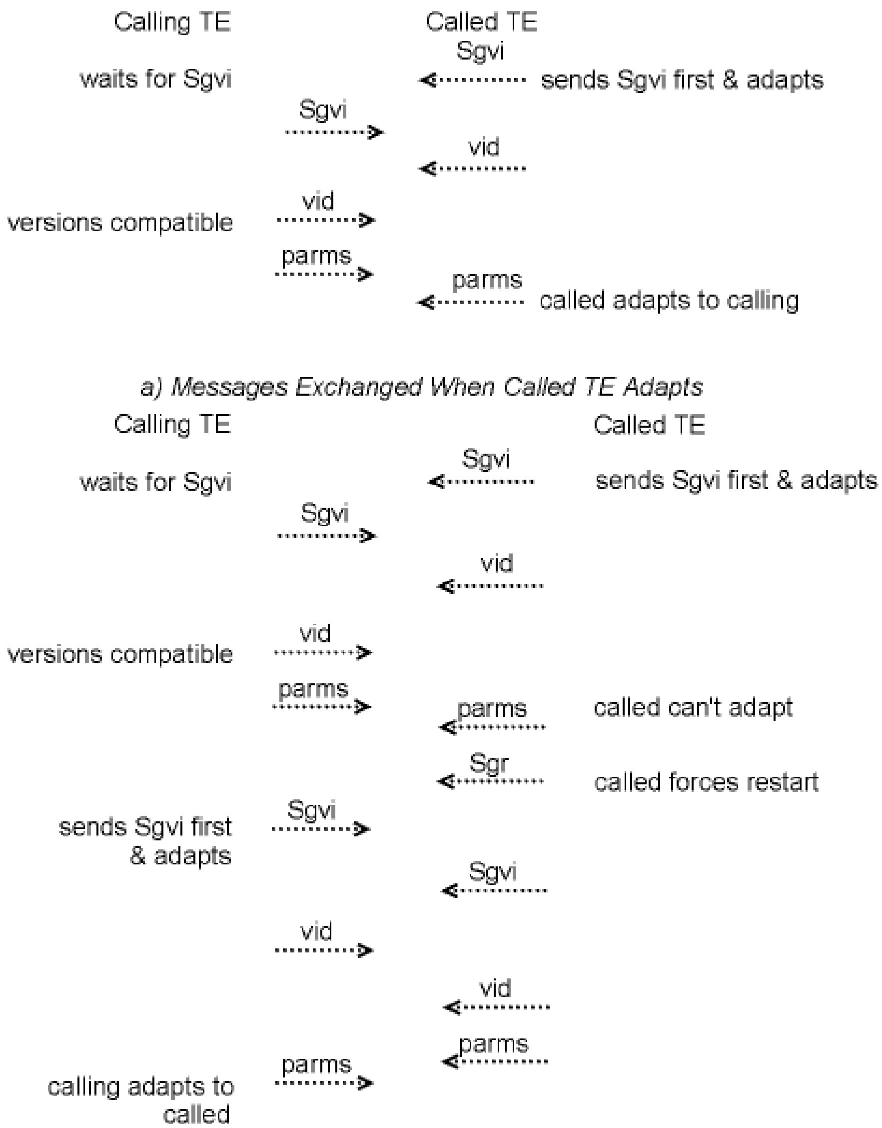
1.4.2.4. Parameter Adaptation

After both TE’s have received a copy of the parameters in use at the far end, a decision must be made on the parameters that will actually be used in data mode. When both ends have sent identical parameters the call may proceed immediately to data mode but in some instances the parameters will differ in which case one TE must adapt to match the other or the connection must be broken. The ability to adapt is optional in T-Link and may not be supported by all TE’s but is a useful feature in applications where one TE may receive calls from or make calls to other TE’s with differing operating parameters (e.g., data rate). An example of this would be a TE connected to a host computer where it is desirable for the computer port to configure itself to match the features of the computer users TE’s.

It should be noted that some parameters need not match end to end (e.g., parity type or loop-back) and the concern here is with the critical parameters such as data rate and word length.

The adaptation process proceeds as follows, assuming both TE's support adaptation. After the initial exchange of parameters each TE compares its local parameters (what it sent) to the far end parameters (what it received). If the parameters differ then the called TE should attempt to adapt to the calling TE's parameters while the calling TE can ignore the differences. If the called TE can adapt to the received parameters it will do so and the call will proceed to the data mode (see Figure 4.4a) otherwise the called party should restart the call by sending Sgr bytes. This forces the calling party to send Sgvi's beginning the synchronization and parameter exchange process again but this time with the calling TE expected to adapt (see Figure 4.4b). The calling TE in turn can adapt to the called TE and proceed to data mode, send Sgr bytes to restart the call with the called party adaptive again or terminate the connection by external means. By making the end that receives Sgr adaptable in the parameter exchange that follows it is possible to alternate the responsibility for adaptation from end-to-end until one TE adapts or the connection is broken by external actions (i.e., by one of the users or via a counter that limits the number of restarts). To allow both ends an opportunity to adapt and take into account the possibility that some restarts may be due to lost or corrupted parameters it is suggested that at least three restarts (four parameter exchanges) be allowed so each TE has as least two opportunities to adapt. Note that the TE that is to adapt should keep the control leads on the user interfaces turned off while it is adapting (see Section 1.4.2.5) so data transmission will be inhibited until adaption is complete.

Figure 4-1. Parameter Adaption at Calling and Called TE's



b) Messages Exchanged When Calling TE Adapts

vid = encoded protocol version identifier data (DL/DH/DL/DH/DL/DH)
parms = parameter message 0 through 4 plus any optional parameters

After a TE has adapted to a specific set of parameters is should send these parameters in response to future restarts until it adapts again or the choices are overridden (e.g., by the user manually changing the parameters or resetting the TE). This is necessary to avoid a possible change in the parameters if both TE's are prepared to adapt and the currently adapted TE forces a restart (due to protocol errors in data mode) forcing the far end TE to adapt in the parameter exchange that follows.

Flexibility is allowed in the adaptation procedure so that different implementations can modify it to match the specific application. For instance a sophisticated TE might modify the parameters it sends in subsequent exchanges if it can adapt to some but not all of the parameters received from the far end TE. Another useful feature would be to provide an option on the TE to made it non-adaptive, that is it will always restart the parameter exchange when it finds itself expected to adapt. (A non-adaptive TE is not the same as one that does not support adaptation at all since in the latter case the TE must terminate the connection if the received parameters do not match its local ones). Non-adaptive TE's are needed at the called end in applications where it is desirable to force the calling party to adapt, for example a central computer that places calls to a number of remote devices each operating at a different data rate. A TE may also support only a subset of the possible parameter choices, for instance only synchronous operation or only certain speeds, in which case it will appear to be adaptable in some calls and non-adaptable in others. Another option that some implementations may provide is for the TE that has just adapted to repeat the parameter exchange one last time using its adapted parameters to verify that the correct parameters have been selected.

1.4.2.5. Transition to Data Mode

Parameter 4 must always be the last parameter transmitted followed by continuous EIA/CCITT control signals (Sd) = "01000011" if secondary signaling is not used) for data rates less that 48 kbps. Neither end of the connection will transmit data under these conditions since an all off Sd will disable the user's device vial the EIA/CCITT control leads (e.g., turning off clear to sent and carrier detect to a DTE). At 48 kbps and higher data rates there is no time to send Sd (the channels carry continuous Ds6 or Ds7 bytes) and so a continuous mark condition (all "l" 's) should be inserted in the data. In this case a higher protocol level must be used to determine the true start of data by detecting the end of this marking condition. Also, during synchronization, parameter exchange, and any subsequent adaption, the TE's should leave the local interface leads that control data transmission (e.g., clear to send and carrier detect for DTE's) turned off. This allows the start of data transmission to be delayed by either TE until parameter adaption, if needed, is complete and the user equipment has been initialized and is ready to operate. It is suggested that a brief delay be inserted between the time that the control leads are asserted and the start of data transmission to allow time for the user device to recognize the transition.

If irresolvable difficulties occur, in adapting to the parameters or for other reasons, either end can exit from this state either by requesting that the connection be broken (by external means) or by restarting the call.

Figures 4.5 through 4.8 show transmitter and receiver state diagrams for both parties in a parameter exchange.

Figure 4-5. Receiver State Diagram for TE That is Not Required To Adapt

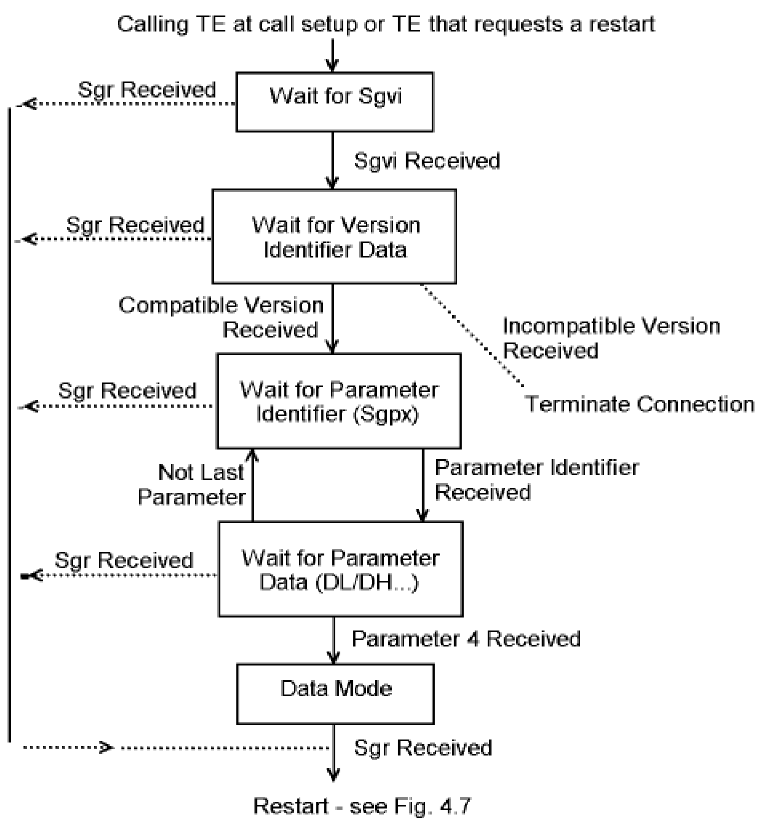


Figure 4-6. Transmitter State Diagram for TE That is Not Required To Adapt

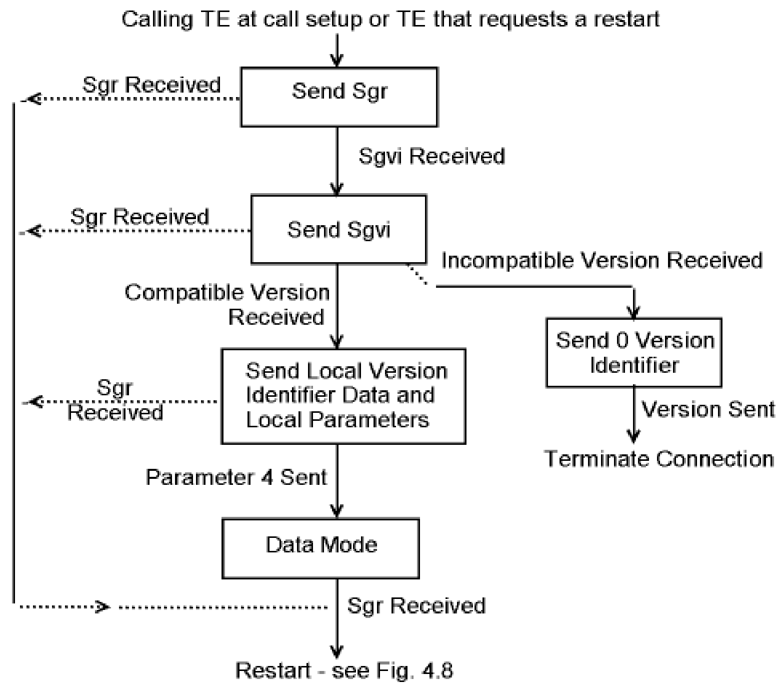


Figure 4-7. Receiver State Diagram for TE That is Required To Adapt

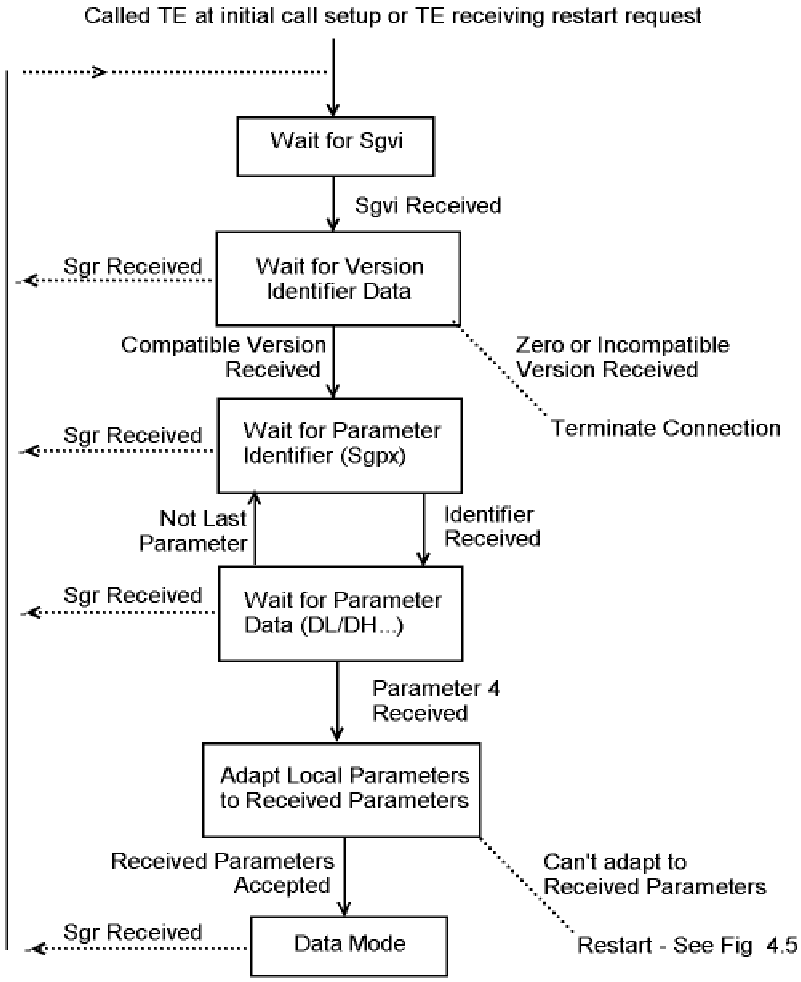
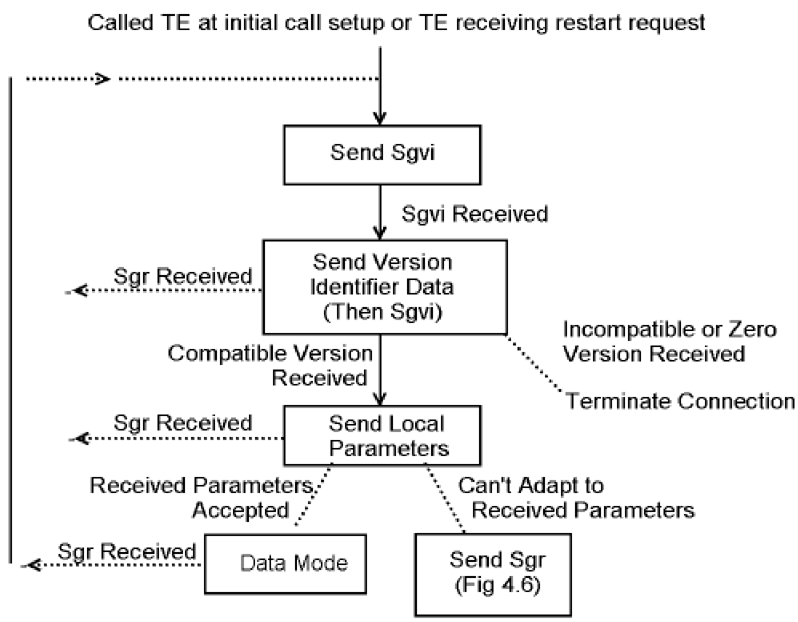


Figure 4-8. Transmitter State Diagram for TE That is Required To Adapt



Note: To allow for the special case where both ends restart simultaneously (see Section 4.4) the transmitter should send a minimum of 128 Sgvi before starting to send version identifier data.

1.4.3. Data Mode Byte Sequences

After completing the exchange and verification of parameters and waiting for any user devices that must initialize to do so the actual exchange of data can begin. The format in which the data is sent will vary depending on the speed and mode of operation, which is determined on an end-to-end basis by the parameter exchange. The following cases are considered:

- low speed asynchronous data (50 - 9600 bps)
- high speed asynchronous data (greater than 9600 bps)
- low speed synchronous data (1200 - 9600 bps)
- high speed synchronous data (14400 - 40800 bps)
- synchronous data at 48 and 56 kpbs

NOTE: T-Link is only an end-to-end protocol and if higher level protocols differ in the user devices at the two ends there will be a flow of data bits end to end but possible no transfer of information.

1.4.3.1. Low Speed Asynchronous Data (50-9600 bps)

Works of data (5 to 8 bits each) received from the user device are encoded into an asynchronous data frame 6 bytes long containing 3 repetitions of a byte pair containing the encoded low nibble (DL) and high nibble (DH) of the word (see Section 1.3.5 for further details). For all data rates up to and including 9600 bps the byte pair is transmitted three times to allow error correction by means of a majority vote at the receiver. To fill the time between user data words on the 56 kbps T-Link stream Sd signaling bytes are sent indicating the levels on the EIA-CCITT control leads. The Sd bytes are also used as delimiters to separate adjacent asynchronous data frames and there will be at least two Sd bytes transmitted between every group of repeated DL/DH bytes for the supported rates. This provides an easy method of detecting when a bit error has caused an Sd byte to be interpreted as DL or DH, incorrectly starting by decoding of a data word. The receive algorithm for signaling bytes should require that a change in an Sd byte be received several times before it affects the EIA/CCITT interface leads to protect against spurious transitions caused by bit errors.

An asynchronous data from (corresponding to one user word) on the 56 kbps byte stream would look like: ...Sd/Sd/DL/DH/DL/DH/DL/DH/Sd/Sd...

The device receiving low speed asynchronous data can decode the groups of DL/DH bytes by first performing a two out of three majority vote for the data nibbles from both the bytes in the pair. If no majority value occurs the last value received (or some other arbitrary choice) could be used. The two resulting nibbles are combined to yield the data to be sent to the user.

Byte repetition allows single bit errors in the received T-Link bytes to be detected and removed so the end to end bit error ration will be a function of the square of the channel (64 kbps) error ratio.

1.4.3.2. High Speed Asynchronous Data (over 9600 bps)

For asynchronous speeds greater than 9600 bps data works are encoded in the same fashion as for low speed data but using frames only 2 bytes long consisting of a low nibble byte (DL) and a high nibble byte (DH) (see Section 1.3.5). The resulting byte pair is transmitted over the 64 kbps channel with Sd bytes used for time fill between user data words and to indicate the levels on the EIA/CCITT interface leads. The only asynchronous data rate greater than 9600 bps currently supported is 19200 bps for which there will be four or five (in most cases four) byte times on the 64 kbps channel per data word. This would not be sufficient to implement byte repetition while leaving time for Sd's to mark the frame boundaries so a 2 byte data frame is used followed by at least 2 Sd bytes.

A frame of user data on the 64 kbps byte stream would look like:

...Sd/Sd/DL/DH/Sd...

The received byte pair is covered back to a data word at the receiver by combining the nibbles from DL to DH. No error correction takes place.

1.4.3.3. Low Speed Synchronous Data (1200-9600 bps)

For synchronous data there are no obvious word boundaries in the data received from the user so it is arbitrarily divided into 6 bit pieces each of which is encoded into a synchronous data byte (Ds6) (see Section 1.3.4 for a description or the byte format). For all data rates up to and including 9600 kbps the byte is repeated four times with Sd used to fill the 64 kbps channel until the next six bits of encoded data are ready and to signal the levels on the EIA/CCITT leads end-to-end. Unlike asynchronous data synchronous data is always being received from the user and the number of Sd bytes between encoded data bytes is almost constant for a given data rate. Variations in the number of Sd's occur for user clocked data and when a non-integral number of fill bytes are required. Samples of encoded data on the 64 kbps channel are shown below:

2400 bps

Ds6/Ds6/Ds6/Ds6/Sd/Sd/Sd/Sd/Sd/Sd/Sd/Sd/Sd/Sd/Sd/Sd/Sd/Sd/Sd/Sd/...

4800 bps

Ds6/Ds6/Ds6/Ds6/Ds6/Sd/Sd/Sd/Sd/Sd/Sd/Ds6/Ds6/Ds6/Sd/Sd/Sd/Sd/Sd/Sd...

9600 bps

Ds6/Ds6/Ds6/Ds6/Ds6/Sd/Ds6/Ds6/Ds6/Ds6/Sd/Ds6/Ds6/Ds6/Sd...

The TE receiving low speed synchronous data can compare the four Ds6 bytes in each group to each other and use the first two bytes that match to yield the data. If no two bytes agree the data may be chosen arbitrarily (e.g., the last byte in the group).

Byte repetition allows single bit errors in the received data to be corrected so the bit error ratio of the user's data will be a function of the 64 kbps channel error rate squared.

1.4.3.4. High Speed Synchronous Data (1440-40800 bps)

At synchronous data rates greater than 9600 bps (the next supported rate is 14400 bps) and less than 48000 bps the user data is encoded in Ds6 bytes in the same fashion as low speed data, however there is not enough time to send the encoded data 4 times. Therefore, the received Ds6 bytes are only sent one with Sd bytes used for time fill and to indicate the levels on the EIACCITT interface leads.

As data rate increases the number of Sd bytes between data bytes decreases until at 48 kbps there is no time to send anything but continuous Ds6's. The number of Sd's between Ds6's are not fixed for any of the data rates. The following is an example of high speed synchronous data as sent on the 64 kbps channel (19200 bps data rate):

Ds6/Sd/Ds6/Sd/Sd/Ds6/Sd/Ds6/Sd/Sd/...

The data bytes are covered directly back to user data at the receiver. There is no error correction.

1.4.3.5. Synchronous Data at 48 Kbps, 56 Kbps and 64 Kbps

At 48 kbps and higher there is no time to transmit signaling information, so Sd is undefined for these rates and all bytes are interpreted as data. Continuous data are sent on the channel using six data bits per byte (Ds6) at 48 kbps and several bits per byte (Ds7) at 56 kbps.

These data rates are derived directly from the 64 kbps channel rate since there is no possibility to use time fill bytes to adjust the rate.

1.4.4. Call Restart

At any time during a call (during the parameter exchange, while adapting or in data mode) the TE at one end of a connection may terminate communication and restart the call by sending continuous Sgr bytes. These must be detected at the far end at any time during the handshake or while in data mode. Detection of the continuous Sgr's can be done by scanning the received signaling bytes for data rates below 48 kbps and all received bytes at 48 kbps and counting how many are Sgr bytes. Every byte should be tested for Sgr's if the TE has the processing power to do so but often this will not be possible in which case periodic sampling is permissible. If all samples are Sgr's for an extended period (a few hundred milliseconds and at least a hundred samples) the receiving TE should resynchronize and start the parameter exchange following the procedures described in Sections 1.4.1 and 1.4.2. It is not necessary to check byte in the sampling process since the aim is to ensure that Sgr's are received over a long period and so sampling can be done as low priority task in the TE reduce the processing requirements. Note that call adaption restart is an optional feature and may not be supported by all implementations.

The capability to restart the call at any time from either and simplifies recovery from serious error or failure conditions by ensuring that both ends can be restored to a known state. As well, it allows the user to change the mode of operation, for example the data rate, after a call has started. The detection method is very rugged and reliable since the probability of T-Link bytes with errors simulating Sgr in a large number of samples is very small.

Call restart is also important in parameter adaption where it is used by one TE to signal that it can not adapt to the parameters it has received and wishes to repeat the parameter exchange with the other TE trying to adapt. The receipt of a restart request (repeated Sgr's) indicates that the TE is to start the synchronization process (send Sgvi) and be the adaptable party in the parameter exchange that follows. In the case of a restart used to clear an error condition while in data mode this may result in either TE being considered adaptive first. To avoid a possible change in parameters following a restart if both TE's are prepared to adapt, a TE that adapts should send its adapted rather than default parameters on a restart unless the user overrides the parameter choices, for instance by changing the TE's data rate.

If Sgr is received while it is also being transmitted, the TE should start sending Sgvi bytes and act as the called party in the parameter exchange that follows but restart the call if adaptation is required (even if the TE is in fact able to adapt). As a result if both TE's request a restart at about the same time neither and will adapt after the parameter exchange, since neither TE knows if the other TE is also prepared to adapt. In cases where the parameters agree, the call will proceed to data mode, otherwise both TE's will start over. To reduce the risk of a second collision and provide a bias towards adaptable TE's being made, the adaptable party in the next restart, an non-adaptable TE, could restart immediately. But only if the parameters do not match after a collision while an adaptable TE delays sending Sgr bytes for a few hundred milliseconds while it checks for a received restart request. As well to ensure that both TE's receive enough Sgvi bytes to synchronize, the TE transmitter should always send more than 16 Sgvi's (128 are suggested) before starting to send version identifier data.

In normal operation collisions between restart requests should be rare and the steps described in the preceding paragraph are needed mainly to allow a TE to call itself during maintenance testing. Implementations that do not face these requirements may use a simpler method of dealing with collisions such as letting the called party back down and send Sgvi's (and subsequently act as the adaptive party).

1.4.5. Call Termination

There is no inherent mechanism within T-Link to disconnect or request the disconnection of the 64 kbps circuit. Therefore, if required, this function must be provided by external means. If either or both ends or the connection have no further data to send they can transmit signaling bytes (Sd) indefinitely. In cases where the integrity of the end-to-end connection is in doubt either end can at any time (during parameter exchange or in data mode) return to the synchronization mode by ending continuous Sgr bytes and waiting for Sgvi's to be returned.

The TE should request call termination if there is a mismatch in received and local version identifiers (see Section 1.4.1.2) or when the user device powers down or turns on the DTR/DSR interface control lead. As well the TE may optionally break the connection on other conditions such as an inability to adapt the two TE's to each other, in response to a disconnect switch on the TE itself or the switch that set up the connection may under certain circumstances break it.

ATTACHMENT 2 - Clock Generation and Recovery

2.1. Clocking Considerations

2.1.1. Clocking Between the End-to-End Connection

For synchronous and asynchronous data transmission, the TE must either generate or obtain a data clock to time the transmission and reception of data to and from the user device (DTE/DCE).

2.1.2. Clocking at the User Terminal

In the case of asynchronous transmission the TE and the user device can both tolerate short term variation in the data rate since each data word has a start bit which provides a reference point from which the following data bits can be located. However, if the data being sent consists of strings of consecutive bytes (i.e., the stop bit of one word is followed immediately by the start bit of the next) the average bit rates must be reasonably close at both ends or data will be lost.

There are two common methods of generating a synchronous clock. The first assumes a completely synchronous system. In a synchronous system all the data clocks used in the system are derived from the same source as the clocks used in the system are derived from the same source as the clock for the 64 kbits/s channel. Synchronous transmission does not require a synchronous system but it is desirable. The timing signal from the 64 kbits/s channel is used to derive independent but synchronized clocks at both ends. this method will require that the TE provides both the transit and receive clocks to the user device and will be prone to slips if the timing signals from the 64 kbits/s channel are not synchronized at both ends.

The second method is network independent and works even if the connection is not synchronized. One end of the connection is designated as the master and derives the data rate from a local oscillator or an external network independent clock source provided by the user. The other end is the slave and synchronizes its data rate clock to the received byte stream.

2.1.3. “Nearly Synchronous” Data

“Nearly synchronous” data events occur when the clock used to generate the events is not synchronized to the clock used to sample the events. This situation occurs frequently when a synchronized data network interchanges data with computers which have their own internal clocks.

If the units external to the network (public or private) control the rate at which data arrives, it will be close but cannot be equal to the 64k bits/s; therefore, the data is “nearly synchronous.”

When a “nearly synchronous” data source is connected to a data sink, the data sources can overrun or underrun the data sink, which results in periodic errors. The number of error-free seconds depends upon the relative clock rates and the amount of buffering between the two systems. The buffering in this protocol is provided by the Sd bytes being inserted between the Ds bytes.

The number of signaling bytes separating the repeated synchronous data bytes is not a constant. The distance between the synchronous and “nearly synchronous” flow of bytes is illustrated below:

(a) Synchronous Byte Repetition

bit rate	# of signaling bytes separating the data bytes
2400	16
4800	6
9600	1

(B) Nearly-synchronous Byte Repetition

bit rate	# of signaling bytes separating the data bytes	
2400+	15 OR 16	
2400-	16 OR 17	
4800+	5 OR 6	1
4800-	6 OR 7	
9600+	0 TO 1	
9600-	1 OR 2	

Note: a + sign indicates a rate slightly higher than indicated.
a - sign indicates a rate slightly lower than indicated.