TELOCATOR NETWORK

PAGING PROTOCOL

(TNPP)

Version 3.8

September 8, 1997

TABLE OF CONTENTS

1.0 Introduction 13
2.0 Description 13
2.1 Data Flow 14
2.2 Block Types 14
2.2.1 Control Blocks 15
2.2.2 Paging Blocks 15
2.2.3 Miscellaneous Blocks 15
3.0 Error Control 16
3.1 Error Detection 16
3.2 Node-to-Node Error Control 17
3.2.1 Link Test 18
3.2.2 Timer and Retry Limits Recommendations 19
3.3 End-to-End Control 19
3.3.1 End-to-End Error Control 19
3.3.2 End-to-End Error Recovery 19
3.4 Simplex/Broadcast Error Control 20
3.5 Restart Sequence 20
4.0 Packet Field Interpretation 22
4.1 Transparency Insertion 22
4.2 Packet Format 23
4.3 Header Format 24
4.4 Block Formats 24
4.4.1 ETE Request Format 25
4.4.2 ETE Response Format 25
4.4.3 CAP Page Format 25
4.4.4 ID Page Format 26
4.4.5 COMMAND Format 26
4.4.6 DATA Format 26
4.4.7 STATUS Format 27
4.4.8 Extended CAP Page Format 27
4.4.9 Cap Page Group Call Address Block Format 28
4.4.10 ID Page Group Call Address Block Format 30
4.4.11 Group Call Data Block Format 31
4.4.12 Congestion Control Block Format 31
5.0 Data Field Interpretation 32
5.1 Common Fields 32
5.1.1 Character Set 32
5.1.2 Control Flags 32
5.1.3 Block Type Flag 32
5.2 Header Data Fields 33
5.2.1 Destination Address 33
5.2.2 Inertia 33
5.2.3 Source Address 33

5.2.4 Packet Serial Number....34
5.2.5 Header Extension....34
5.3 ETE Request Fields....35
5.3.1 ETE Identifier....35
5.3.2 Any Other Data Block....35
5.4 ETE Response Fields....36
5.4.1 ETE Identifier....36
5.4.2 ETE Response Code....36
5.4.3 Reject Code....37
5.5 CAP Page Fields....38
5.5.1 Page Type....38
5.5.2 Page Class....40
5.5.3 RF Channel Designator....41
5.5.4 RF Zone Designator....41
5.5.5 CAP Page Function Code....42
5.5.6 CAP Code....42
5.5.7 CAP Page Message Text....43
5.6 ID Page Fields....43
5.6.1 ID Page Function Code....43
5.6.2 ID Page Identifier....43
5.6.3 ID Page Message Text....43
5.7 COMMAND Block Fields....44
5.7.1 Manufacturer Designators....44
5.7.2 Command Code....44
5.7.3 Command Parameters....45
5.7.4 COMMAND Block Example....45
5.8 DATA Block Fields....45
5.8.1 User Defined Data....45
5.9 STATUS Block Fields....46
5.9.1 Status Code....46
5.9.2 Priority....46
5.9.3 Error Code....46
5.9.4 Date....46
5.9.5 Time....46
5.9.6 Text....47
5.10 Extended CAP Fields - Optional Block....47
5.10.1 Page Type....47
5.10.2 Page Class....47
5.10.3 RF Channel Designator....47
5.10.4 RF Zone Designator....47
5.10.5 Function Code....48
5.10.6 MSN Flag/Priority Designator....48
5.10.7 CAP Code Field....50
5.10.8 Message Sequence Number....62
5.10.9 Message Text....62
5.11 Cap Page Group Call Address Fields - Optional Block....62
5.11.1 Group Call ID....63
5.11.2 Block M....63
5.11.3 Block N....63
5.11.4 Page Type....64
5.11.5 Page Class....64

5.11.6 RF Channel Designator....64
5.11.7 RF Zone Designator....64
5.11.8 MSN Flag/Priority Designator....64
5.11.9 Options Bitmap #n....65
5.11.10 Page Type #n....65
5.11.11 Cap Code #n....65
5.11.12 Destination Address #n....66
5.11.13 RF Channel Designator #n....66
5.11.14 RF Zone Designator #n....66
5.11.15 Function Code #n....66
5.11.16 Message Sequence Number #n....67
5.11.17 Expansion Options Bitmap #n....67
5.11.18 Using Group Paging Blocks....68
5.12 ID Page Group Call Address Fields - Optional Block....68
5.12.1 Group Call ID....69
5.12.2 Block M....69
5.12.3 Block N....69
5.12.4 Options Bitmap #n....70
5.12.5 Identifier #n....70
5.12.6 Function Code #n....70
5.12.7 Expansion Options Bitmap #n....71
5.12.8 Using Group Paging Blocks....71
5.13 Group Call Data Fields - Optional Block....71
5.13.1 Group Call ID....72
5.13.2 Message Text....72
5.13.3 Using Group Paging Blocks....72
5.14 Congestion Control Block....72
5.14.1 Block Descriptions....73
5.14.2 Congestion Control Block Usage Rules....75
5.14.3 Field Descriptions....75
5.14.4 Application Notes....78
5.14.5 Glossary....79
6.0 Optional Features and Capabilities....79
6.1 Transparent CRC....79
6.1.1 Transparent CRC Data Format....81
6.2 Two-Tone Paging Formats....81
6.2.1 Two-Tone Paging Format A....81
6.2.2 Two-Tone Paging Format B....83
6.3 Voice Paging Via TNPP....84
6.3.1 Restrictions....85
6.3.2 Voice Paging Call Progress....85
6.3.3 Packet Formats....87
6.3.4 Implementation Notes....92
6.4 TNPP Dial In/Out....93
6.4.1 Dial In/Out TNPP State Tables....93
6.5 Transmitting FLEX Page Requests From Systems....95
6.5.1 FLEX Phantom - Mechanism 1....95
6.5.2 FLEX Phantom - Mechanism 2....96
6.6 Increase of Packet Size to 4096 Bytes....96
Appendix A - CRC-16 Process and Code Example....98

A.1 CRC-16 Calculation and Use 98
A.2 Sample Packet with CRC-16 BCC Included 100
Appendix B - Node to Node Communications States 101
Appendix C - Figures and Illustrations 107

LIST OF ILLUSTRATIONS

Figure 1 GENERAL DATA STRUCTURE 107

Figure 2 CAST OF CHARACTERS (ASCII CODE TABLE) 108

Figure 3 RESERVED FLAG BYTES AND THEIR USAGE 109

Figure 4 POWER-UP/RESTART SEQUENCE 110

Figure 5 T_{ict} TIME-OUT OR ENQ RECEIVED DURING PACKET TRANSMISSION 111

Figure 6 NAK RECEIVED AFTER PACKET IS SENT 112

Figure 7 NO ACKNOWLEDGMENT RECEIVED BEFORE T_{nri} EXPIRES 113

Figure 8 NO ACKNOWLEDGMENT RECEIVED BEFORE T_{nrb} EXPIRES 114

Figure 9 FLEX™-AUGMENTED CAP CODE 115

Figure 10 FLEX™-AUGMENTED CAP CODE OPTIONAL BYTES 116

UPDATE SUMMARY

Questions concerning the TNPP protocol as well as submissions for protocol updates may be directed to:

Jay Moskowitz
Chairman - TNPP Committee
Real Time Strategies, Inc.
51 East Bethpage Road
Suite 200
Plainview, NY 11803
Phone: (516) 939-6655
Fax: (516) 939-6189

Revision History:

Rev 3.8 Specification - September 8, 1997

Clarifications and Corrections:

1) Section 3.5 - Clarified Start-up sequence step 3
2) Section 4.1, 4.2, 5.5.7, 5.6.3 - Clarified Transparency insertion rules
3) Section 5.2.4 - Clarified Packet Serial Number paragraph 3
4) Section 5.5.6, 5.10.7 - Clarified Cap Code definition
5) Section 5.5.1, 5.5.6 - Correction to text to allow FLEX™ to appear in a cap page
6) Section 5.10.7, Note C - Corrected ERMES character count
7) Section 6.1 - Corrected wording (paragraph 1) and Transparent CRC (paragraph 3)
8) Minor spelling, punctuation, format, and trademark corrections.
9) April 29, 1998 corrected section numbering in Section 5.13 and 5.14.

Additions:

1) Section 4.4.9, 4.4.10, 4.4.11, 5.1.3, 5.11, 5.12, 5.13 - Group Call block types
2) Section 4.4.12, 5.1.3, 5.14 - Congestion Control blocks
3) Section 5.5.1 - APOC 1200 pager type
4) Section 5.5.1 - NewsPager 2400 pager type
5) Section 5.5.1 - SA206 pager type and note
6) Section 5.5.1, 5.5.2, 5.10.7 - FLEX™ Roaming
7) Section 5.5.1, 5.10.7 - ERMES Roaming
8) Section 5.10.7 - APOC Cap Code
9) Section 6.6 - 4K Packets
10) Appendix C - Added figures to illustrate FLEX™ -Augmented Cap Code

Rev 3.7 Specification - July 27, 1995

Update:

1) Section 4.2, 4.3, 4.4 - changed all control structure to appear in a tabular form.
2) Section 4.4.8 updated to clarify that the length of the optional Extended Cap Page Message Sequence Number field is fixed length
3) Section 5.5.1 added ERMES and APOC pager types and update Note
4) Section 5.5.2 FLEX™ Special Class 3 corrected definition
5) Section 5.5.2 updated to indicate that the Data page class is also known as transparent data and that the text field of the message will contain 8 bit binary data.
6) Section 5.7.1 updated Manufacturer list
7) Updated 5.10.6 to clarify the use of Carrier or Manufacturer specific priority definitions
8) Section 5.10.7 added ERMES Cap code field definition within Extended CAP page description
9) Updated Appendix A - A.1 - CRC-16 Calculation and Use to define the proper order of entries in this table

Rev 3.6 Specification - October 20, 1993

Update:

1) Define the Extended CAP Page format (sections 4.4.8 and 5.10).
2) Define the FLEX™ page type (section 5.5.1).
3) Break out TNPP transparency insertion requirements from section 4.0 (new section 4.1).
4) Define new page class codes (section 5.5.2).
5) Add a new block type flag (section 5.1.3).
6) Review a standardized mechanism by which FLEX™ pages may be sent across TNPP networks from nodes which do not support the FLEX™ page type (section 6.5).

Corrections:

1) Section 4.3.3 change terminology of "RF Channel Zone" to "RF Zone Designator".
2) Add missing bit pattern 0100 to the example in section 5.5.5.

Rev 3.5 Specification - March 29,1993

Update:

1) Section 3.2 - Node-to-Node Error Control- indicate inertia for that packet should be decremented.
2) New manufacturer ID code added (section 5.7.1)

Corrections:

1) Section 6.1.1 - Transparent CRC Data Format. Last sentence CRC-16 value should be hexadecimal 9A4B.

Rev 3.4 Specification - November 24, 1992

Updates:

1) Page type hex 2E (ASCII period) has been reserved (section 5.5).
2) New manufacturer ID codes have been added (section 5.7.1)
3) Update section 3.4 and the notes in Appendix B to reflect the proper processing of received sequence numbers.
4) Indicate that upper case characters A - F should be accepted as hexadecimal digits in the packet header. This is discussed in section 5.2.
5) Add section 6.4 on Dial In / Out TNPP (as submitted by Commonwealth Communications Inc.).

Corrections:

1) Correct value in column 7 of CRC-16 table. Value 4C80 was incorrectly specified as 4080.

Rev 3.3 Specification - September 20, 1991

Note: The TNPP protocol has been implemented by a large number of manufacturers and is currently in operation at thousands of sites around the world. The purpose of the 3.3 specification is to point out subtle differences which have occurred is various implementations of the protocol, because of interpretation differences between manufacturers. In addition, this revision will discuss some special features which have been implemented by certain manufacturers for use within their own network of nodes operating via the TNPP protocol. These "optional features and capabilities" are NOT part of the "official" TNPP specification and do not need to be implemented in a "standard" TNPP implementation. They are presented as part of the specification so that manufacturers are aware of the manner in which other firms have implemented certain capabilities. The original TNPP specification never detailed the implementation of the Two-Tone paging format. This update also reviews two distinct implementations which are currently in use in the field.

Updates:

1) Convert TNPP documentation from Word Perfect on an IBM-PC to Microsoft Word on a Macintosh. Convert the type font to Palatino.

2) Clarify the use of the STX in packet zero (section 3.5).

3) Clarify the use of ETX / ETB within packets (section 4.3).

4) Specify additional pager type (pager encoding format) codes (section 5.5.1)

5) Inclusion of section 6.0, the "Optional Features and Capabilities" section. This section reviews the following:

- Transparent CRC
- Two-Tone Paging Formats
- Dial In and Dial Out TNPP implementations
- Voice Paging Via TNPP (note in section 5.5.2 updated refer to this section)

Rev 3.2 Specification - January 10, 1989

Corrections:

1) The stated value for the End-to-End request and response block types were inconsistent in sections 4.3.1, 4.3.2 and 5.1.3. This has been corrected and the correct values are: ETE request as hex 3E and ETE response as hex 3C.

2) Correct various typos in the document.

3) Update State 1 (Await ENQ RESPONSE) in appendix B. There were a number of unexpected events which could occur in this state. The description stated that an ENQ should be sent in response to each of these events. Further testing has shown that the ENQ response to these events could interfere with the line resynchronization process. The updates indicate that these events should be ignored. The time-out of the T_{nre} timer will handle the necessary line resynchronization.

Updates:

1) New manufacturer ID code added to section 5.7.1.

Rev 3.1 Specification - October 10, 1988

Corrections:

1) The CRC-16 table in Appendix A contained 3 incorrect values in the prior specification. If implemented as stated in Version 3.0, a TNPP implementation would work within a network using the same incorrect values, but would not work with a network which properly calculated CRC-16 values. The 3 values and an incorrect column total has been corrected.

2) The state tables in appendix B have been corrected. If TNPP was implemented as per the Version 3.0 state tables, there was a possibility of

entering into a condition where each side of the link would go into and out of synchronization, and intersystem communications would never be re-established. The state tables in this specification have been implemented and are fully operational.

Updates:

1) New pager encoding formats have been added to section 5.5.1.

2) New manufacturer ID codes have been added to section 5.7.1.

FLEX™ and Motorola are trademarks of Motorola, Inc.

Telocator Network Paging Protocol

1.0 Introduction

This document describes a data communication protocol that is used for communications between paging terminals or other types of equipment required to implement a paging system network. The protocol is ASCII character oriented which is transmitted via a standard RS-232 port. The protocol supports variable length packets and grouping of different page and data blocks within the same data packet. The protocol uses three types of error recovery methods, two for full duplex applications and one for broadcast/simplex application.

2.0 Description

A common header is used in all of the data formats to indicate the source and destination of each data packet. The header also contains a serial number that is used for identifying packet repeats and an inertia value that represents the number of nodes the packet is allowed to pass through. The header length is variable and is delimited from the remainder of the packet with the start-of-text flag (STX). The first 12 bytes of the header are defined in the following sections. After the common header, an optional header extension can be used to extend the source and destination address fields. If a device does not use this feature, the header extension data between the first 12 bytes of the header and the STX flag should be ignored. After the STX flag the data block is sent. Multiple data blocks can be transmitted within the packet by terminating each block with an end of text block flag (ETB) and terminating the last data block with an end of text flag (ETX) followed by the block check character bytes (CRC-16). Each data block can optionally have a block modifier added. This modifier is called ETE request and is used for multiple packet messages and end-to-end control.

Standard RS-232 asynchronous data communication is used. The character format used is 8 bit, no parity, one stop bit. The data rate can vary from 300 to 9600 baud and will be determined by the application and the available data distribution network.

A packet consists of the following items:

Start-of-header flag (SOH)
Header
Start-of-text flag (STX)
Data block(s)
End flag (ETX)
Block check code (CRC-16)

2.1 Data Flow

In order to pass data through the network, the address of the final destination is assigned to the data packet by the originator. The data packet is then transmitted on the port that is connected to a path to the destination. Each node in this path must then know the correct path to the destination (routing table) so that the data packet will be routed through the network in the desired manner. A node in the network will then re-transmit the data packet without changing either the source or destination address. A new serial number is assigned to the data packet and error control is implemented each time it is transmitted from a node.

The inertia value is decremented whenever the data packet is received by a node in the network and must result in a non-zero value in order to be re-transmitted to any other node.

2.2 Block Types

The data portion of a packet may contain one or more data blocks. The format and meaning of these data blocks is determined by the first byte of each data block. Each data block is terminated with an ETB. The last data block in a packet is terminated with the end-of-text flag (ETX).

The following sections describe the block types defined as of the release date of this document. New block type definitions may be created by proposal and acceptance by the networking committee. The two control block types, end-to-end request and end-to-end response, are required in all implementations of the protocol. Other block types are application-related and are required only to the extent that the application program (paging, data transfer, etc.) must participate in information exchange.

2.2.1 Control Blocks

There are two types of blocks defined for end-to-end (ETE) control. These blocks are ETE request and ETE response.

The ETE request block is used to request receipt acknowledgment by the destination node and also to provide linking information for data blocks that span multiple packets. The ETE request may be applied as a header to any other data block type. The use of this modifier enables end-to-end acknowledgment for the data block to which it is appended.

The ETE response block is used to carry receipt acknowledgment or rejection from the destination back to the source and to inform the source if the destination can process multi-packet data blocks. If an ETE request is used to link segments of a long data block together, the first ETE request must immediately precede the first data block segment without an intervening ETB. Each successive segment of the data block must be preceded by an ETE request containing sequentially numbered segment numbers. The segments must be linked with appropriate FIRST and LAST flag settings.

An ETE response must be generated whenever an ETE request is encountered by an addressed destination. The ETE response must be addressed to the source of the ETE request and must contain the segment number of the ETE request.

2.2.2 Paging Blocks

There are two data block types that are currently defined for paging applications. CAP page block is used to send paging information which is ready for encoding. A CAP page block contains all signaling information required to generate the page.

The ID page block is used to send a request for generation of a page to be sent to a customer. An ID page block contains a customer ID number to be used in determining the signaling information for the pager. This block type requires that a data base record exist for the customer at the destination node.

2.2.3 Miscellaneous Blocks

There are three data block types currently defined for purposes other than paging or end-to-end communications. These types are: COMMAND, DATA, and STATUS.

The COMMAND block is used to send configuration and control commands to a destination. Included in the command block is a three letter manufacturer code to simplify recognition of manufacturer-specific commands.

The DATA block is used to send generic data to a destination. This document does not attempt to define the data meaning or usage. The data portion of the block is restricted to non-flag characters.

The STATUS block is used for reporting error conditions of external equipment attached to a node in the network. It is not used for reporting network conditions or for data flow related functions. Typical uses would be: transmitter failure reporting, power loss reporting, external alarm reporting, etc. The status code value assignments should be coordinated with the networking committee.

3.0 Error Control

Error detection is achieved by using a CRC-16 scheme that consists of two bytes (16 bits) of data called Block Check Code (BCC) that is transmitted after the ETX flag of each packet. The BCC is transmitted LSB first.

Node to node error control is accomplished by using a stop-and-wait automatic request for repeat (ARQ) scheme (for full/half duplex applications). End to end error recovery may be achieved, optionally, by using a ETE request to specify to the destination to respond with an ETE response block. For simplex/broadcast application a redundant transmission scheme is used.

Every packet begins with a start-of-header flag (SOH). Any time the receiver receives a SOH flag between an SOH flag and the ETX flag it should abort the packet in process and assume that a new packet has been started. Each packet is identified by a serial number field in the header block. The serial number is used by the receiver as a unique identifier for each packet. The receiver maintains a table of serial numbers of the last 64 data packets that were correctly received. When a new data packet is received without errors the serial number will be checked against this table. If a duplicate serial number is found in the table the data packet will be ignored. If not, the data packet will be processed normally and its serial number will be added to the table. If the table is full, the oldest serial number will be discarded. This scheme allows the sender to re-transmit each packet any number of times.

3.1 Error Detection

Errors in the transmitted bit stream are detected by a process called a cyclic redundancy check (CRC). See appendix A for CRC-16 process and code example. The CRC concept treats the binary string of ones and zeros contained in the packet as a single long binary number and uses that number as a dividend in a long-division problem. The divisor is called the generator polynomial and is.

$$Q(x) = x16 + x15 + x2 + 1$$

The remainder of the long division is the 16 bit BCC code which follows the ETX flag (LSB transmitted first) at the end of every packet. At the receiver, the CRC is calculated on the entire packet starting with the SOH and ending with the last byte of the transmitted CRC. If there were no errors in the packet data stream, the resulting CRC will be zero. If there were errors in the data stream, there is a very high probability that the resulting CRC will be non-zero.

3.2 Node-to-Node Error Control

Refer to section 3.2.2 for timer and counter values. See figures 3 through 8 at end of document for typical DTE to DTE communication scenarios.

Node to node error control is accomplished using a stop-and-wait automatic request for repeat (ARQ) scheme. This method is used in duplex applications where a return path is available for the receiver to respond to the sender after each data packet. The response is a single ASCII control character.

The receiver should send a positive acknowledgment flag (ACK) when a data packet has been received and the receiver generated BCC is the same as the one received in the packet. A negative acknowledgment flag (NAK) should be sent by the receiver only if the BCC check is not the same. If the sender receives a NAK flag after the packet has been sent, the sender should re-transmit the data packet up to C_{retry} times and then discard the packet. It is suggested, as a minimum, after C_{retry} an error be logged and the packet saved for later review. The inertia for that packet should be decremented.

When a receiver receives an enquiry flag (ENQ) it should respond with a end of transmission flag (EOT) to indicate that it is capable of communications. This is what is referred to as the link test (see section 3.2.1). If the sender receives a valid control flag during transmission of a packet, the sender must abandon the transmission and perform a link test. This is done to synchronize the communications path. After the link test the sender should retransmit the packet using the same serial number. Link failures can be completely recovered from with no duplicate or lost packets.

If a packet is sent, and the link has been idle for t_{nri} or the link has been busy for t_{nrb} and no acknowledgment has been received, the sender should perform a link test. After the EOT flag has been received, communication is re-established, and the sender should retransmit the packet for which there was no response. In the case where the receiver received a good packet and sent a ACK flag and a transmission error occurs in the ACK flag, the sender will time-out and perform a link test. After the EOT flag the sender will then re-transmit the packet. If the packet is received with no errors an ACK flag is sent and the serial number of the packet is checked in the received serial number table. If the number is in the table then the packet is ignored. This process will guarantee that the packet would not be processed more than once at the receiver.

A CAN flag is used to respond to a correctly received packet when the receiver cannot process the packet. When a CAN flag is received, the sender should save the packet and log a CAN error. As an option, the sender may attempt to re-route the packet over an alternate link. The serial number must be incremented for the next packet transmission.

If the receiving node cannot process a data packet at the time it is received and it is a temporary situation (due to buffer overflow, link unavailable, etc.) a hold flag (RS) will be transmitted to the sender. When an RS flag is received in response to a data packet, the sender should hold off re-transmitting that particular data packet for approximately t_{hold} seconds. During this hold time other data packets may be transmitted normally. The serial number of the packet that was RS'ed will be used for the next available packet. The original data packet is re-transmitted after t_{hold} seconds with the next available serial number. This is done to keep the serial numbers in order. A packet should be discarded after C_{hold} retries in response to a RS flag. Again, as a minimum, an RS error should be logged and the packet should be saved for later review.

3.2.1 Link Test

Whenever the link is idle for more than t_{idle} a link integrity test must be done to verify proper operation of communications link. This check is referred to as the link test. The link test consist of the sender sending an ENQ flag incrementing C_{enq} and waiting for either an EOT flag response from the receiver or t_{nre} timer to expire. If t_{nre} expires before an EOT is received, then another ENQ flag is sent and C_{enq} is incremented. This sequence is repeated until an EOT flag is received. When an EOT flag is received normal communications resumes. If C_{enq} limit is reached before an EOT flag is received then the sender should log a link failure error.

This procedure may occur during any state of operation of the network protocol.

3.2.2 Timer and Retry Limits Recommendations

The following lists the timers and retry limits:

t_{ict}	-Inter-character time within packet:	2 seconds
t_{nri}	-Time-out on response with idle receiver:	10 seconds
t_{nrb}	-Time-out on response with busy receiver:	60 seconds
t_{nre}	-Time-out on response to ENQ:	10 seconds
t_{hold}	-RS flag hold off time:	10 seconds
t_{idle}	-Time-out on idle link for keep alive ENQ:	60 seconds
C_{retry}	-Retries by sending station per packet:	6
C_{hold}	-Re-transmissions in response to RS holdbacks:	24
C_{enq}	-ENQ retries before error logging:	6

3.3 End-to-End Control

3.3.1 End-to-End Error Control

End-to-End error control is implemented by the use of the ETE request and ETE response data blocks. An ETE request block is issued in the packet by the originator when end-to-end control is required. The destination will respond back to the originator with an ETE response block which contains the segment number from the ETE request that evoked the response. A definition of the ETE request and ETE response data fields are listed in sections 5.3 and 5.4, respectively.

3.3.2 End-to-End Error Recovery

Recovery consists of performing a link test, and upon successful completion of a link check, assignment of the segment number (in the event of a two-station failure). In the event of a single-station failure, the segment number is reassigned to the value of the last correctly received block at the remote station. In the event of a link failure, no sequence information is lost, and, upon completion of a link check, transmission continues. In the event of a single node failure lost packets can be recovered but duplication of blocks may occur. Therefore it is the responsibility of the end-to-end functions to recover from duplicate packets.

The ETE request provides data fields for indicating multiple-packet messages. Multiple-packet messages are useful when the data to be sent exceeds the maximum length of a packet. If an originator wishes to indicate a sequence of more than one block, it uses the first and last block flags within the ETE request. A sequence of data blocks with the first block flag set in the first block, and the last block flag set in the ending block would be interpreted as a single data block at the destination node. The ETE response contains a data field that indicates the number of packets the originator can have outstanding to a destination before an ETE response is required. This field is referred to as the window size.

The destination has the option of accepting only single-block sequences by responding to the first block with the Multi-block OK flag set to zero. The destination would then reject any other blocks of the sequence with a reject code of 49 (no multi-block sequences allowed).

3.4 Simplex/Broadcast Error Control

The simplex/broadcast method of error recovery consists of repeating the transmission of a data packet (redundant transmission). This scheme would be used on a one-way (simplex) distribution network with no return path for the acknowledgment. The data packet serial number is assigned by the sender and is used to identify the packet for multiple transmissions (repeats). The serial number is checked by the receiver in the received serial number table and if the packet has already been processed then the packet is ignored. Therefore error correction can be achieved by repeating the packet. In a one-way distribution network, time diversity may be enhanced by grouping packets in sets and repeating the set. When grouping packets the total number of packets in the set must not exceed the size of the table in the receiver, 64.

Note: When a packet containing a correct checksum is received, the serial number of the packet should be added to the received serial number table. This should be performed regardless of the validity of the source and destination address information in the packet.

3.5 Restart Sequence

The link test consists of the exchange of the enquire flag (ENQ) and the end of transmission (EOT) flag. During startup each node must go through a link test procedure before any communications is initiated.

Startup sequence for each node/link consists of:

1) clear the receiver serial number table,
2) perform link test procedure,
3) send an empty packet (no data blocks) with the serial number and destination address set to zero. The source address in the empty packet should be the address of the transmitting system just as it appears in any packet which contains data blocks.

The empty data packet must be acknowledged (ACK) by the receiving node. The empty packet must follow the link test procedure and precede the first data transmission. The time between the link test and the empty data packet can vary. The packet with serial number zero causes the receiving node to clear its receive serial number table. Serial number zero is reserved for serial number synchronization and must not be used at any other time. Restart procedures are listed in appendix B, node-to-node communication states.

REV 3.3 NOTE: According to the specification, the empty packet consists of a packet without any data blocks. There has been some confusion regarding the format of packet zero. One interpretation does not place an STX in the packet, since there is no text (no data block) which is following the header. The alternate interpretation maintains the STX/ETX combination in the packet, without any data block between these control characters. Because many implementations already exist in the field, it is recommended that the STX in packet zero be considered as an optional character.

4.0 Packet Field Interpretation

The following section shows the interpretation and arrangement of the data fields within the packet.

The same type of header is used for all packets. The end of block flag indicates which type of data is to follow and is used to allow multiple data blocks within a packet.

4.1 Transparency Insertion

See figure 3 for a list of control characters which are used by the protocol. All ASCII character values which are not listed in figure 3 are permitted within a data block. If any control characters used by the protocol, as listed in figure 3, are to be transmitted within a block as data, the originator must perform "transparency insertion" to make these control characters transparent to the protocol. This is done by converting the control byte to two bytes consisting of a SUB (1A) character followed by the printable ASCII character formed by adding 40 hex to the byte to be sent. Note that a data byte value of 1A, the SUB character itself, is transmitted as 1A, 5A. Also note that the hex FF character appears in figure 3 and is therefore transmitted as 1A, 3F. The data receiver will remove all SUB characters from the data stream upon receipt and subtract hex 40 from the byte which follows.

REV 3.8 NOTE: Prior to revision 3.8, there was conflicting information in the specification with regard to when transparency insertion was to be performed. In addition to the methodology described in the prior paragraph, the specification could have been interpreted as requiring that all control characters have transparency inserted. Although the 3.8 specification clarifies this point, for compatibility with legacy systems it is recommended, yet not required, that a TNPP implementation provide an option to insert transparency on all control characters to be sent within a TNPP packet. If such an option does not exist to utilize a single form of transparency insertion between two systems, the TNPP links in question could be routed through a third system, which does provide such an option. In that way transparent data of one methodology can be converted into transparent data of the alternate methodology, as it passes through the third system which connects to the otherwise incompatible pair.

4.2 Packet Format

Field Name	# of bytes
SOH (start-of-header flag)	1
Header	12
STX (start-of-text flag)	1
Data block(s)(one or more)	N
ETX (end of text flag)	1
CRC	2

REV 3.3 NOTE: Refer to note regarding the format of packet zero in section 3.5.

The maximum size of a TNPP packet is normally 1024 bytes (see section 6.6). This includes all characters from the SOH at the start of each packet to the last byte of the CRC, and includes all data bytes, and transparency control characters which exists in the packet.

4.3 Header Format

Field name	# of bytes
Destination address	4
Inertia	2
Source Address	4
Data Packet Serial Number	2
Header extension (optional) (See section 5.2.5)	Reserved

NOTE: A packet must not exceed 1024 bytes including all flag bytes and CRC bytes.

4.4 Block Formats

The following sections list the data field formats and the number of bytes per field of the data blocks.

REV 3.3 NOTE - Multiple data blocks may be placed within one TNPP packet. This section explains that each data block ends with an "End-of-Block" flag. The notes at the bottom of each page in section 4.4 go on to explain that the last data block should be terminated with an ETX. These descriptions have lead to a certain degree of mis-interpretation. In one interpretation, an ETB (hexadecimal 17) is placed at the end of each block in a packet, and an ETX is placed directly after the ETB in the last block in the packet. In the other interpretation, every block except the last block in the packet ends in an ETB, but, the last block ends in an ETX rather than an ETB. Because both implementations have been installed in a large number of sites, we recommend that the combination of ETB followed by ETX, which may occur at the last block in a packet, be interpreted as if this block ended in an ETX alone. In this way, the use of an ETB will be considered optional for the last block in a packet.

4.4.1 ETE Request Format

Field Name	# of bytes
Block type (3E hex)	1
ETE Identifier	2
<any other data block>	0-N
ETB (end-of-block flag)	1

4.4.2 ETE Response Format

Field Name	# of bytes
Block type flag (3C hex)	1
ETE Identifier	2
Response code	1
Reject Code	1
ETB (end-of-block flag)	1

NOTE: The final end flag in the last data field of a packet must be an ETX to indicate the end of the packet and the CRC-16 BCC should then follow. (Refer to note in section 4.4)

4.4.3 CAP Page Format

Field Name	# of bytes
Block type flag (41 hex)	1
Page Type	1
Page Class	1
RF Channel Designator	1
RF Channel Zone	1
Function code	1
Cap Code	8
Message Text	0 - N
ETB (end-of-block flag)	1

4.4.4 ID Page Format

Field Name	# of bytes
Block type flag (42 hex)	1
Function code	1
Identifier	10
Message Text	0 - N
ETB (end-of-block flag)	1

NOTE: The final end flag in the last data field of a packet must be an ETX to indicate the end of the packet and the CRC-16 will follow. (Refer to note in section 4.4)

4.4.5 COMMAND Format

Field Name	# of bytes
Block type flag (43 hex)	1
Command set / Manufacturer code	3
Command	3
Command parameters	0 - N
ETB (end-of-block flag)	1

4.4.6 DATA Format

Field Name	# of bytes
Block type flag (44 hex)	1
User defined data	0 - N
ETB (end-of-block flag)	1

NOTE: The final end flag in the last data field of a packet must be an ETX to indicate the end of the packet and the CRC-16 will follow. (Refer to note in section 4.4)

4.4.7 STATUS Format

Field Name	# of bytes
Block type flag (45 hex)	1
Status code	1
Priority	1
Error code	4
Date	6
Time	4
Text (optional)	0 - N
ETB (end-of-block flag)	1

NOTE: The final end flag in the last data field of a packet must be an ETX to indicate the end of the packet and the CRC-16 will follow. (Refer to note in section 4.4)

4.4.8 Extended CAP Page Format (Optional - Refer to section 5.10)

Field Name	# of bytes	Comments
Block type flag (46 hex)	1	
Page Type	1	
Page Class	1	
RF Channel Designator	1	
RF Zone Designator	1	
Function Code	1	
MSN Flag / Priority Designator	1	
CAP Code	Pager Type Specific	
Message Sequence Number	2	Optional field included only if "MSN flag" is set
Message Text	0 - N	
ETB (end-of-block flag)	1	

NOTE: The final end flag in the last data field of a packet must be an ETX to indicate the end of the packet and the CRC-16 will follow. (Refer to note in section 4.4)

4.4.9 Cap Page Group Call Address Block Format (Optional - Refer to section 5.11)

Field Name	Number of Bytes	Comments
Block type flag (47 hex)	1	
Group Call ID	1	Format: 01ABCDEF, where A through F give the temporary group call ID number associated with this group call - max of 64.
Block M	1	Format: 01ABCDEF, where A through F give the current block number associated with this group call's address list - max of 64.
Block N	1	Format: 01ABCDEF, where A through F give the total number of blocks associated with this group call's address list - max of 64.
Page Type	1	
Page Class	1	
RF Channel Designator	1	
RF Zone Designator	1	
MSN Flag / Priority Designator	1	
Options Bitmap #1	1	Options Bitmap #1 - Format: 01ABCDEF, where A through F are individual flags indicating which cap code-related optional fields are present in relation to Cap Code #1.
Page Type #1	1	Page Type #1 - optional, present if specified in Options Bitmap #1 - bit A.
Cap Code #1	Pager Type Specific	Cap Code #1
Destination Address #1	4	Destination Address #1 - optional, present if specified in Options Bitmap #1 - bit B.
RF Channel Designator #1	1	RF Channel Designator #1 - optional, present if specified in Options Bitmap #1 - bit C.
RF Zone Designator #1	1	RF Zone Designator #1 - optional, present if specified in Options Bitmap #1 - bit C.
Function Code #1	1	Function Code #1 - optional, present if specified in Options Bitmap #1 - bit D.
Message Sequence Number #1	2	Message Sequence Number #1 - optional, present if specified in Options Bitmap #1 - bit E. Format: 2 ASCII hexadecimal digits.

Expansion Options Bitmap #1	1	Expansion Options Bitmap #1 - optional, present if specified in Options Bitmap #1 - bit F. Format: 01ABCDEF, where A through F are individual flags indicating which cap code-related optional expansion fields are present in relation to Cap Code #1.
	• • •	
Options Bitmap #n	1	Options Bitmap #n - Format: 01ABCDEF, where A through F are individual flags indicating which cap code-related optional fields are present in relation to Cap Code #n.
Page Type #n	1	Page Type #n - optional, present if specified in Options Bitmap #n - bit A.
Cap Code #n	Pager Type Specific	Cap Code #n
Destination Address #n	4	Destination Address #n - optional, present if specified in Options Bitmap #n - bit B.
RF Channel Designator #n	1	RF Channel Designator #n - optional, present if specified in Options Bitmap #n - bit C.
RF Zone Designator #n	1	RF Zone Designator #n - optional, present if specified in Options Bitmap #n - bit C.
Function Code #n	1	Function Code #n - optional, present if specified in Options Bitmap #n - bit D.
Message Sequence Number #n	2	Message Sequence Number #n - optional, present if specified in Options Bitmap #n - bit E. Format: 2 ASCII hexadecimal digits.
Expansion Options Bitmap #n	1	Expansion Options Bitmap #n - optional, present if specified in Options Bitmap #n - bit F. Format: 01ABCDEF, where A through F are individual flags indicating which cap code-related optional expansion fields are present in relation to Cap Code #n.
ETB (end-of-block flag)	1	

NOTE: The final end flag in the last data field of a packet must be an ETX to indicate the end of the packet and the CRC-16 will follow. (Refer to note in section 4.4)

4.4.10 ID Page Group Call Address Block Format (Optional - Refer to section 5.12)

Field Name	Number of Bytes	Comments
Block type flag (48 hex)	1	
Group Call ID	1	Format: 01ABCDEF, where A through F give the temporary group call ID number associated with this group call - max of 64.
Block M	1	Format: 01ABCDEF, where A through F give the current block number associated with this group call's address list - max of 64.
Block N	1	Format: 01ABCDEF, where A through F give the total number of blocks associated with this group call's address list - max of 64.
Options Bitmap #1	1	Options Bitmap #1 - Format: 01ABCDEF, where A through F are individual flags indicating which Identifier-related optional fields are present in relation to Identifier #1.
Identifier #1	10	Identifier #1
Function Code #1	1	Function Code #1 - optional, present if specified in Options Bitmap #1 - bit D.
Expansion Options Bitmap #1	1	Expansion Options Bitmap #1 - optional, present if specified in Options Bitmap #1 - bit F. Format: 01ABCDEF, where A through F are individual flags indicating which Identifier -related optional expansion fields are present in relation to Identifier #1.
	• • •	
Options Bitmap #n	1	Options Bitmap #n - Format: 01ABCDEF, where A through F are individual flags indicating which Identifier-related optional fields are present in relation to Identifier #n.
Identifier #n	10	Identifier #n
Function Code #n	1	Function Code #n - optional, present if specified in Options Bitmap #n - bit D.
Expansion Options Bitmap #n	1	Expansion Options Bitmap #n - optional, present if specified in Options Bitmap #n - bit F. Format: 01ABCDEF, where A through F are individual flags indicating which Identifier-related optional expansion fields are present in relation to Identifier #n.

ETB (end-of-block flag)	1	

NOTE: The final end flag in the last data field of a packet must be an ETX to indicate the end of the packet and the CRC-16 will follow. (Refer to note in section 4.4)

4.4.11 Group Call Data Block Format (Optional - Refer to section 5.13)

Field Name	Number of Bytes	Comments
Block type flag (49 hex)	1	
Group Call ID	1	Format: 01ABCDEF, where A through F give the temporary group call ID number associated with this group call - max of 64.
Message Text	0 - N	
ETB (end-of-block flag)	1	

NOTE: The final end flag in the last data field of a packet must be an ETX to indicate the end of the packet and the CRC-16 will follow. (Refer to note in section 4.4)

4.4.12 Congestion Control Block Format (Optional - Refer to section 5.14)

Field Name	Number of Bytes	Comments
Block type flag (4A hex)	1	
CSN	1	Congestion Sequence Number.
Object Block (first)	>= 5	Destination type (see 5.14.1.2).
Object Block (intermediate or last)	>= 4	Destination or CZL type (see 5.14.1.2). Always CZL type if last Object Block.
Object Block (intermediate or last)	>= 4	Destination or CZL type (see 5.14.1.2) - Optional. Always CZL type if last Object Block.
	• • •	
ETB (end-of-block flag)	1	

NOTE: The final end flag in the last data field of a packet must be an ETX to indicate the end of the packet and the CRC-16 will follow. (Refer to note in section 4.4)

5.0 Data Field Interpretation

The following sections show the interpretation of the indicated data field of the data blocks.

5.1 Common Fields

5.1.1 Character Set

All characters consist of eight (8) bits. Their value can range from 00 thru FF hex. All protocol generators and receivers must recognize the standard ASCII character set 00 thru 7F hex. See figure 2 for ASCII code table. All intermediate nodes must be capable of passing the entire character set 00 thru FF hex. In order to use the characters in the range of 80 thru FE, their meaning must be agreed upon by the source and destination systems.

5.1.2 Control Flags

The control flags listed in figure 3 are not permitted in the header or text/data portions of the packet unless used with the control character modifier described in section 4.0.

5.1.3 Block Type Flag

This is the first field in every data block. It describes the format of the data block being received. The following data block types are currently defined.

Description	Hex value	ASCII	
ETE request	3E	>	
ETE response	3C	<	
CAP page	41	A	
ID page	42	B	
COMMAND	43	C	
DATA	44	D	
STATUS	45	E	
Extended CAP Page	46	F	(Optional Block)
Cap Page Group Call Address	47	G	(Optional Block)
ID Page Group Call Address	48	H	(Optional Block)
Group Call Data	49	I	(Optional Block)
Congestion Control	4A	J	(Optional Block)

5.2 Header Data Fields

The packet header contains address, inertia, and serial number information as described in this section. Each of these fields are specified as hexadecimal values which are represented in ASCII (see example in appendix A.2). The hexadecimal values may be specified as the ASCII digits 0-9 as well as the characters "A"-"F". For example, the two-hexadecimal-digit field containing the inertia (see section 5.2.2) is represented as follows: decimal 0 is "00", decimal 32 is "20", and decimal 255 is "FF". Prior to release 3.4 of this document, the specification did not explicitly state if the lower case characters "a"-"f" would also be permitted in the header. Some implementations accept either upper or lower case "A" through "F", and a few implementations utilize lower case in headers being sent. But, since there are a large number of implementations which only support the upper case characters, it is recommended that new implementations only utilize upper case .

5.2.1 Destination Address

This field is a four-hexadecimal-digit value that represents the 16 bit numeric address assigned to the message destination. This allows for 65,535 possible destination addresses. A binary destination address of zero is reserved for node to node control and is not permitted for general communications.

5.2.2 Inertia

This field is a two-hexadecimal-digit value that represents the number of nodes that the current packet may pass through before being removed from the network. This value is decremented at each node that re-transmits the packet. If the inertia value reaches zero, the node must discard the packet and report the error (if possible). This is used to prevent a packet from remaining in the network forever due to routing errors.

5.2.3 Source Address

This field is a four-hexadecimal-digit value that represents the 16 bit numeric address assigned to the message originator. This allows for 65,535 possible source addresses. A binary source address of zero is not permitted.

5.2.4 Packet Serial Number

This field is a two-hexadecimal-digit value (modulo 256). This value is assigned to the packet before it is transmitted from one node in the network to another. This number is used by the receiver to distinguish an original packet from a duplicate (redundant transmission). The receiver must maintain a table of at least 64 serial numbers to recognize duplicate packets. In a one-way distribution network (simplex) time diversity may be enhanced by grouping packets in redundant transmission sets and repeating the packet set. The total number of packets in a set must not exceed 64 in length.

The sender assigns a unique number to each packet before attempting transmission. Serial numbers are incremented from one (01 hex) to 255 (FF hex), each time a new packet is transmitted. If a packet is re-transmitted for redundant transmission, in response to NAK flag, or time-out, the same serial number is used.

Serial number zero (00 hex) is reserved for startup synchronization. If a packet is received with serial number zero, the receiver serial number table is cleared, then the packet is processed normally. At initialization each transmitter will use serial number zero in the first packet to clear the receiver serial number table. The transmitter should use an empty packet (no data blocks) to pass the zero serial number (refer to section 3.5). No application transaction blocks should appear in the packet which contains the zero serial number.

5.2.5 Header Extension

The header extension is currently only used for extending the source and destination address data fields in the header. The source and destination extension is defined as !DDDDSSSS where the character ! (21 hex) is the header extension type and the code DDDD and SSSS are both a four-hexadecimal-digit value that represents a 16-bit destination and source address extension, respectively.

5.3 ETE Request Fields

5.3.1 ETE Identifier

This field contains two bytes (range 40 - 7F hex) which are interpreted as follows:

first byte: 01ABCDEF
second byte: 01GHIJKL

where:

AB: means:

00 middle block of multi-block sequence
01 last block of multi-block sequence
10 first block of multi-block sequence
11 single block (not part of a sequence)

and CDEFGHIJKL is a 10-bit segment number to be used in the ETE response block. The values used for the segment number must be adjacent for a multi-block sequence.

If a node is not capable of receiving a multi-block sequence, it must respond with an ETE response with the "Multi-block OK" flag set to zero.

5.3.2 Any Other Data Block

This portion is the data block to which the ETE request applies.

5.4 ETE Response Fields

5.4.1 ETE Identifier

This field contains the two-byte identifier which identifies the ETE request for which the response was generated. This should be identical to the two byte field received in the ETE request.

5.4.2 ETE Response Code

This field contains a one-byte code which indicates the status of the data transfer related to the identifier. This byte has the format 01ABCDEF where:

A	is the Multi-block OK flag
B	is the Reject / Abort flag
C	is reserved
DEF	is a three-bit window size value

The Multi-block OK flag is set to one (1) if multi-block sequences are supported by the destination.

The reject / abort flag is set to one (1) if the data block is rejected by the destination.

The window size is a three-bit value (1 to 7) which indicates the number of packets the originator may have out pending response by the destination. This is used by the destination to control the amount of data it is required to buffer. A value of 000 is not allowed.

5.4.3 Reject Code

This field is a one-byte code (range 40 - 7F hex) which is the reject code applying the data block identified by the identifier field. The currently defined values for the reject code are:

Description	Hex Value
No reject	40
Media fail - full	41
Incompatible data type	42
Invalid segment number	43
Incorrect block length	44
Access barred	45
Incompatible destination	46
Destination out of order	47
Remote procedure error	48
No multi-block sequences allowed	49

5.5 CAP Page Fields

5.5.1 Page Type

This is the second field in the CAP page block and it describes the pager signaling code scheme.

Description	Hex value	ASCII	
POCSAG - 512 baud (CCIR #1)	50	P	
POCSAG - 1200 baud	70	p	
POCSAG - 2400 baud	51	Q	
Golay Sequential Code	47	G	
Golay Type II-A	54	T	
FLEX™	46	F	*
FLEX™-Augmented	66	f	*
ERMES	65	e	*
ERMES-Augmented	72	r	*
NEC D2 format	6E	n	
NEC D3 format	4E	N	
APOC 1200	61	a	
APOC 2400	41	A	
HSC	48	H	
5/6 tone	35	5	
2 tone	32	2	
RDS	52	R	
MBS	4D	M	
Multitone - Mark IV	34	4	
- Mark V	56	V	
- Mark VI	36	6	
Spantel	53	S	
DTMF	44	D	
DTMF Networking	64	d	
Echo format	45	E	
Newspager - 512 baud	5C	\	
Newspager - 1200 baud	7C	|	
Newspager - 2400 baud	2F	/	
Load Management Format (SA206)	4C	L	**
Reserved	2E	.	

* Note: The FLEX™, FLEX™-Augmented, ERMES, ERMES-Augmented, and APOC page types are sent as part of the Extended CAP page block. The FLEX™ page type may also be sent as part of the CAP Page block. Full FLEX™, ERMES, and APOC addressing support requires the use of the Extended CAP page in order to adequately define the full pager address range and all parameters required to send a remote page. The CAP page block may only be used in those cases where various pager specific parameters may be derived algorithmically from the pagers CAP code value. In subsequent areas, all references to FLEX™-related properties also reflect FLEX™-Augmented properties.

** Note: The Load Management Format (SA206) page type is a non-paging format which is used to transmit Scientific Atlanta's proprietary SA105, SA205 or SA206 protocols using existing paging infrastructure. The messages consist of strings of ASCII 0 and 1 (Hex 30 and 31) digits. Over TNPP these messages are sent as CAP pages with Page Type L (Hex 4C) and Page Class N (Hex 4E). The Load Management devices do not use function digits or capcodes. The default CAP page function code value 0000 and capcode value 00000001 shall be used for Load Management messages, all other capcode and function code values are reserved.

For further information on the SA105, SA205 and SA206 protocols, contact Scientific Atlanta at:

Scientific Atlanta Inc.
Steve Meissel
Product Manager - Load Management Products
4311 Communications Drive
Norcross, Ga 30093
770-903-2941 (Direct Line)
770-903-2900 (General)
Email: steve.meissel@sciatl.com

Note that the inclusion of pager signaling mechanisms in this table reserves a code to represent the mechanism and does not imply that the destination system is capable of generating pages in the specific page encoding type.

5.5.2 Page Class

This is the third data field in the CAP page block and it describes the message encoding class of the pager.

	Description	Hex value	ASCII
	Beep Only	42	B
	Numeric Display	4E	N
	Shifted Numeric	6E	n
	Alphanumeric display	41	A
*	Voice	56	V
	Data/Transparent Data	44	D
	Special Class 1	31	1
	Special Class 2	32	2
	Special Class 3	33	3
	Special Class 4	34	4
	.	.	.
	.	.	.
	.	.	.
	Special Class 9	39	9

The TNPP Data page class is sometimes referred to as the Transparent Data paging class. This class supports the over-the-air transmission of binary data. The text field of the block contains 8 bit binary data represented in a form which adheres to the "transparency insertion" rules discussed in section 4.1.

The Special Class designators 1,2,3, etc., are used to invoke paging functions which are specific to a particular pager type.

* NOTE: This document does not attempt to define how voice paging will work through a network but is simply reserving the V class for future use. (The section on Optional Features and Capabilities reviews how one manufacturer has implemented voice paging within TNPP networks).

The Special Class designators which are currently assigned are as follows:

Pager Type	Special Class	Definition
FLEX™	1	Special format numeric message designator.

2	Short numeric message designator including up to 3 digits for "short address" pagers and up to 8 digits for "long address" pagers.
3	Tone only message designator for source number 0. If a message sequence number is specified (see Extended CAP Page information) this is a tone only sequenced page for source 0. All other tone only pages may be specified as Pager Class "B" and a non-zero function digit to represent the source number.
4	Secure Message type 0 (Alphanumeric) designator.
5	Data page class with mail drop designator.
6	Alphanumeric page class with mail drop designator.
7	Secure Message type 1 (Defined Fields) designator.
8	Secure Message type 2 (Binary) designator.
9	Secure Message type 3.

5.5.3 RF Channel Designator

The RF channel designator is a one-byte field of the format of 01ABCDEF where bits ABCDEF is the binary channel code at the destination system.

5.5.4 RF Zone Designator

The RF zone designator is a one-byte field of the format 01ABCDEF where the code ABCDEF is the binary zone ID on the selected channel.

5.5.5 CAP Page Function Code

The function code is common to the CAP page and ID page data blocks. This field is one byte in length and has the format 01P0ABCD where P is a call priority indicator and the value ABCD has the following meaning:

0000 - default pager address / function
0001 - address / function 1
0010 - address / function 2
0011 - address / function 3
0100 - address / function 4

0101 thru 1111 - reserved for future expansion
(used for pagers supporting more than 4 functions)

If the priority indicator is set to a 1, the destination system is expected to process the call with encoding priority.

5.5.6 CAP Code

This is an eight-byte field. It is an eight digit ASCII representation of the pager CAP code. This field should be right justified and all unused characters should be filled with an ASCII zero (30 hex).

This field of eight digits will be interpreted as follows for the various pager types:

Pager type:	Cap Code:	CAP field:
Golay	123456	00123456
POCSAG	1234567	01234567
FLEX™	12345678	12345678
NEC D2	123456	00123456
NEC D3	123456	00123456
Six tone	123456	00123456
HSC	123456	00123456
Five tone	12345	00F12345
Two tone	See section on Optional Features and capabilities	

Note the substitution of the hex digit F for the nonexistent preamble in the five-tone pager type. This is done to identify a five tone CAP code.

5.5.7 CAP Page Message Text

The message text data field consists of characters that are to be sent out with the associated page. This field is variable in length and is limited by the pager type and the character limit on packet size. Refer to section 4.1 regarding transparency insertion requirements and potential compatibility issues with systems prior to TNPP revision 3.8.

5.6 ID Page Fields

5.6.1 ID Page Function Code

The function code is common to the CAP page and ID page data blocks. This field is one byte in length and has the format 01P0ABCD where P is a call priority indicator and the value ABCD has the following meaning.

0000 - default pager address/function
0001 - address/function 1
0010 - address/function 2
0011 - address/function 3
0100 - address/function 4
0101 thru 1111 - reserved for future expansion
(used for pagers supporting more than 4 functions)

If the priority indicator is set to a one (1), the destination system is expected to process the call with encoding priority.

5.6.2 ID Page Identifier

The ID page identifier field consist of ten (10) ASCII characters that represent the customer ID to be paged. Only non-control ASCII characters may be used (20 to 7F hex). This field should be left justified and all unused characters should be filled with an ASCII space (20 hex).

5.6.3 ID Page Message Text

The message text data field consists of characters that are to be sent out with the associated page. This field is variable in length and is limited by the pager type and the character limit on packet size. Refer to section 4.1 regarding transparency insertion requirements and potential compatibility issues with systems prior to TNPP revision 3.8.

5.7 COMMAND Block Fields

5.7.1 Manufacturer Designators

This is a three-byte field that consists of a manufacturers ID (command set). Each manufacturer will be assigned a three letter code to prevent manufacturer-specific commands from being misinterpreted. Any commands that are determined to be non-specific (common among users) will be assigned to a standard command set with a manufacturer designator / Command set code of '000'. Only non-control characters may be used (20 to 7F hex).

Manufacturer	Code
Common / standard set	000
ASE	ASE
Advanced Interactive Systems	AIS
BBL Industries, Inc.	BBL
Blackhawk EPD, Inc.	BLA
Commonwealth Communication Industries	COM
Commonwealth Communication Industries	CCI
CTI Manufacturing	CTI
Day Telecommunications	DAY
Freeman Engineering Associates, Inc.	FRE
Glenayre Electronics	GLE
Information Radio Technology	IRT
Metriplex, Inc.	MPX
Microlink, Inc.	MIC
Motorola, Inc.	MOT
Quintron Corporation	QUI
Real Time Strategies	RTS
Statistical Control Systems	SCS
Spectrum Comm. and Electronics Corp.	SCE
Teknow	TEK
TGA	TGA
Telelink Technologies Inc.	TTI
Unipage, Inc.	UNI
Zetron, Inc.	ZET

5.7.2 Command Code

This is a three-byte data field representing the command for the manufacturer designator / command set specified in the previous data field. These commands are defined by each manufacturer. Only non-control characters may be used.

5.7.3 Command Parameters

This data field can vary from 0 to 999 bytes. It consists of the parameters needed to execute the command in the previous data field and is defined by each manufacturer.

5.7.4 COMMAND Block Example

The COMMAND block type is used to pass node setup and control commands. Since most commands will be manufacturer dependent, each command includes a three letter manufacturer code. There is a prefix defined for "standard" commands, but no commands have been defined to date. The following example represents a typical usage of a manufacturer-dependent command:

Byte:	Meaning:
C	command block identifier
XYZ	manufacturer code (XYZ Paging Equipment)
SQL	command code (Set Queuing Limit)
1,	first parameter (queue number)
15	second parameter (maximum number of blocks)
<ETB>	end of command block

Please note that for all manufacturer-dependent commands, all bytes after the manufacturer code are open for definition. The only restrictions are:

1) the command must start with an ASCII C followed by the manufacturer code,
2) the command may not contain a reserved flag byte,
3) the command block length must not exceed 1007 bytes,
4) the command block may not cause the packet length to exceed 1024 bytes.

If a command acknowledgment is required, the originating node must add an ETE request prefix before the command.

5.8 DATA Block Fields

5.8.1 User Defined Data

The message text field consists of up to 1000 characters that are to be defined by the user or manufacturer. Only non control characters may be used.

5.9 STATUS Block Fields

5.9.1 Status Code

This field contains one byte which describes the condition causing the status message. There are three possible codes which may be passed in this field:

Description:	Hex value	ASCII
error set	53	S
error reset	52	R
local priority change	50	P
non-latched error	45	E

5.9.2 Priority

This field contains one byte which describes the local evaluation of the priority of the error. The value may range from an ASCII 1 (highest priority) to an ASCII 9 (lowest priority).

5.9.3 Error Code

This field contains a four-digit decimal error code number. The code number describes what type of error occurred. The first 1000 codes (0000-0999) are reserved for committee assignment to common error types. The remaining 9000 error codes (1000-9999) are available for manufacturer definition.

5.9.4 Date

This field contains the date of the error in the format:

MMDDYY

where MM is the month (01 - 12), DD is the day (01 - 31), and YY is the last two digits of the year (00 - 99).

5.9.5 Time

This field contains the time of the error in the format:

HHMM

where HH is the hour (00 - 23) and MM is the minute (00 - 59).

5.9.6 Text

This is an optional field containing either a text description of the error, further details on the nature of the error, or both.

5.10 Extended CAP Fields - Optional Block

The Extended CAP Page block is an optional TNPP data block. It may be utilized only if the destination terminal is programmed to support this block type.

The Extended CAP block is similar to the CAP block described in section 5.5 with the following additions:

- Support for up to 32 page delivery priorities as opposed to the 2 priorities in a CAP block.
- The ability to support a wider pager address range and transfer pager specific information for those page types which require additional data for full support.
- The ability to define a message sequence number for those page types which integrate a sequence number as part of the pager encoding format.

5.10.1 Page Type

Any page type which may be sent in the CAP Page block may be sent in the Extended CAP Page block. Refer to section 5.5.1 for a list of valid codes.

5.10.2 Page Class

The Extended CAP Page supports each of the Page Class codes defined in section 5.5.2.

5.10.3 RF Channel Designator

This field has the same definition and format as in the CAP Page block. Refer to section 5.5.3.

5.10.4 RF Zone Designator

This field has the same definition and format as in the CAP Page block. Refer to section 5.5.4.

5.10.5 Function Code

The function code field of the Extended CAP Page is similar to its definition in the CAP Page block (section 5.5.5) with the exception of the priority indicator. The format is 0100ABCD where ABCD has the following meaning:

0000 - default pager address/function
0001 - address/function 1
0010 - address/function 2
0011 - address/function 3
0100 - address/function 4
0101 thru 1111 - reserved for future expansion
(used for pagers supporting more than 4 functions)

5.10.6 MSN Flag/Priority Designator

The priority designator is used to define the priority at which the page request should be processed at the destination paging terminal. The bit format of this one byte field is 01MABCDE where:

M is a page class modifier flag which is set to 1 if the page request contains a message sequence number (MSN) Field. The MSN field itself is used to forward MSN information to remote systems which will eventually transmit to a pager type which is capable of receiving a message sequence number as part of the "over-the-air" page.

ABCDE is a 5 bit binary field which defines one of up to 32 paging priorities. The definition of the priority values is as follows:

Priority Designators	Definition
0	Normal
1	Priority (Same as section 5.5.5 and 5.6.1)
2 - 15	Reserved for industry wide standardization of priority definition (defined later in this section)
16 - 31	Carrier or paging infrastructure equipment manufacturer/model specific priority definitions

Priority codes in the range 2-15 are reserved for common meaning across all implementations of TNPP regardless of equipment manufacturer. Any specific values defined in this specification which fall within this range, may be used. All undefined values in this range remain reserved and should not be utilized.

Priority codes in the range 16-31 have been set aside for manufacturer specific use. The definition, meaning and ultimate functionality which results in the use of any values in this range is specific to the particular manufacturers equipment which is utilizing it. The primary purpose of values in this range is to convey service priority information within a network of equipment manufactured by a particular vendor or within a specific paging service providers network, if the paging control equipment is supporting Carrier specific priority definitions.

It is important to note the implication of this definition when crossing a boundary between the equipment of different manufacturers. The definition of values in this range in one manufacturers equipment could differ from the definition of values in this range from another manufacturer. Crossing a boundary between equipment vendors could potentially result in unexpected or undesirable processing of a received message.

To avoid such potential problems, adherence to at least one of the following recommendations is desirable:

- utilize industrywide accepted priority definitions across multi-vendor equipment boundaries within a network
- provide a mechanism at the sending system to redefine each manufacturer specific value to an appropriate receiving system value between 0-31 before forwarding a message across equipment boundaries
- provide a mechanism to change a received manufacturer specific value to an appropriate value between 0-31 before being processed at the receiving equipment
- ensure that the use of particular values in the 2-15 range, when transferred between sending and receiving equipment, does not cause undesirable results. It must be noted here that a future upgrade in the sending or receiving equipment could later result in an incompatibility and thereby require the use of one of the other recommended mechanisms to resolve the incompatibility.

Industrywide Priority Designator	**Definition**
2 - 15	Currently undefined and reserved

5.10.7 CAP Code Field

The CAP Code field size and its interpretation shall be specific to each pager type. The size of field is as follows:

Pager Type	Field Size in Bytes	
Golay	8	See note A
POCSAG	8	See note A
NEC D2	8	See note A
NEC D3	8	See note A
Five/Six tone	8	See note A
HSC	8	See note A
FLEX™	13	See note B
FLEX™-Augmented	12+5	See note D
ERMES	12	See note C
ERMES-Augmented	19	See note E
APOC	10	See note F
All others listed in Section 5.5.1	8	See note A

Note A: The definition of the CAP Code field is the same as that which is specified in section 5.5.6.

Note B: The Cap Code field for a FLEX™ page is a 13 character field. This is broken down as follows:

ABCDEFGHI J K LM

where:

ABCDEFGHI	is a 9 ASCII digit decimal pager number
J	is an ASCII digit value from 0 to 5 (phase - 0, 1, 2, 3, Any, All)
K	is an ASCII digit value from 0 to 7 (collapse value)
LM	is a hexadecimal value which ranges from 00 to 7F (base frame). The ASCII character codes capital "A" through "F" shall be utilized.

The proper values to assign to the various sub-fields can be derived from the Cap Code label on the pager. The format of the label is:

ppp q R sssssssss

Where fields "p", "q", and "s" are numeric and "R" is alphabetic. Fields "R" and "s" always appear on the label while "p" and "q" are optional.

The conversion from a label to TNPP Extended Cap Code field is as follows:

ABCDEFGHI — is the same as the 7 or 9 digit value of label field "s", with leading zeroes if necessary to create a 9 digit value.

J

Label Field "R" Value	J Value
U	0
V	1
W	2
X	3
E, F, G, H, Y	4
I, J, K, L, Z	5

If the label field "R" is A, B, C, or D, then the "J" value is derived from label fields "R" and "s" algorithmically. The formula is:

J = (Integer ((s-t)/4)) Modulo 4

which will result in a value from 0 to 3. The value of "t" can be taken from the following table:

Label Field "R" Value	"t" Value
A	0
B	1
C	2
D	3

K — If a "q" field exists, its value is used as digit K. When field "q" is not specified, K is set to 4.

LM — If label field "p" exists, this 3 digit decimal digit value is converted from decimal to its 2 hexadecimal digit equivalent to form the "LM" value. If "p" is not specified, the 3 decimal digit value is derived from label fields "R" and "s" algorithmically.

The formula is:

p = (Integer ((s-t) / 16)) Modulo 128

which results in a value from 0 to 127. The value of "t" can be taken from the following table:

Label Field "R" Value	"t" Value
A, E, I	0
B, F, J	1
C, G, K	2
D, H, L	3

Note C: The Cap Code for an ERMES page is a 12 character field. This is broken down as follows:

ABCDEFGH IJ K L

where:

ABCDEFGH — is an ASCII represented eight digit value which is the decimal equivalent of the pager address.

IJ — is an ASCII represented two digit decimal value which ranges from 00 to 31 and represents the pagers Subsequence Mask.

K — is one character ASCII digit value from 0 to 5 representing the home network cycle mask.

L — is a one character ASCII digit value from 0 to 5 representing the external network cycle mask.

Note D: The Cap Code field for a FLEX™-Augmented page is a 12 +5 character field of the format ABCDEFGHIJKLMNOPQ, where Cap Code fields 'M' through 'Q' are optional based on the content of other Cap Code fields. (See figures 9 and 10) This is broken down as follows:

ABCDEFGHIJ KLMNOPQ

where:

ABCDEFGHIJ is the FLEX™ pager address - a 10 digit decimal number. Interpretation of short addresses vs. long addresses is determined solely on the defined address ranges.

K The Collapse and Information Fields byte is a one-byte field of the format of AB C D E FGH. Bits ‘A’ and ‘B’ are reserved and set to zero. Bit C, when set to '1' indicates that the message is an information service message. Bit D indicates the presence of the optional base frame field 'M', when set to '1'. If bit ‘E’ is set to ‘1’, then the message is a retransmission. Bits ‘F’ through ‘H’ are used to convey the FLEX™ collapse value within the range of 0 through 7.

- A - B Reserved, set to 0.
- C=0 Not an Information Service message.
- C=1 An Information Service message.
- D=0 Do not expect optional Base Frame field ‘M’. Base frame is to be derived from the Cap Code.
- D=1 Base Frame. Expect optional Base Frame field ‘M’.
- E=0 The message is not a retransmission.
- E=1 The message is a retransmission.
- F - H FLEX™ collapse value, a number from 0 to 7

L The Phase and Options Fields byte is a one-byte field of the format of ABCD E F G H. Bits ‘A’ through ‘D’ are used to convey the Phase Assignment(s) of Phase 0 through 3 via the setting of the flag bits in ‘A’ through ‘D’ respectively. The setting of multiple flag bits to ‘1’ indicates that the message should be sent in all the indicated phases as would be the case in a device-based (‘Pager’ or ‘common address’) group call. The setting of all flags to '0' indicates 'Any Phase'. If bit ‘E’ is set to ‘1’, then the device shall be treated as defined by the FLEX™ ‘All Phase’ rules, regardless of the setting of bits ‘A’ through ‘D’. Bits ‘F’ through ‘H’ are used to activate optional bytes.

- A - D Phase Assignment Flag(s). Set to ‘1’ for ‘Active’. All flags set to ‘0’ indicates ‘Any Phase’. The setting of multiple flags to ‘1’ indicates that the message should be sent out on all the indicated phases as would be the case in a device-based group call.
- E=0 Do not use ‘All Phase’ device rules.
- E=1 Use ‘All Phase’ device Rules.
- F=0 No RF Channel byte. Do not expect optional RF Channel field ‘N’.

F=1 RF Channel. Expect optional RF Channel field 'N'.

G=0 No Special Information. Do not expect optional Special Information field 'O'.

G=1 Special Information. Expect optional Special Information field 'O'.

H=0 No Expansion Options Bits field. Do not expect optional Expansion Options Bits fields 'PQ'.

H=1 Expansion Options Bits field. Expect optional Expansion Options Bits fields 'PQ'.

M The Base Frame Field byte is an optional one-byte field of the format of A BCDEFGH. Bit 'A' is reserved and set to '0'. Bits 'B' through 'H' are used to define the FLEX™ base frame number when the base frame can not be uniquely determined by other means (e.g., default, calculation). The range of this value is Hex 00 through Hex 7F. This field is included only if bit 'D' in Cap Code field 'K' is set to '1'.

N The RF Channel Field byte is a one-byte field of the format of A B C DEFG H. These bits are used to define Options Bit flags.

A=0 No Frame Offset Support. This device does not support Frame Offset.

A=1 Frame Offset Support. This device supports Frame Offset.

B=0 Group Frame Offset Support. All of the devices in a device-based group call (common address) support Frame Offset.

B=1 Group Frame Offset Support. Not all of the devices in a device-based group call support Frame Offset.

C=0 No Max Carry On Support. This device does not support Max Carry On.

C=1 Max Carry On Support. This device supports Max Carry On.

D - G Base Traffic Management Group Flag(s). If all flags are set to '0', then the device does not support Traffic Management channel reassignment. The setting of multiple flag bits to '1' indicates that the message shall be sent on all the indicated managed channels as would be the case in a device-based group call.

TNPP Bit -> Traffic Management Group Channel \/	**D**	**E**	**F**	**G**
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

H=0 All devices on this address support Traffic Management. Valid only if at least one of bits 'D' through 'G' is set.

H=1 Not all devices on this address support Traffic Management. Send the message on both the TNPP-indicated channel and on the output terminal's managed channel(s). Valid only if at least one of bits 'D' through 'G' is set.

O The Special Information byte is an optional one-byte field of the format of A B C D EFGH where Bits 'A' through 'H' are defined by the associated Page Class definition. If the Page Class indicates a Data/Transparent Data, Data page class with mail drop designator, Secure Message with Defined Fields, Secure Message Binary, or Secure Message with Status Information Field, then bits 'E' through 'H' identify the Information Blocking Length. If the Page Class indicates an Alphanumeric Display, Secure Message Alphanumeric, or Alphanumeric page class with mail drop designator, then Bits 'E' through 'H' identify the Shift Character Set rules to be followed when using Enhanced Fragmentation Rules with this device. Bits 'A' through 'D' are reserved and set to '0' in both cases. Unless specified, binary information is expected to use 8 bit blocking length, and alphanumeric information is expected to use English fragmentation rules. This field is included only if bit 'G' in field 'L' is set to '1'

Page Class is Data/Transparent Data, Data page class with mail drop designator, Secure Message with Defined Fields, Secure Message Binary, or Secure Message with Status Information Field:

E - H Information Blocking Length

TNPP Bit -> Information Blocking Length (in bits) \/	**E (B_3)**	**F (B_2)**	**G (B_1)**	**H (B_0)**
16	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

A - D Reserved and Set to 0

Page Class is Alphanumeric Display, Secure Message Alphanumeric, or Alphanumeric page class with mail drop designator:

E - H Enhanced Fragmentation Rules - Shift Character Sets

TNPP Bit -> Enhanced Fragmentation Rules - Shift Character Sets \/	**E**	**F**	**G**	**H**
Shift Character Set 1	0	0	0	0
Shift Character Set 2	0	0	0	1
Shift Character Set 3	0	0	1	0
Shift Character Set 4	0	0	1	1
Shift Character Set 5	0	1	0	0
Shift Character Set 6	0	1	0	1
Shift Character Set 7	0	1	1	0
Shift Character Set 8	0	1	1	1
Shift Character Set 9	1	0	0	0
Shift Character Set 10	1	0	0	1
Shift Character Set 11	1	0	1	0
Shift Character Set 12	1	0	1	1
Shift Character Set 13	1	1	0	0
Shift Character Set 14	1	1	0	1
Shift Character Set 15	1	1	1	0
Do Not Fragment	1	1	1	1

A - D Reserved and Set to 0

PQ The Expansion Options Bits Field with Length to End of Expansion Fields bytes are a two or more byte field of the format of A B C D E F G H (Byte 'P') and IJKLMNOP (Byte 'Q'). These bits and the associated length value byte are used to define Expansion Options Bit flags and allow features to be added to the FLEX™-Augmented Cap Code fields without breaking existing implementations. These fields are included only if Bit 'H' in Cap Code Field 'L' is set to '1'.

- A=0 No additional Expansion Options Bits bytes.
- A=1 Expect an additional Expansion Options Bits byte.
- B - H Reserved and set to '0'. Each flag can be used in a future version of TNPP to represent a new feature.
- I - P Byte count from the start to the end of any to-be-defined expansion fields. This field's value may range from Hex 00 to Hex FF.

Note E: The Cap Code Field for ERMES-Augmented page is an 19 character field of the format ABCDEFGHIJKLMNOPQRS. This is broken down as follows:

ABCDEFG HI J K LM N OPQ R S

where:

ABCDEFG is an ASCII represented seven digit value which is the decimal equivalent of the pager local address. The range is 0000001 to 4194303.

HI is an ASCII represented two digit decimal value which ranges from 00 to 31 and represents the pagers Subsequence Mask.

J is one character ASCII digit value from 0 to 5 representing the home network cycle mask.

K is a one character ASCII digit value from 0 to 5 representing the external network cycle mask.

LM is an ASCII represented two digit decimal value which ranges from 00 to 15 and represents the pagers Frequency Subset Number.

N is a one character ASCII digit value from 0 to 7 representing the pagers Zone Code.

OPQ is an ASCII represented three digit decimal value which ranges from 000 to 099 and represents the pagers Country Code. (The field allows 0..127, but currently, only country codes 00..99 are defined in the ERMES documentation)

R is a one character ASCII digit value from 0 to 7 representing the pagers Operator Code.

S is a one-byte field of the format AB C DEFGH. Bits 'A' and 'B' are reserved and set to zero. Bit 'C', when set to '1' indicates that the message is an Urgent message. Bits 'D' through 'H' is a number from 0 through 31 which indicates the character set of the message as specified in ETS 300 133. The default character set number is 0.

A - B Reserved, set to 0.
C=0 Not an Urgent message.
C=1 An Urgent message.
D - H ERMES Character Set. A number from 0 to 31.

Note F: The Cap Code Field for an APOC page is a 10 character field of the format ABCDEFGHIJ. This is broken down as follows:

ABCDEFGH I J

where:

ABCDEFGH is an eight character ASCII representation of the decimal equivalent of the pagers cap code.

I	is an ASCII represented one digit decimal value which ranges from 0 to 3 and represents the pagers batch number (3 is used for batch mode).
J	is an ASCII represented one digit decimal value from 0 to 4 and represents the sub batch number.

5.10.8 Message Sequence Number

If the M bit of the MSN/Priority Designator field is set to 1, the Extended CAP Page block contains a Message Sequence Number (MSN) field. If the M bit is 0, the MSN field does not exist.

The MSN field is a 2 byte field which contains 2 ASCII hexadecimal digits. The field can be used to represent values from 00 to FF. Some paging types may limit sequence numbers to a subset of the 256 possible values.

5.10.9 Message Text

This field is the same as in the CAP page format (section 5.5.7). For the "Data" page class, binary data (as opposed to ASCII printable characters) are placed in this field, subject to the data transparency rules presented in section 4.1.

5.11 Cap Page Group Call Address Fields - Optional Block

The Cap Page Group Call Address block is an optional TNPP data block. It is utilized only if the destination terminal is programmed to support this block type and the Group Call Data block that is associated with this block type to specify a complete Group Call. The purpose of this block is to pass a list of Extended Cap Page Cap Codes to the destination TNPP node to be used with the corresponding Group Call Data block. These two block types must be processed together as a Group Call.

Address lists longer than may be accommodated in one TNPP Cap Page Group Call Address block are defined through the use of multiple Cap Page Group Call Address blocks. Such multiple Cap Page Group Call Address blocks are tied together through the Block M (current block) of Block N (total number of blocks) fields and the Group Call ID number field. The order of reception of these blocks and the associated Group Call Data block is not significant.

The protocol does not specify the period of time a destination node should wait to collect the Group Call Data block and all associated Cap Page Group Call Address

blocks or a mechanism to request the resending of a missing block. Presumably, after an implementation-specific period of time beyond the receipt of the first block of a new group, if all blocks have not been collected the destination may attempt to process the partial group. The time-out period should be based upon the total number of blocks expected for the particular group page request.

There is no provision in this method for older, non-group-call-supporting paging terminals to support any portion of this. Thus, older terminals are expected to ignore this new block.

5.11.1 Group Call ID

The Group Call ID is a mandatory one-byte field of the bit format 01ABCDEF. The binary value of bits A through F represents a specific Group Call ID number (0-63) which will appear in each of the Cap Page Group Call Address blocks and the associated Group Call Data block which, in their entirety, represent a single group page request. In order for TNPP destinations to collect and process group page requests originating simultaneously from several remote nodes, the destination node will collect all Cap Page Group Call Address and Group Call data blocks containing the same Group Call ID which have come from the same TNPP Source Address as a single group page request. A Cap Page Group Call Address or Data block containing the same Group Call ID but originating from another TNPP source address is a different group page request. The use of the Source Address, given in the header of this block, with the Group Call ID to form a unique identification number allows the reuse of Group Call IDs across multiple Source Addresses. This stretches the number of simultaneously-open Group Call transactions to the number of Source Addresses multiplied by the number of Group Call IDs.

5.11.2 Block M

The Block M is a mandatory one-byte field of the format of 01ABCDEF where the binary value of bits A through F are used to identify the block number of the current packet in relation to the total number of blocks given in the Block N field. These block numbers identify the current list of addresses in relation to the total number of blocks used for addresses. Blocks may be processed out of sequence.

5.11.3 Block N

The Block N is a mandatory one-byte field of the format of 01ABCDEF where the binary value of bits A through F are used to identify the total number of blocks used to convey the address list. Blocks may be processed out of sequence.

5.11.4 Page Type

The Cap Page Group Call Address supports each of the Page Type codes defined in section 5.5.1. In the case where one or more cap code(s) are identified as destined for an optional page type, then this field indicates the Page Type for the majority of the identified cap codes.

5.11.5 Page Class

The Cap Page Group Call Address supports each of the Page Class codes defined in section 5.5.2.

5.11.6 RF Channel Designator

This field has the same definition and format as in the CAP Page block described in section 5.5.3. In the case where one or more cap code(s) are identified as destined for an optional RF channel, then this field indicates the RF Channel for the majority of the identified cap codes.

5.11.7 RF Zone Designator

This field has the same definition and format as in the CAP Page block described in section 5.5.4. In the case where one or more cap code(s) are identified as destined for an optional RF Zone, then this field indicates the RF Zone for the majority of the identified cap codes.

5.11.8 MSN Flag/Priority Designator

This field has the same definition and format as in the Extended CAP Page block. Refer to section 5.10.6. Due to the nature of a Group Call, the MSN Flag must always be set to '0' indicating that a common Message Sequence Number is not present. Message Sequence Numbers for individual group members may optionally be specified through the use of the member specific Message Sequence Number field described in section 5.11.16 Individual Message Sequence Numbers may, however, be present in the 'Options' area.

5.11.9 Options Bitmap #n

This field defines which of the optional cap-code-related fields are present for a particular group member fields is present for the particular cap-code in question. The format for this field is 01ABCDEF where A through F are individual flags indicating which of the cap code-related optional fields are present for this group member specification in relation to Cap Code #n.

Bit Flag	Definition
Bit A	Indicates that the optional override field 'Page Type #n' is present.
Bit B	Indicates that the optional override field 'Destination Address #n' is present.
Bit C	Indicates that the optional override fields 'RF Channel #n' and 'RF Zone #n' are present.
Bit D	Indicates that the optional override field 'Function Code #n' is present.
Bit E	Indicates that the optional field 'Message Sequence Number #n' is present.
Bit F	Indicates that the optional field 'Expansion Options Bitmap #n' is present.

This field is repeated as necessary to identify the options present for each of the included cap codes. There must be a one-to-one correspondence of Options Bitmap fields to Cap Code fields.

5.11.10 Page Type #n

This optional field has the same definition and format as in the CAP Page block defined in section 5.5.1. This field is used to identify the Page Type of the indicated member of this Group Call when the Options Bitmap #n Bit A is set to one. When specified, this field overrides the Page Type field's value for the identified cap code.

5.11.11 Cap Code #n

This field has the same definition and format as in the Extended CAP Page block defined in section 5.10.7. This field is used to identify one of the members of this Group Call. This field is repeated as necessary to identify all of the included cap codes. There must be a one-to-one correspondence of Options Bitmap fields to Cap Code fields.

5.11.12 Destination Address #n

This optional field has the same definition and format as the Destination field in the header of the TNPP packet defined in section 5.2.1. This field is only used if the paging terminals have implemented the concept of a TNPP destination representing multiple terminals and the Options Bitmap #n Bit B is set to one. (This is not defined in the TNPP specification.) In this situation, the field can be used to specify a subset of the terminals implied in the Destination field of the header. This gives the member no more RF coverage than their service subscription allows, thereby saving air time in the remaining areas. This field is not utilized to route TNPP packets, but is only used as an indicator to include or exclude the member in a group page at a particular node. If a node which is collecting group page information finds members which specify this optional Destination Address field, the member is included in the locally generated group page only if this node is one of the paging terminals normally included when this Destination Address represents multiple terminals.

5.11.13 RF Channel Designator #n

This optional field has the same definition and format as in the CAP Page block defined in section 5.5.3. This field is used to identify the RF Channel of the indicated member of this Group Call when the Options Bitmap #n Bit C is set to one. When specified, this field overrides the RF Channel Designator field's value for the identified cap code.

5.11.14 RF Zone Designator #n

This optional field has the same definition and format as in the CAP Page block defined in section 5.5.4. This field is used to identify the RF Zone of the indicated member of this Group Call when the Options Bitmap #n Bit C is set to one. When specified, this field overrides the RF Zone Designator field's value for the identified cap code.

5.11.15 Function Code #n

This optional field has the same definition and format as in the Extended CAP Page block defined in section 5.10.5. This field is used to identify a member-specific Function Code when the Options Bitmap #n Bit D is set to one.

5.11.16 Message Sequence Number #n

This optional field has the same definition and format as in the Extended CAP Page block defined in section 5.10.8. This field is used to identify the Message Sequence Number of the indicated member of this Group Call when the Options Bitmap #n Bit E is set to one, not when the M bit of the MSN / Priority Designator field is set to one as is the case in section 5.10.8. When specified, this field provides a Message Sequence Number for the group member for the identified cap code. Individual Message Sequence Numbers in a Group Call are only applicable when supported by the Page Type given in either the common Group Call parameters or in the optional Page Type #n override.

The Message Sequence Number field is a 2 byte field which contains 2 ASCII hexadecimal digits. The field can be used to represent values from 00 to FF. Some paging types may limit sequence numbers to a subset of the 256 possible values.

5.11.17 Expansion Options Bitmap #n

This optional field defines which of the expansion optional cap-code-related fields is present for the particular cap-code in question. The format for this field is 01ABCDEF where A through F are individual flags indicating which of the cap code-related optional expansion fields are present in relation to Cap Code #n. This field is present when the Options Bitmap #n Bit F is set to one. All bit flags are currently reserved with the exception of Bit F which is used to indicate an additional Expansion Options Bitmap n+1 field.

Bit Flag	Definition
Bit A	Reserved.
Bit B	Reserved.
Bit C	Reserved.
Bit D	Reserved.
Bit E	Reserved.
Bit F	Indicates inclusion of an Expansion n+1 Option field Reserved.

However, it is expected that Bit F will be a flag indicating that an expansion to this field is present similar in nature to this expansion bitmap.

5.11.18 Using Group Paging Blocks

When the destination node sees a Cap Page Group Call Address block type, it must then monitor the TNPP network for other Cap Page Group Call Address block types, when the Block M of N fields indicate more than one Cap Page Group Call Address block type, and a Group Call Data block type with the same Group Call ID number. No attempt will be made to assign a 'staleness' factor to receipt of these blocks as time-out values are implementation specific and would normally be based on the number of expected blocks as specified by the "Block N" field of the Cap Page Group Call Address block. Use of the ID allows other, higher priority pages to be interspersed with the group call without affecting the group call's ability to be coherently reconstructed. The Group Call may be coherently reconstructed without the need to receive all the Group Call Address blocks when more than one Group Call Address block is indicated in the Block M of N fields. The Group Call may be coherently reconstructed regardless of the order in which the blocks are received.

The Group Call ID allows the destination node to collect each of the Cap Page Group Call Address blocks and the Group Call Data block for a particular group page as they arrive at the destination. It is up to the implementer if partial group pages are sent over-the-air as some sub-set of all address blocks are received; if the destination will wait a specific period for all N blocks to be received and then send only those received; if Group Address blocks which arrive "late" will be processed or dropped; etc.

5.12 ID Page Group Call Address Fields - Optional Block

The ID Page Group Call Address block is an optional TNPP data block. It is utilized only if the destination terminal is programmed to support this block type and its associated Group Call Data block, which together convey a complete Group Call. The purpose of this block is to pass a list of ID Pages to the destination TNPP node to be used with the corresponding Group Call Data block. These two block types must be processed together as a Group Call.

Address lists longer than that which may be accommodated in one TNPP ID Page Group Call Address block are defined through the use of multiple ID Page Group Call Address blocks. Such multiple ID Page Group Call Address blocks are tied together through the Block M (current block) of Block N (total number of blocks) fields and the Group Call ID number field. The order of reception of these blocks and the associated Group Call Data block is not significant.

The protocol does not specify the period of time a destination node should wait to collect the Group Call Data block and all associated ID Page Group Call Address blocks or a mechanism to request the resending of a missing block. Presumably, after an implementation-specific period of time beyond the receipt of the first block of a new group, if all blocks have not been collected the destination may attempt to process the

partial group. The time-out period should be based upon the total number of blocks expected for the particular group page request.

There is no provision in this method for older, non-group-call-supporting paging terminals to support any portion of this. Thus, older terminals are expected to ignore this new block.

5.12.1 Group Call ID

The Group Call ID is a mandatory one-byte field of the bit format 01ABCDEF. The binary value of bits A through F represents a specific Group Call ID number (0-63) which will appear in each of the ID Page Group Call Address blocks and the associated Group Call Data block which, in their entirety, represent a single group page request. In order for TNPP destinations to collect and process group page requests originating simultaneously from several remote nodes, the destination node will collect all ID Page Group Call Address and Group Call data blocks containing the same Group Call ID which have come from the same TNPP Source Address as a single group page request. An ID Page Group Call Address or Data block containing the same Group Call ID but originating from another TNPP source address is a different group page request.

5.12.2 Block M

The Block M is a mandatory one-byte field of the format of 01ABCDEF where the binary value of bits A through F is used to identify the block number of the current packet in relation to the total number of blocks given in the Block N field. These block numbers identify the current list of addresses in relation to the total number of blocks used for addresses. Blocks may be processed out of sequence.

5.12.3 Block N

The Block N is a mandatory one-byte field of the format of 01ABCDEF where the binary value of bits A through F is used to identify the total number of blocks used to convey the address list. Blocks may be processed out of sequence.

5.12.4 Options Bitmap #n

This field defines which of the optional identifier-related fields are present for a particular group member. The format for this field is 01ABCDEF where A through F are individual flags indicating which of the optional fields are present for this group member specification.

Bit Flag	Definition
Bit A	Reserved.
Bit B	Reserved.
Bit C	Reserved.
Bit D	Indicates that the optional override field 'Function Code #n' is present.
Bit E	Reserved.
Bit F	Indicates that the optional field 'Expansion Options Bitmap #n' is present.

5.12.5 Identifier #n

This field has the same definition and format as in the ID Page block defined in section 5.6.2. This field is used to identify one of the members of this Group Call. This field is repeated as necessary to identify all of the included identifiers. There must be a one-to-one correspondence of Options Bitmap fields to Identifier fields.

5.12.6 Function Code #n

This optional field has the same definition and format as in the ID Page block defined in section 5.6.1. Due to the nature of a Group Call, the Function Code must be supplied, where appropriate, for the indicated member of this Group Call when the Options Bitmap #n Bit D is set to one.

5.12.7 Expansion Options Bitmap #n

This optional field defines which of the expansion optional identifier-related fields is present for the particular identifier in question. The format for this field is 01ABCDEF where A through F are individual flags indicating which of the identifier-related optional expansion fields are present in relation to Identifier #n. This field is present when the Options Bitmap #n Bit F is set to one. All bit flags are currently reserved with the except of Bit F which is used to indicate an additional Expansion Options Bitmap n+1 field.

Bit Flag	Definition
Bit A	Reserved.
Bit B	Reserved.
Bit C	Reserved.
Bit D	Reserved.
Bit E	Reserved.
Bit F	Indicates inclusion of an Expansion n+1 Option field.

5.12.8 Using Group Paging Blocks

When the destination node sees an ID Page Group Call Address block type, it must then monitor the TNPP network for other ID Page Group Call Address block types, when the Block M of N fields indicate more than one ID Page Group Call Address block type, and a Group Call Data block type with the same Group Call ID number as described in section 5.11.18.

5.13 Group Call Data Fields - Optional Block

The Group Call Data block is an optional TNPP data block. It is utilized only if the destination terminal is programmed to support this block type and the related Cap Page or ID Page Group Call Address block(s) or blocks that must also be received in order to specify a complete Group Call. The purpose of this block is to pass the message content associated with the previously sent list of Cap Codes to the destination TNPP node. These two block types must be processed and broadcast together as a Group Call. However, the order of reception of this block and the associated Group Call Address block or group of blocks is not significant.

There is no provision in this method for older, non-group-call-supporting paging terminals to support any portion of this. Thus older terminals are expected to ignore this new block.

5.13.1 Group Call ID

The Group Call ID is a one-byte field of the format of 01ABCDEF where bits A through F are used to link to the temporary Group Call ID number given in the associated Cap Page or ID Page Group Call Address block or group of blocks. The use of the Source Address, given in the header of this block, with the Group Call ID to form a unique identification number allows the reuse of Group Call IDs across multiple Source Addresses as described in section 5.11.1. This stretches the number of simultaneously-open Group Call transactions to the number of Source Addresses by the number of Group Call IDs.

5.13.2 Message Text

This field is the same as in the CAP page format (section 5.5.7). For the "Data" page class, binary data (as opposed to ASCII printable characters) are placed in this field, subject to the data transparency rules presented in section 4.1.

5.13.3 Using Group Paging Blocks

When the destination node sees a Group Call Data block type, it must monitor the TNPP network for one or more Cap Page or ID Page Group Call Address blocks with the same Group Call ID number originating from the same Source Address as described in section 5.11.18.

5.14 Congestion Control Block

The Congestion Control block is an optional TNPP data block. It is utilized only if the destination terminal is programmed to support this block type. The Congestion Control block defines an optional mechanism to notify the sending node of the congestion level of an encoding node by destination code, RF channel and RF zone combination or combinations. This is in opposition to the current method, which is for the encoding node to. Without the use of congestion control the encoding node normally accepts traffic for channels and zones that are not congested and discards the remainder. With the support of accept traffic for the channel and zone that is not congested and discarding the remainder. The encoding node sends a notification to the sending node via congestion control, the encoding node may alert the sending node through the use of a Congestion Block as specified in the following description.

5.14.1 Block Descriptions

5.14.1.1 Congestion Control Block Format

The format of the Congestion Control block is given in Section 4.4.12 and is repeated here for ease of understanding of the component blocks that make up the Congestion Control block.

Field Name	Number of Bytes	Comments
Block type flag (4A hex)	1	
CSN	1	Congestion Sequence Number.
Object Block (first)	>= 5	Destination type (see 5.14.1.2).
Object Block (intermediate or last)	>= 4	Destination or CZL type (see 5.14.1.2). Always CZL type if last Object Block.
Object Block (intermediate or last)	>= 4	Destination or CZL type (see 5.14.1.2) - Optional. Always CZL type if last Object Block.
	• • •	
ETB (end-of-block flag)	1	

Each Congestion Block is composed of two or more Object Blocks as described in Section 5.14.1.2 - Object Block.

5.14.1.2 Object Block Format

Each Congestion Block is comprised of two or more Object Blocks as shown below.

Field Name	Bits/Fields 7654 3210	Comments
Object Identifier / Object Count	RRRD NNNN	If D = 0, following objects are Destinations; otherwise, following objects are CZLs. NNNN is number of objects minus one to follow. RRR is reserved and should initially be set to 010. Note however that this could change in future revisions to support additional features.
Object (first)		Four hex digit Destination or variable length CZL (three byte minimum).
	• • •	
Object (last)		Optional

Each Object is comprised of either a list of Destinations as defined in Section 5.14.3 - Field Descriptions or a list of CZL Blocks as defined in Section 5.14.1.3 - CZL Block.

5.14.1.3 CZL Block Format

Each CZL Block is composed of CZL Sub-Fields as shown below.

Field Name	Byte Count	Comments
RF Channel	1	
RF Zone	1	
Congestion Level	1	
Encoding Format ID (first)	1	Optional
	• • •	
Encoding Format ID (last)	1	Optional

5.14.2 Congestion Control Block Usage Rules

There are a number of rules associated with the Congestion Block:

1) The first Object Block type in a Congestion Block must be of type Destination.
2) The last Object Block type in a Congestion Block must be of type CZL.
3) All Object Blocks of type Destination refer to all subsequent Object Blocks of type CZL up to the next Object Block of type Destination (if any).
4) Alternating Object Blocks (those of different types) may be included within the same Congestion Block, subject to these rules.
5) Sequential Object Blocks of the same type are treated as if they were a single Object Block which contained more handle Objects than one Object Block may contain.

5.14.3 Field Descriptions

5.14.3.1 Congestion Sequence Number (CSN)

The congestion sequence number, or CSN, is utilized to determine the relative order in which Congestion Blocks are being received at a node which is processing such a block. Nodes which generate Congestion Blocks, assign a monotonically increasing sequence number to each block which is sent. Nodes which receive and process congestion blocks, keep track of the "largest" sequence number received from each source node.

If a "smaller" CSN is received from a particular source code, the Congestion Block is ignored since most current congestion data has already been received. Because of the various multi-nodal paths which an individual packet may traverse while moving between any source and destination nodes, TNPP can not guarantee that packets will arrive at a node in the same order as they were originally transmitted. The CSN value insures that should individual packet routing result in Congestion Blocks arriving out of order, only those blocks with the most up to date information will be processed.

The CSN is a value between 0 and 255 which is represented as a single binary hexadecimal digit. Therefore, Congestion Blocks sent are assigned the sequential numbers 0 to 255 wrapping back to 0 following the value of 255.

The concept of "larger" and "smaller" refers to logically larger or smaller when the wrap around nature of the CSN value is considered. With the premise that the CSN is assigned sequentially and with a wrap around occurring only every 256 Congestion Blocks, is expected that receiving nodes will be able to make the larger/smaller determination, because gross jumps in CSN values are not expected.

5.14.3.2 Object Identifier

The Object Identifier serves to identify the Object type as either a Destination (if set to 0), or as a CZL Block (if set to 1).

5.14.3.3 Object Count

The Object Count serves three functions:

- Delineate Object Blocks.
- Inform the sending node of the number of Objects to expect in the Object Block.
- Prevent an empty Object Block (no Objects).
- The Object Count is in the resultant range of one to sixteen.

This allows multiple destinations to be notified of one or more CZL Indicators.

5.14.3.4 Congested Destination

This is the destination code programmed in the encoding node that is congested. It has the form and values of the destination code as defined in a conventional TNPP block header (refer to Section 5.2.1).

5.14.3.5 RF Channel

This is the RF channel in the encoding node that is congested. It has the form and values of the RF channel as defined in a conventional TNPP cap page block header (refer to Section 5.5.3).

5.14.3.6 RF Zone

This is the RF zone in the encoding node that is congested. It has the form and values of the RF zone as defined in a conventional TNPP cap page block header (refer to Section 5.5.4).

5.14.3.7 Congestion Level

The Congestion Level is defined through the bit fields labeled INABCDEF as shown below.

Field	Meaning
I	If set to 1, encoding format bytes follow this field. See Section 5.14.3.8.
N	Flag bit that means that the Sending node cannot override this Object meaning.
ABCDEF	Congestion Level as defined in the next two tables.

The actual Congestion Level values denote the amount of time the encoding node expects that the channel, zone, and, optionally the Encoding Format ID (Pager Type) to experience congestion.

#	Meaning	#	Meaning	#	Meaning	#	Meaning
0	clear	16	100 sec.	32	420 sec.	48	30 min.
1	5 sec.	17	110 sec.	33	450 sec.	49	33 min.
2	10 sec.	18	120 sec.	34	480 sec.	50	36 min.
3	15 sec.	19	135 sec.	35	510 sec.	51	39 min.
4	20 sec.	20	150 sec.	36	540 sec.	52	42 min.
5	25 sec.	21	165 sec.	37	570 sec.	53	45 min.
6	30 sec.	22	180 sec.	38	600 sec.	54	50 min.
7	35 sec.	23	200 sec.	39	12 min.	55	55 min.
8	40 sec.	24	220 sec.	40	14 min.	56	60 min.
9	45 sec.	25	240 sec.	41	16 min.	57	reserved
10	50 sec.	26	260 sec.	42	18 min.	58	reserved
11	55 sec.	27	280 sec.	43	20 min.	59	reserved
12	60 sec.	28	300 sec.	44	22 min.	60	reserved
13	70 sec.	29	330 sec.	45	24 min.	61	reserved
14	80 sec.	30	360 sec.	46	26 min.	62	main't.
15	90 sec.	31	390 sec.	47	28 min.	63	down

A further explanation from a programmatic standpoint is shown in the next table.

Start Value	End Value	Start Meaning	End Meaning	Increment
0		Clear		N / A
1	11	5 sec.	55 sec.	5 sec.
12	17	60 sec.	110 sec.	10 sec.
18	21	120 sec.	165 sec.	15 sec.
22	27	180 sec.	280 sec.	20 sec.
28	37	300 sec.	570 sec.	30 sec.
38	47	10 min.	28 min.	2 min.
48	52	30 min.	42 min.	3 min.
53	56	45 min.	60 min.	5 min.
57	61	Reserved		N / A
62		Maintenance		N / A
63		Down		N / A

5.14.3.8 Optional Encoding Format ID

This optional byte or bytes define exactly which encoding formats are congested.

It has the form of the Page Type as defined in Section 5.5.1 with one exception: If the high order bit is set, additional Encoding Format IDs immediately follow this byte.

The number of Encoding Format IDs that may be included in any particular CZL Block is limited only by the number of Encoding Format IDs (Pager Types) defined in Section 5.5.1, and the size of the Congestion Block as it relates to the maximum size of a TNPP packet.

If this field is omitted, all encoding formats for this CZL are to be considered congested.

5.14.4 Application Notes

The packet destination address may specify sending the Congestion Block or Blocks to a specific sending node or group of sending nodes.

When a Congestion Block is sent to a group of sending nodes, the encoder can notify many sending nodes of the status of the entire encoding node.

Only a sending node (not an intermediate node) would take action on the congestion block.

5.14.5 Glossary

The following terms are defined within the context of channel congestion control.

Term	Description
Congestion Block	TNPP data block with a ëJí block type defined for congestion control.
Congestion Level	The amount of time A level of RF Channel / RF Zone congestion
CSN	Congestion Sequence Number - The application level mechanism which reduces the probability of receiving order-sensitive information out of order.
CZ	Congested RF Channel / RF Zone.
CZL	CZL Indicator - The application level of the CZL Object
CZL Object	A Congestion Block sub-block that contains RF Channel and RF Zone and Congestion Level information.
DCZ	Destination / Channel / Zone - A composite of specific destination / channel / zone combinations about which congestion is being reported.
Destination (or D)	The destination address at the encoding node which is reporting congestion. This element has no direct relationship to the destination code in the TNPP block header which contains the Congestion Block.
Encoding node	The node terminating or encoding paging traffic and sending congestion control notifications.
Encoding Format ID	If supplied, this is the specific encoding format (Pager Type as defined in section 5.5.1) which is experiencing congestion.
Object Block	A Congestion Block sub-block which may contain either Destination or CZL Objects.
Sending node	The node originating paging traffic and receiving congestion control notifications.

6.0 Optional Features and Capabilities

This section will discuss some special features which have been implemented by certain manufacturers for use within their own network of nodes operating via the TNPP protocol. These "optional features and capabilities" are NOT part of the "official" TNPP specification and do not need to be implemented in a "standard" TNPP implementation. They are presented as part of the specification so that manufacturers are aware of the manner in which other firms have implemented certain capabilities. In addition, the original TNPP specification never detailed the implementation of the Two-Tone paging format. Because of this, two distinct implementations are currently in use in the field. These implementations are also discussed.

6.1 Transparent CRC - Submitted by Glenayre Electronics Inc.

The TNPP protocol utilizes the Cyclic Redundancy error checking mechanism known as CRC-16 (refer to Appendix A). Once the two byte value of CRC-16 is calculated, these bytes are appended to the packet following the ETX character. Data bytes may be any binary 8 bit data. In order to insure that control characters which are sent as data are not confused as TNPP protocol control characters, a transparency insertion mechanism is utilized, as described in section 4.1. This mechanism converts certain control characters into a two character sequence which represents the specific control character.

Unfortunately, according to the specification, this same mechanism of converting binary information to printable ASCII characters does not apply to the two byte CRC-16 value. Therefore, dependent upon the result of the CRC-16 calculations, the two bytes appended to a packet may contain any possible binary configuration.

In TNPP network implementations where direct links are being utilized to connect nodes to each other, this does not represent a problem. But, if TNPP is utilized through packet switched circuits, through statistical multiplexers or through other equipment which utilizes certain ASCII control characters for "flow control" (XON/XOFF is typically utilized), then the CRC-16 values can falsely enable flow control. This can potentially shut down communications by telling modems to stop sending data.

The following mechanism has been employed to convert the two byte CRC-16 value into a four byte field in a similar manner as that utilized in the TNPP header to forward source and destination information. If the transparent CRC feature is activated in an implementation, then the TNPP can be forwarded through other communications equipment without running the risk that data byte will falsely shut down the communications equipment. Of course, for this mechanism to operate correctly, both the sending and receiving equipment must have implemented this transparent CRC feature. If only one side is utilizing this implementation, then all packet transfer will fail.

6.1.1 Transparent CRC Data Format

This field is a four-hexadecimal-digit value that represents the 16 bit CRC-16 value. For example, if the CRC-16 value is hexadecimal 9A4B, the packet will appear as follows:

Normal TNPP	Transparent CRC Enabled
<SOH>	<SOH>
[Packet Header]	[Packet Header]
<STX>	<STX>
Data Blocks]	[Data Blocks]
<ETX>	<ETX>
<Hex 4B>	<ASCII 4>
<Hex 9A>	<ASCII B>
	<ASCII 9>
	<ASCII A>

6.2 Two-Tone Paging Formats

The original version of the TNPP specification did not indicate the manner in TNPP would convey the correct frequency pairs to utilize in order to alert a Two-Tone pager at a remote node in a network. Because no specification had been set, two implementations have been used in the field. These formats are described in this section. If a new TNPP implementation is being created and Two-Tone pagers are to be supported, either of these formats may be used. But, if a TNPP implementation is to forward Two-Tone page requests to another manufacturers implementation, then they must both agree which of the two formats are to be used.

The general format of a Two-Tone pager request has already been specified. What is missing, is how to convey the frequency data, tone duration, and gap duration information which are associated with the particular pager to be alerted. The two formats discussed in this section, each describe a mechanism for forwarding information within the 8 byte Cap Code field of the CAP-Page block.

6.2.1 Two-Tone Paging Format A - Commonwealth/UniPage Format

There are five pieces of information associated with paging a two-tone pager.

- Tone A Duration
- Tone A Frequency
- Inter-Tone Gap Duration
- Tone B Duration
- Tone B Frequency

Under this format data is specified in the following manner:

- Tone frequencies are specified in units of 0.1 Hertz over a frequency range of 40.0 to 3200.0 Hz.
- The A tone duration is specified in units of 100 milliseconds over a range of 0 to 3100 milliseconds.
- The B tone duration is specified in units of 100 milliseconds over a range of 0 to 6300 milliseconds.
- The gap duration is either 0 or 200 milliseconds.

With the data specified in this way:

- frequency is specified as a 15 bit field
- tone A duration is specified as a 5 bit field
- tone B duration is specified as a 6 bit field
- gap duration is specified as a 1 bit field

In keeping with the TNPP requirements of leaving binary fields as ASCII printable data, the 15 bit frequency data is specified as three bytes as follows:

bit: 76543210

01000abc	first byte	(most significant 3 bits)
01defghi	second byte	(middle 6 bits)
01jklmno	third byte	(low order 6 bits)

The Tone A duration is in the following format:

bit: 76543210

01pqrstu

where: p is set to 1 if inter-tone gap is required
qrstu is the 5 bit tone A duration

The Tone B duration is in the following format:

bit: 76543210

01xxxxxx

where: xxxxxx is the 6 bit tone B duration

Given this format, the 8 byte cap code field is specified as follows:

Byte Number	Data
1	Tone A duration plus inter-tone gap flag
2, 3 and 4	Tone A frequency
5	Tone B duration
6, 7 and 8	Tone B frequency

As an example, we assume we wish to specify the following two tone pager:

- 0.4 second tone A
- 368.5 Hz (0E65 hex)
- no inter-tone gap
- 0.8 second tone B
- 979.9 Hz (2647 hex)

The 8 byte cap code field is programmed as:

Byte Number	Breakdown	Data	ASCII	Hex
1	01 p qrstu	01 0 00100	D	44
2	01000 abc	01000 000	@	40
3	01 defghi	01 111001	y	79
4	01 jklmno	01 100101	e	65
5	01 xxxxxx	01 001000	H	48
6	01000 abc	01000 010	B	42
7	01 defghi	01 011001	Y	59
8	01 jklmno	01 000111	G	47

6.2.2 Two-Tone Paging Format B - Glenayre Format

In this two-tone paging format, the Tone A and Tone B frequency values are stored in the 8 byte cap code field as two four digit values. Frequency accuracy is rounded to the nearest decimal integer frequency. The tone duration and inter-tone gap timing values are assumed to be the default values set at the destination paging terminal. As a further option, with coordination of all nodes in the network, the function code field of the CAP PAGE format, may be utilized to represent a unique Tone A and Tone B timing pair which should be utilized at the node which receives this packet.

As an example, the following format is used to denote a two-tone pager where tone A is 787.3Hz and tone B is 801.8Hz:

Byte Number	ASCII Data	
1	0	Tone A - Most Significant Byte
2	7	
3	8	
4	7	
5	0	Tone B - Most Significant Byte
6	8	
7	0	
8	2	

6.3 Voice Paging Via TNPP - Submitted by Commonwealth Communications

A mechanism by which voice pages can be forwarded over the same communication lines which are being used to execute the TNPP protocol, is not specified in the standard TNPP specification. This section describes an implementation which supports analog type voice paging and TNPP digital packets over the same communication links. In order to operate properly, special hardware is required on each of the TNPP links so that the communication lines which are being used, can be connected to the output of a modem (for TNPP digital packets) or connected to the voice input/output subsystem of a particular paging terminal. By sharing one set of communication lines in this way, all types of paging may be supported without increasing communication costs. But, it must be noted that during the time voice pages are being forwarded, a process which can take many seconds, no other pages are being sent. Therefore, overall network load must be able to support the delays which are introduced by mixing voice pages with the TNPP requests which are sent digitally.

Although there may be other possible implementations of voice networking via TNPP, this section describes an implementation which has been operational for some period of time.

6.3.1 Restrictions

The mechanism described in this section is operational under the following restrictions:

- The network must contain a maximum of 15 nodes
- Each of the network nodes must contain two TNPP links. The topology of the network must be such that the nodes form a "string" where the output of node A is the input of node B, the output of Node B is the input of Node C, etc. for all of the nodes in the network. The two nodes at the start and end of the chain perform special functions as far as synchronizing and allocating voice access to intermediate nodes.
- Communications links should be full-duplex with separate transmit and receive paths (four-wire circuits).
- One of the two end nodes in the network must be designated as the MASTER. The other end is known as the SUB-MASTER.

With the special switching hardware installed, any of the TNPP links at a node may be put into a TRANSMIT-VOICE or RECEIVE-VOICE configuration. Any intermediate node which is transmitting voice over the network will simultaneously transmit the audio over the transmit circuits of both of its network ports. Only one node may transmit voice at a time. All other nodes function as a RECEIVE-VOICE node at that time. In this configuration, the audio entering on a receive circuit of the first network port is transmitted in real time on the transmit circuit of the second network port. Conversely, the receive circuits of the second network port are coupled to the transmit circuit of the first network port. Therefore, a node configured as a RECEIVE-VOICE node, becomes a bi-directional voice repeater for all audio in the network. Furthermore, this node can also monitor both of the cross-coupled audio paths for the presence of audio signaling.

6.3.2 Voice Paging Call Progress

When a voice page is to be sent to the network, the originating node initiates a packet using the COMMAND format. (The Command format utilizes manufacturer specific commands which Commonwealth Communications has utilized to perform these specialized functions while staying within the overall framework of the TNPP protocol). This voice initiation request is known as the VOICE-REQUEST. The VOICE-REQUEST propagates through the network until it reaches the node known as the MASTER (as defined in 6.3.1). It is the responsibility of the MASTER to accept VOICE-REQUESTs and process them as necessary.

The MASTER maintains a queue of all VOICE-REQUESTs. The MASTER will provide the coordination (network arbitration) so that only one node forwards voice requests at a time. This arbitration is performed by another COMMAND format packet known as the VOICE-ACK packet.

Upon receipt of a VOICE-ACK packet, the originating node responds by initiating a two-block COMMAND format packet, known as the VOICE-INFO packet. The VOICE-INFO packet contains the encoding parameters of the particular pager along with a list of the destinations to which the voice page is to be directed. The VOICE-INFO packet propagates to "both ends" of the network.

A VOICE-INFO packet which reaches the MASTER or SUB-MASTER node, causes a sequence of events to occur in order to set up the voice path across the network. The same sequence of events occurs from the MASTER as well as from the SUB-MASTER.

Upon receipt of the VOICE-INFO packet, the MASTER examines the list of destinations. The MASTER initiates another COMMAND format packet known as the VOICE-COMMAND. This packet propagates toward the originating node. It contains all of the data in the VOICE-INFO packet. After the VOICE-COMMAND packet is received by the next node in the network, the MASTER configures itself as a VOICE-RECEIVE node.

A node receiving a VOICE-COMMAND packet, examines the destination list, and allocates resources to receive a voice page if that node is on the destination list. In either case, the VOICE-COMMAND packet is forwarded out of the other network port in the direction of the originating node. Eventually, the originating node will receive a VOICE-COMMAND packet initiated from the MASTER and SUB-MASTER. Once both VOICE-COMMAND packets are received, the originating node configures itself as a TRANSMIT-VOICE node and forwards its voice message over the transmit pair of both of its communication links. In this implementation, the voice message is mixed with a low level pilot tone to provide a 'start recording' cue to the various destinations. Termination of this pilot tone causes all network nodes to return to the full-duplex data configuration and begin 'normal' TNPP digital communications.

Default timers run during all states of voice networking. Expiration of these 'guard' timers during the set-up of voice networking, causes the node to return to 'normal' TNPP digital operations and causes a special VOICE-RESTART packet to be transmitted out of both TNPP ports. A node receiving a VOICE-RESTART packet forwards this packet out of its alternate port thereby propagating the restart message through all other nodes.

6.3.3 Packet Formats

This section describes the format of the five COMMAND packets used to support voice paging:

- VOICE-REQUEST
- VOICE-ACK
- VOICE-INFO
- VOICE-COMMAND
- VOICE-RESTART

6.3.3.1 VOICE-REQUEST

Field Name	# of bytes	
SOH (Start of Header)	1	
<Packet Header>	12	(See note)
STX (Start of Text)	1	
Block Type = "C" (Command Format)	1	
Manufacturer ID = "COM"	3	
Command = "VRQ"	3	
Resource Number = "1"	1	
ETX (End of Text)	1	
CRC-16	2	

Note: The destination address of a voice request packet is the MASTER and SUB-MASTER of the network. Theses requests propagate through all nodes until an end node is reached. In the header of the TNPP packet, the destination address is normally set to the source address so that the destination is not located before the packet reaches the end point (MASTER or SUB-MASTER).

6.3.3.2 VOICE-ACK

Field Name	# of bytes
SOH	1
<Packet Header>	12
STX	1
Block Type = "C" (Command Format)	1
Manufacturer ID = "COM"	3
Command = "VAK"	3
ETX (End of Text)	1
CRC-16	2

6.3.3.3 VOICE-INFO

Field Name	# of bytes
SOH	1
<Packet Header>	12
STX	1
Block Type = "C" (Command Format)	1
Manufacturer ID = "COM"	3
Command = "VIP"	3
Resource Number = "1"	1
Pager Type (As in 4.3.3)	1
Pager Class (As in 4.3.3)	1
Function Code (As in 4.3.3)	1
Cap Code (As in 4.3.3)	8
Message Text	<variable>
ETB (End of Block)	1
Block Type = "C" (Command Format)	1
Manufacturer ID = "COM"	3
Command = "VIP"	3
Delimiter = ":"	1
Destination 1 (As in 4.2)	4
Dest 1 RF Channel (As in 4.3.3)	1
Dest 1 RF Zone (As in 4.3.3)	1
Delimiter = ":"	1
Destination 2 (As in 4.2)	4
Dest 2 RF Channel (As in 4.3.3)	1
Dest 2 RF Zone (As in 4.3.3)	1
•	
•	
•	
Delimiter = ":"	1
Destination 'n' (As in 4.2)	4
Dest 'n' RF Channel (As in 4.3.3)	1
Dest 'n' RF Zone (As in 4.3.3)	1
ETB (End of Block)	1
ETX (End of Text)	1
CRC-16	2

6.3.3.4 VOICE-COMMAND

Field Name	# of bytes
SOH	1
<Packet Header>	12
STX	1
Block Type = "C" (Command Format)	1
Manufacturer ID = "COM"	3
Command = "CMD"	3
Resource Number = "1"	1
Pager Type (As in 4.3.3)	1
Pager Class (As in 4.3.3)	1
Function Code (As in 4.3.3)	1
Cap Code (As in 4.3.3)	8
Message Text	<variable>
ETB (End of Block)	1
Block Type = "C" (Command Format)	1
Manufacturer ID = "COM"	3
Command = "CMD"	3
Delimiter = ":"	1
Destination 1 (As in 4.2)	4
Dest 1 RF Channel (As in 4.3.3)	1
Dest 1 RF Zone (As in 4.3.3)	1
Delimiter = ":"	1
Destination 2 (As in 4.2)	4
Dest 2 RF Channel (As in 4.3.3)	1
Dest 2 RF Zone (As in 4.3.3)	1
•	
•	
•	
Delimiter = ":"	1
Destination 'n' (As in 4.2)	4
Dest 'n' RF Channel (As in 4.3.3)	1
Dest 'n' RF Zone (As in 4.3.3)	1
ETB (End of Block)	1
ETX (End of Text)	1
CRC-16	2

6.3.3.5 VOICE-RESTART

Field Name	# of bytes
SOH	1
<Packet Header>	12
STX	1
Block Type = "C" (Command Format)	1
Manufacturer ID = "COM"	3
Command = "RST"	3
ETX (End of Text)	1
CRC-16	2

6.3.4 Implementation Notes

This section contains additional notes regarding this implementation.

- For 15 seconds following the receipt of a VOICE-RESTART packet, no other voice packets will be initiated. Receipt of any voice packet other than another VOICE-RESTART will be responded to with a CANCEL.

- A guard timer is started upon forwarding a VOICE-REQUEST packet. This timer is canceled upon receipt of a VOICE-ACK. If the timer expires, a VOICE-RESTART is issued. The default timer value is 10 minutes.

- The MASTER will start a guard timer upon generating a VOICE-ACK packet. This timer is cleared upon receipt of a VOICE-INFO packet. If the timer expires, a VOICE-RESTART is issued. The default timer value is 3 minutes.

- A guard timer is set after generating a VOICE-INFO packet. This timer is cleared upon receipt of a VOICE-COMMAND packet. If the timer expires, a VOICE-RESTART is issued. The default timer value is 3 minutes.

- The MASTER and SUB-MASTER initiate a guard timer after generating the VOICE-COMMAND packet. The timer is canceled upon successful completion of voice network paging. Expiration of this timer causes a VOICE-RESTART packet to be transmitted. The default timer value is 3 minutes.

- All nodes start a guard timer upon entering the RECEIVE VOICE node configuration. Expiration of this timer causes a return to 'normal' TNPP digital data transmission. This timer is cleared upon the completion of a voice network page. The default timer value is 3 minutes.

- Nodes in the RECEIVE VOICE configuration are essentially "Standing-By" waiting to record a voice message. The node configured in the TRANSMIT VOICE mode mixes a 2805 Hz pilot tone with the voice message. This pilot tone indicates to the receivers that voice is present. When the pilot tone is no longer heard after if was already present, this is an indication that voice page transmission is completed. The node returns to 'normal' TNPP digital data transmission mode.

- Only one VOICE-REQUEST may be pending from a single network node at a time.

. If a VOICE-RESTART packet is received, the queue of VOICE REQUESTs is cleared. All outstanding VOICE REQUESTs must be re-issued.

. While in the RECEIVE VOICE configuration, each node should level correct the two cross-coupled audio paths.

. Upon exiting a RECEIVE VOICE or TRANSMIT VOICE configuration, the node returns to TNPP State 0, the link initiation state (see Appendix B)

6.4 TNPP Dial In/Out

TNPP using dial-up circuits is exactly like TNPP using dedicated circuits. The key to efficient implementation of dial in/out TNPP is that the nodes involved remain in the initialization state 0 (see Appendix B) until a physical connection is made.

After link initialization and subsequent exchange of the zero packet, both nodes "know" the node address of the other by virtue of the source address in the exchanged zero-packets. This allows both nodes to route TNPP traffic regardless of which node initiated the connection. Traffic need not be limited only to those packets addressed to the immediately connected node, but, may include packets addressed to nodes which are determined (via internal routing tables)to be reachable through the connected node. Traffic already enqueued and not delivered to the connected node, may be forwarded upon establishment of a connection by either node in the pair.

6.4.1 Dial In/Out TNPP State Tables

Dial In/Out TNPP may be implemented by modifying the Appendix B state tables as shown in this section. Note the addition of State 0a between State 0 and 1.

In the event that the dialing out action in State 0a is unsuccessful, a delay of 30 to 60 seconds, randomly selected, is suggested. This should preclude two nodes with traffic queued, for each other from falling into a synchronous loop of dialing each other simultaneously and continually receiving a busy signal.

State: 0 Name: Initialization State

Actions:

1) set next packet serial number to zero.
2) clear receive serial number table.
3) queue empty packet for next transmission with destination set to 0000.
4) go to State 0a

State: 0a Name: AWAIT connect state

Input:	State:	Response:	Description:
Carrier Detected	1	ENQ	Connected
Packets to Send	0a	---	Dial Out

State: 1 Name: Await ENQ RESPONSE (modified)

Input:	State:	Response:	Description:
(add to State 1 table) no carrier	0	on hook	connection lost
(change to State 1 table) C_{enq}	0	on hook	remote failed to respond, log error and disconnect

State: 2 Name: READY to communicate (modified)

Input:	State:	Response:	Description:
(add to State 2 table) no carrier	0	on hook	connection lost
(change to State 2 table) t_{idle}	0	on hook	no traffic , disconnect

State: 3 Name: TRANSMITting (modified)

Input:	State:	Response:	Description:
(add to State 3 table) no carrier	0	on hook	connection lost

State: 4 Name: awaiting TRANSMIT response (modified)

Input:	State:	Response:	Description:
(add to State 4 table) no carrier	0	on hook	connection lost

6.5 Transmitting FLEX™ page requests from systems where FLEX™ is not supported.

The FLEX™ page type (designator "F" in section 5.5.1) was introduced with revision 3.6 of the specification. If the FLEX™ format is not supported within a system, then the data base for that system and TNPP packets sent from that system are probably not capable of sending page requests with the "F" designator.

Because many carriers have expressed a need to send FLEX™ page requests from systems which do not support FLEX™ themselves, a "standard" has been defined which does not require the software of the source paging terminal to be modified. If paging terminals which do not support FLEX™ adhere to the mechanisms described in this section, then they will be capable of sending "Phantom" FLEX™ paging requests. Destination paging terminals which support FLEX™ should be capable of recognizing these Phantom requests and will transmit them as FLEX™ pages. This mechanism can be utilized at a source paging terminal until its software is updated to send the "F" pager type.

Either of two different mechanisms may be used at the source terminal to send a Phantom FLEX™ page. The selection of the mechanism which is to be employed is a function of possible restrictions in the source paging terminal and a restriction of the destination paging terminal if it has only been programmed to support one of these mechanisms.

6.5.1 FLEX™ Phantom - Mechanism 1

FLEX™ pagers are programmed into the paging systems data base as any speed POCSAG pager (512, 1200 or 2400 baud). The FLEX™ pager address is offset by 3,000,000 (3 million is added to the address), to form an invalid POCSAG address.

Note: Some source paging terminals may require a simple patch to allow invalid POCSAG numbers to be programmed into the data base. Those terminals which can not program invalid numbers at all, may employ Mechanism 2.

When a paging terminal with FLEX™ Phantom format software support installed receives a POCSAG request with a CAP code value above 3 million, it shall subtract 3 million from this value and use the resulting value as the FLEX™ pager number. The POCSAG pager type is ignored and the request is converted to a FLEX™ pager type.

Under this mechanism a subset of the full FLEX™ number range is supported. Within a 7 digit POCSAG pager address, values from 3,000,000 to 9,999,999 will support 7 million FLEX™ pagers.

6.5.2 FLEX™ Phantom - Mechanism 2

This mechanism may be employed when invalid POCSAG addresses can not be programmed into a customer data base. Under this mechanism, specific TNPP Destination Address, RF Channel and RF Zone combinations are specially programmed at the destination paging terminal as "FLEX™ paging zones". When a POCSAG type page request sent from a source paging terminal is received at a FLEX™ paging zone, it is treated as a FLEX™ page rather than a POCSAG page. The POCSAG pager number is treated as a FLEX™ pager number. This mechanism can be used to alert FLEX™ pagers numbered from 1 to 2,031,614 (the range of valid POCSAG address values). Under this mechanism the source paging terminal only needs to specify that POCSAG page requests should be sent to the specific TNPP destination address/channel/zone combination which transmits POCSAG requests as FLEX™ pages.

FLEX™ pager numbers above 2,100,000 can be sent in a similar way by directing these pager numbers to a destination paging terminal TNPP address, RF Channel and RF Zone combination which has been designated as a "FLEX™ paging zone/Nationwide Address". The CAP Code of pages directed to this type of zone, should be reduced by 2,100,000 before they are stored in the data base of the source paging terminal. When a POCSAG page is received at this type of zone, 2,100,000 is added to the POCSAG address and a FLEX™ page for the resultant address is transmitted.

6.6 Increase of Packet Size to 4096 Bytes

The standard TNPP packet is never larger than 1024 bytes. But, in order to handle larger message sizes without fragmenting data into a sequence of individual packets, some implementations have been upgraded to allow for packet sizes of up to 4096 bytes. Like the standard packet, the entire packet size from the SOH to the last byte of CRC, is never larger than the maximum packet size.

There is no methodology in place to automatically determine if 1K or 4K maximum packet sizes are being supported. It is only via the manual configuration of the systems on either side of a link, that the maximum packet size may be allowed to be up to 4096 bytes. If a link supporting 1024 byte packets is connected to one supporting 4096 byte packets, the links will continue to communicate until the link supporting packets larger than 1K in size, sends a packet larger than 1024 bytes. The standard TNPP will not be able to receive the larger size packet.

Appendix A - CRC-16 Process and Code Example

A.1 CRC-16 Calculation and Use

This section describes a method of implementing the algorithm for computing the CRC-16 Block Check Code (BCC) used for error detection in this protocol. The CRC-16 polynomial used for the BCC is:

$$Q(y) = X16 + X15 + X2 + 1$$

The CRC is calculated on each packet starting with the SOH and ending with the ETX. The resulting sixteen-bit code is then split into two bytes and transmitted after the ETX with the least significant byte first.

At the receiver, the CRC is calculated on the entire packet starting with the SOH and ending with the last byte of the transmitted CRC. If there were no errors in the packet data stream, the resulting CRC will be zero. If there were errors in the data stream, there is a very high probability that the resulting CRC will be non-zero.

The following procedure is a simple and fast method of performing the CRC calculation by table lookup:

1) At the beginning of each packet, set the CRC to 0.
2) Fetch the first byte of the packet
3) Exclusive OR the fetched byte with the low-order byte of the CRC to obtain a byte "V".
4) Right shift the CRC eight bits with the high-order bits being zero-filled.
5) Using "V" as an index, fetch a 16 bit unsigned integer from the table (below) and exclusive OR it into the shifted CRC.
6) The result of the exclusive OR operation is the new value of the CRC.
7) If any more unprocessed bytes are left in the packet, fetch next unprocessed byte and go to step 3.

A detailed description of this procedure can be found in CRC Calculation, PC TECH JOURNAL, Vol. 3, No. 4, April 1985, page 118. A copy can be obtained for $7.00 from: PC TECH JOURNAL, Box CN, 1914, Morristown, NJ 07960.

The procedure above uses the following table of 16 bit unsigned hexadecimal numbers in computing the CRC-16:

```
00000,  0C0C1,  0C181,  00140,  0C301,  003C0,  00280,  0C241
0C601,  006C0,  00780,  0C741,  00500,  0C5C1,  0C481,  00440
0CC01,  00CC0,  00D80,  0CD41,  00F00,  0CFC1,  0CE81,  00E40
00A00,  0CAC1,  0CB81,  00B40,  0C901,  009C0,  00880,  0C841
0D801,  018C0,  01980,  0D941,  01B00,  0DBC1,  0DA81,  01A40
01E00,  0DEC1,  0DF81,  01F40,  0DD01,  01DC0,  01C80,  0DC41
01400,  0D4C1,  0D581,  01540,  0D701,  017C0,  01680,  0D641
0D201,  012C0,  01380,  0D341,  01100,  0D1C1,  0D081,  01040
0F001,  030C0,  03180,  0F141,  03300,  0F3C1,  0F281,  03240
03600,  0F6C1,  0F781,  03740,  0F501,  035C0,  03480,  0F441
03C00,  0FCC1,  0FD81,  03D40,  0FF01,  03FC0,  03E80,  0FE41
0FA01,  03AC0,  03B80,  0FB41,  03900,  0F9C1,  0F881,  03840
02800,  0E8C1,  0E981,  02940,  0EB01,  02BC0,  02A80,  0EA41
0EE01,  02EC0,  02F80,  0EF41,  02D00,  0EDC1,  0EC81,  02C40
0E401,  024C0,  02580,  0E541,  02700,  0E7C1,  0E681,  02640
02200,  0E2C1,  0E381,  02340,  0E101,  021C0,  02080,  0E041
0A001,  060C0,  06180,  0A141,  06300,  0A3C1,  0A281,  06240
06600,  0A6C1,  0A781,  06740,  0A501,  065C0,  06480,  0A441
06C00,  0ACC1,  0AD81,  06D40,  0AF01,  06FC0,  06E80,  0AE41
0AA01,  06AC0,  06B80,  0AB41,  06900,  0A9C1,  0A881,  06840
07800,  0B8C1,  0B981,  07940,  0BB01,  07BC0,  07A80,  0BA41
0BE01,  07EC0,  07F80,  0BF41,  07D00,  0BDC1,  0BC81,  07C40
0B401,  074C0,  07580,  0B541,  07700,  0B7C1,  0B681,  07640
07200,  0B2C1,  0B381,  07340,  0B101,  071C0,  07080,  0B041
05000,  090C1,  09181,  05140,  09301,  053C0,  05280,  09241
09601,  056C0,  05780,  09741,  05500,  095C1,  09481,  05440
09C01,  05CC0,  05D80,  09D41,  05F00,  09FC1,  09E81,  05E40
05A00,  09AC1,  09B81,  05B40,  09901,  059C0,  05880,  09841
08801,  048C0,  04980,  08941,  04B00,  08BC1,  08A81,  04A40
04E00,  08EC1,  08F81,  04F40,  08D01,  04DC0,  04C80,  08C41
04400,  084C1,  08581,  04540,  08701,  047C0,  04680,  08641
08201,  042C0,  04380,  08341,  04100,  081C1,  08081,  04040
```

The following numbers are the column totals for table verification:

```
0FE010, 0FF810, 101010, 100810, 100010, 101810, 0FF010, 0FE810
```

NOTE: Entries in this table are used left to right row by row. In other words Entry 0 is 00000, entry 1 is 0C0C1, entry 2 is 0C181, entry 8 is 0C601 and entry 255 is 04040.

A.2 Sample Packet with CRC-16 BCC Included

The following list of byte values represents a packet with the BCC included:

Hex:	Character:	Meaning:
01	SOH	start of packet
30	0	destination address
30	0	"
30	0	"
31	1	"
30	0	inertia value
39	9	"
30	0	source address
30	0	"
30	0	"
32	2	"
30	0	packet serial number
31	1	"
02	STX	start of data section
44	D	data block
41	A	"
54	T	"
41	A	"
03	ETX	end of data section
AA	BCC1	low byte of CRC-16 code
AC	BCC2	high byte of CRC-16 code

This data stream will result in a zero CRC when processed through a correctly coded CRC-16 generator.

Appendix B - Node to Node Communications States

These state transition tables describe the mechanism which is used for adjacent-node communications. Each state description contains a list of possible events that can stimulate an action (input), the action that results (response), and the new state after the action is taken (state). The communications process will start with state 0 whenever a node is initialized at power-on or at reset.

Legend (special symbols):

<good>	packet received without error
<bad>	packet received with CRC error
<ugly>	packet received without error, cannot process
<no.q>	packet received without error, no queue space
<rdy>	packet ready for transmit
t_{nrx}	either t_{nri} or t_{nrb} expires
SOH...	begin transmitting packet
<xmt done>	packet transmission has completed

NOTE: In all states, when t_{hold} times out, Re-Queue packet

State: 0 Name: INITialization state

Actions:

1) set next packet serial number to zero
2) clear receive serial number table
3) queue empty packet for next transmission with destination set to 0000.
4) send ENQ flag
5) go to state 1

State: 1 Name: Await ENQ RESPONSE state

Input:	State:	Response:	Description:
ENQ	1	EOT	other end attempting resync also
EOT	2	---	other end responding to ENQ (reset count of ENQ time-outs) (start up t_{idle} timer)
ACK	1	---	unexpected event
NAK	1	---	unexpected event
CAN	1	---	unexpected event
RS	1	---	unexpected event
<good>	1	ACK	packet rcvd OK, acknowledge
<bad>	1	NAK	packet rcvd CRC bad, reject
<ugly>	1	CAN	packet rcvd cannot process
<no.q>	1	RS	packet rcvd no queue space (note 3)
t_{ict}	1	---	unexpected event
t_{nrx}	-	---	not applicable
C_{retry}	-	---	not applicable
t_{nre}	1	ENQ	no response to ENQ, try again (count as ENQ time-out)
C_{enq}	1	ENQ	log error and try again (reset count of ENQ time-outs)
C_{hold}	-	---	not applicable
<rdy>	-	---	queue packets ready for transmission until READY state

State: 2 Name: READY to communicate state

Input:	State:	Response:	Description:
ENQ	2	EOT	other testing link
EOT	2	---	unexpected event - ignore
ACK	2	---	unexpected event - ignore
NAK	2	---	unexpected event - ignore
CAN	2	---	unexpected event - ignore
RS	2	---	unexpected event - ignore
<good>	2	ACK	packet rcvd OK, acknowledge (note 2), restart t_{idle} timer
<bad>	2	NAK	packet rcvd CRC bad, reject, restart t_{idle} timer
<ugly>	2	CAN	packet rcvd cannot process restart t_{idle} timer
<no.q>	2	RS	packet rcvd no queue space (note 3), restart t_{idle} timer
t_{ict}	1	ENQ	packet started, ETX lost, resync
t_{nrx}	-	---	not applicable
C_{retry}	-	---	not applicable
t_{nre}	-	---	not applicable
C_{enq}	-	---	not applicable
C_{hold}	-	---	not applicable
<rdy>	3	SOH...	packet ready to send begin transmission
t_{idle}	1	ENQ	link integrity test

State: 3 Name: TRANSMITting state

Input:	State:	Response:	Description:
ENQ	3	(EOT)	remote system checking link status
EOT	1	(ENQ)	unexpected event
ACK	1	(ENQ)	unexpected event
NAK	1	(ENQ)	unexpected event
CAN	1	(ENQ)	unexpected event
RS	1	(ENQ)	unexpected event
<good>	3	(ACK)	packet rcvd OK, queue an ACK (note 1,2)
<bad>	3	(NAK)	packet rcvd bad, queue a NAK (note 1)
<ugly>	3	(CAN)	cannot process, queue a CAN (note 1)
<no.q>	3	(RS)	no queue space, queue an RS (note 1,3)
t_{ict}	3	(NAK)	packet ETX lost, queue a NAK (note 1)
t_{nrx}	-	---	not applicable
C_{retry}	-	---	not applicable
t_{nre}	-	---	not applicable
C_{enq}	-	---	not applicable
C_{hold}	-	---	not applicable
<rdy>	-	---	queue packets ready for transmission until READY state
<xmt done>	4	---	start t_{nrx} timer

State: 4 Name: awaiting TRANSMIT RESPONSE state

Input:	State:	Response:	Description:
ENQ	4	EOT	remote system checking link status
EOT	1	ENQ	unexpected event
ACK	2	---	packet received by other end OK
NAK	3	---	start packet retransmission
CAN	2	---	reroute or discard, log error
RS	2	---	tag packet for later xmit, requeue
<good>	4	ACK	incoming packet rcvd OK, send ACK (note 2)
<bad>	4	NAK	incoming packet rcvd bad, reject
<ugly>	4	CAN	incoming packet, cannot process, cancel
<no.q>	4	RS	incoming packet, no queue space (note 3)
t_{ict}	4	NAK	incoming packet lost ETX, reject
t_{nrx}	1	ENQ	loss of communications, resync
C_{retry}	2	---	retry limit exceeded, log error and discard
t_{nre}	-	---	not applicable
C_{enq}	-	---	not applicable
C_{hold}	2	---	packet held too long, log error and discard
<rdy>	-	---	queue packets ready for transmission until READY state

NOTES:

1) If a response to a packet cannot be sent due to outbound traffic, the response is held until the end of the outbound packet and is transmitted immediately after the CRC bytes.

2) When a packet is received without error, the following steps are taken:

   a) if the packet serial number is zero, the receive serial number table is cleared,
   b) if the packet serial number is in the receive serial number table, the packet is discarded as a duplicate,
   c) the packet serial number is added to the receive serial number table,
   d) an ACK is sent in response to the packet. If the packet contains a source or destination address which is not considered valid at this node, a CAN response may be sent in lieu of an ACK to indicate that the packet will not be processed at this node. The serial number of this packet remains in the serial number table.

e) If the packet contains a ETE request and this node is the addressed destination, an ETE response is queued for transmission back to the originator.

3) If the packet is not a duplicate (in the receive serial number table) and there is no space available in the destination queue, an RS is sent to cause the sender to requeue the packet for later transmission and the received packet would be discarded. This test would typically occur immediately before step d) above.

Appendix C - Figures and Illustrations

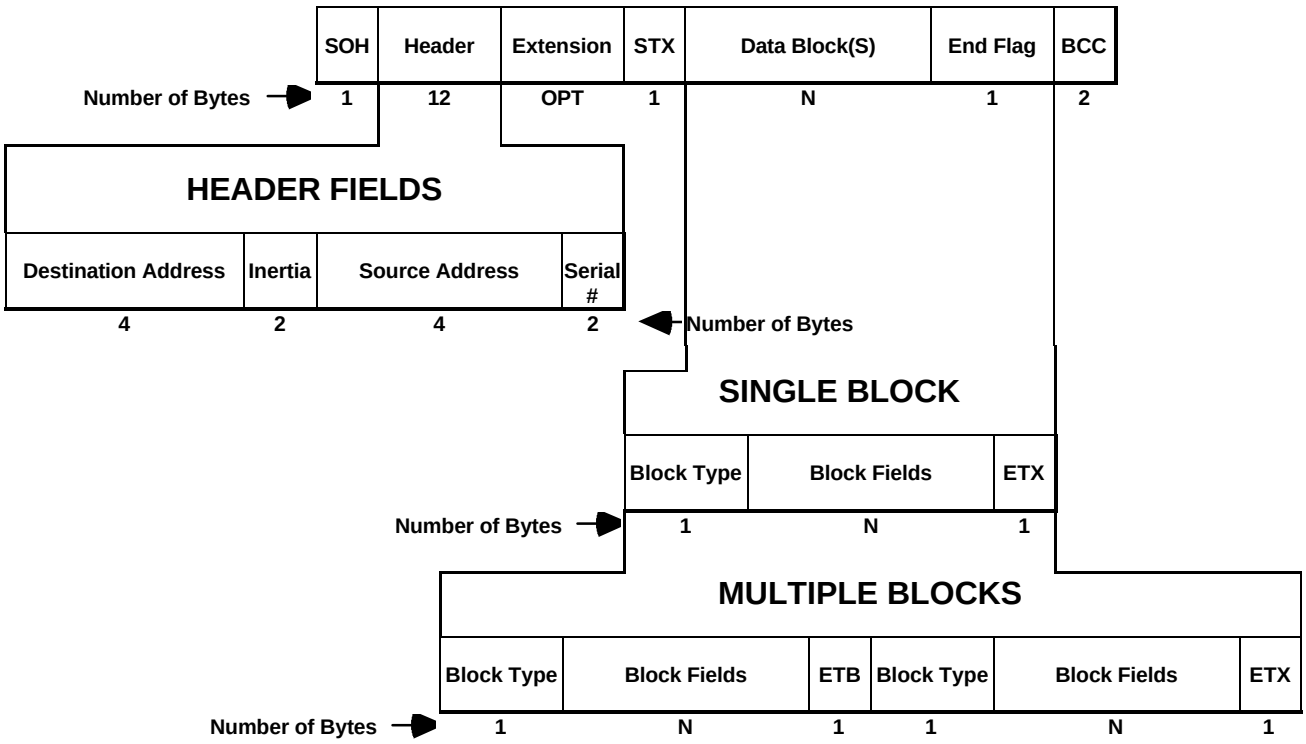

Figure 1 GENERAL DATA STRUCTURE

Notes:

1. **N - Indicates the number of bytes in the remaining data block fields.**
2. **A packet must not exceed 1024 bytes including all control flags and CRC.**
3. **OPT - indicates optional data bytes.**

MS LS CHAR CHAR	0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
0 0000	NUL	DLE	SP Space	0	@ 'At' Sign	P	` Single Left Quote	p
1 0001	SOH	DC1	! Exclamation Point	1	A	Q	a	q
2 0010	STX	DC2	" Double Quote	2	B	R	b	r
3 0011	ETX	DC3	# Pound or Number Sign	3	C	S	c	s
4 0100	EOT	DC4	$ Dollar Sign	4	D	T	d	t
5 0101	ENQ	NAK	% Percent Sign	5	E	U	e	u
6 0110	ACK	SYN	& Ampersand	6	F	V	f	v
7 0111	BEL	ETB	' Apostrophe	7	G	W	g	w
8 1000	BS	CAN	(Open Parenthesis	8	H	X	h	x
9 1001	HT	EM	) Close Parenthesis	9	I	Y	i	y
A 1010	LF	SUB	* Asterisk	: Colon	J	Z	j	z
B 1011	VT	ESC	+ Plus Sign	; Semi-Colon	K	[Open Bracket	k	{ Left Curly Brace
C 1100	FF	FS	, Comma	< Less Than	L	\ Back Slash	l	| Vertical Bar
D 1101	CR	GS	- Minus Sign	= Equal Sign	M	] Close Bracket	m	} Right Curly Brace
E 1110	SO	RS	. Period	> Greater Than	N	^ Carrot	n	~ Tilde Accent
F 1111	SI	US	/ Forward Slash	? Question Mark	O	_ Under Score	o	DEL

Figure 2 CAST OF CHARACTERS (ASCII CODE TABLE)

(Both in character format and description, where appropriate)

CHARACTER	HEX VALUE	USAGE
NUL	00	Reserved - not currently defined
SOH	01	Start of packet
STX	02	Start of data block section of packet
ETX	03	End of packet, two byte CRC-16 follows
EOT	04	Response to link test (ENQ, enquiry)
ENQ	05	Link Test Character
ACK	06	Positive Acknowledgment. Indicates packet received OK.
DLE	10	Reserved - not currently defined
DC1	11	Reserved - not currently defined
DC2	12	Reserved - not currently defined
DC3	13	Reserved - not currently defined
DC4	14	Reserved - not currently defined
NAK	15	Negative Acknowledgment. Indicates BCC checked bad in last packet sent
SYN	16	Reserved - not currently defined
ETB	17	End of data block within packet
CAN	18	Cancel packet. Indicates last packet sent cannot be processed
EM	19	Reserved - not currently defined
SUB	1A	Used to flag control characters in message fields
RS	1E	Hold and re-transmit previous packet t_{hold} later
N/A	FF	Reserved - not currently defined

Figure 3 RESERVED FLAG BYTES AND THEIR USAGE

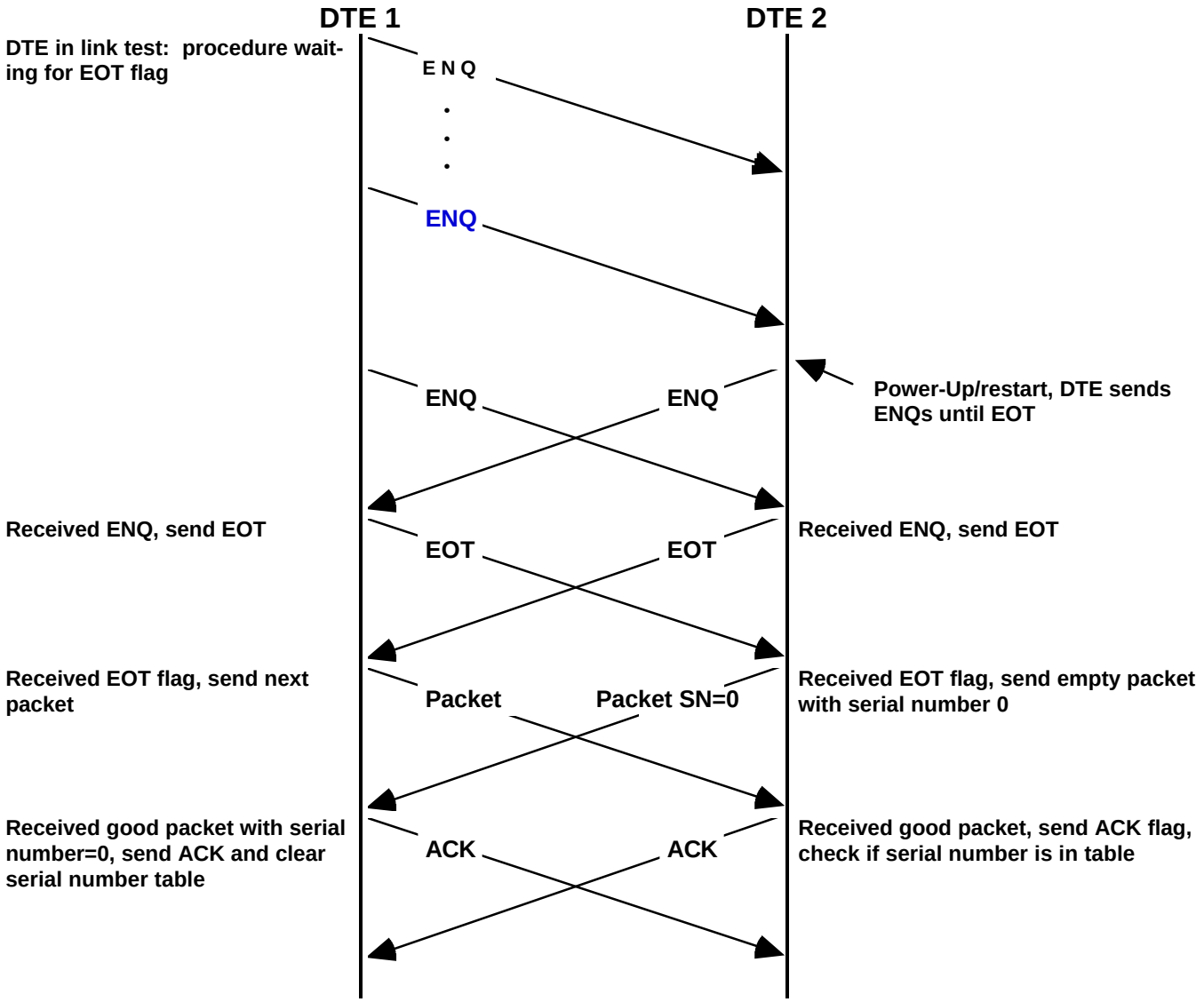

Figure 4 POWER-UP/RESTART SEQUENCE

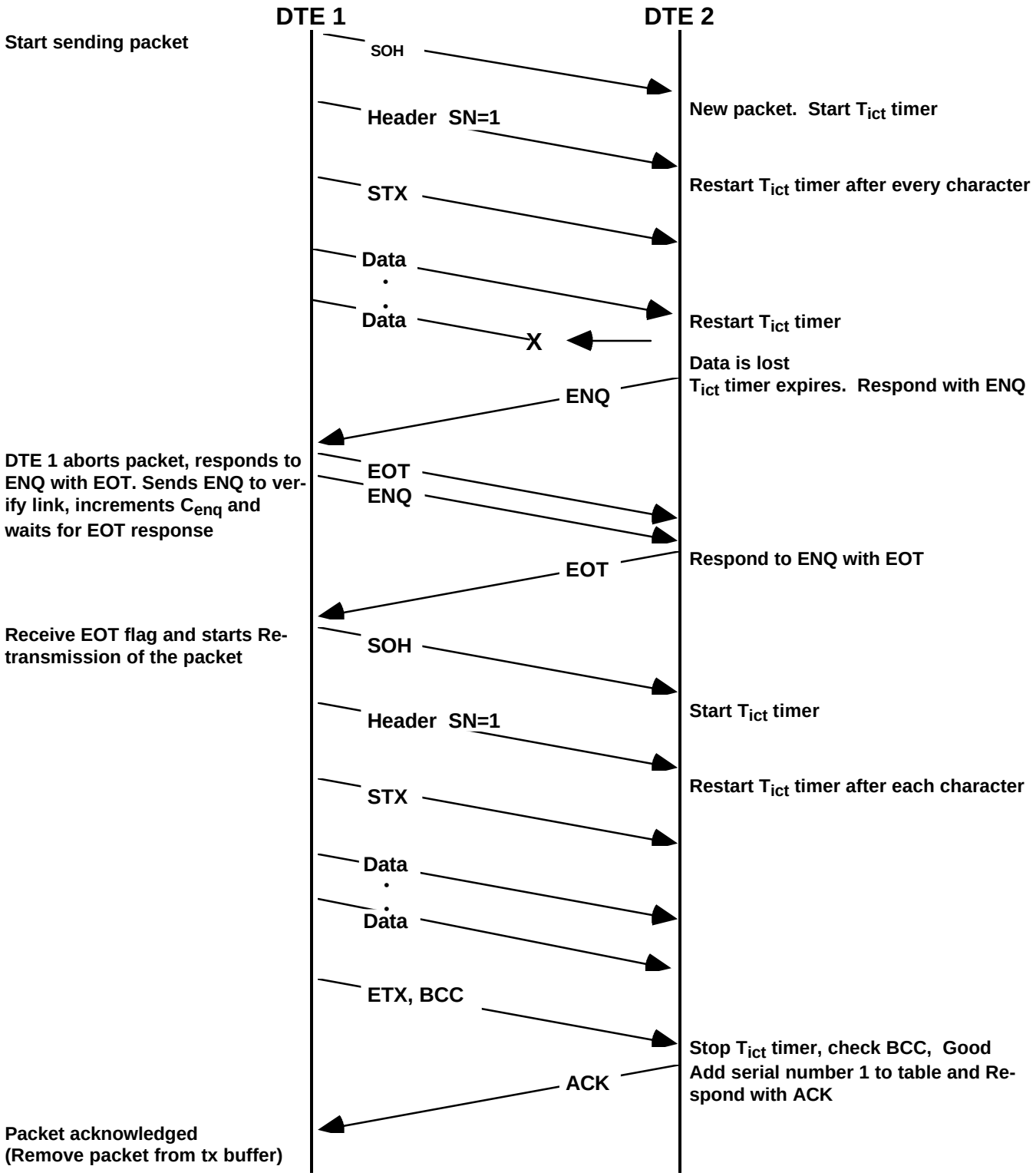

Figure 5 T_{ict} TIME-OUT OR ENQ RECEIVED DURING PACKET TRANSMISSION

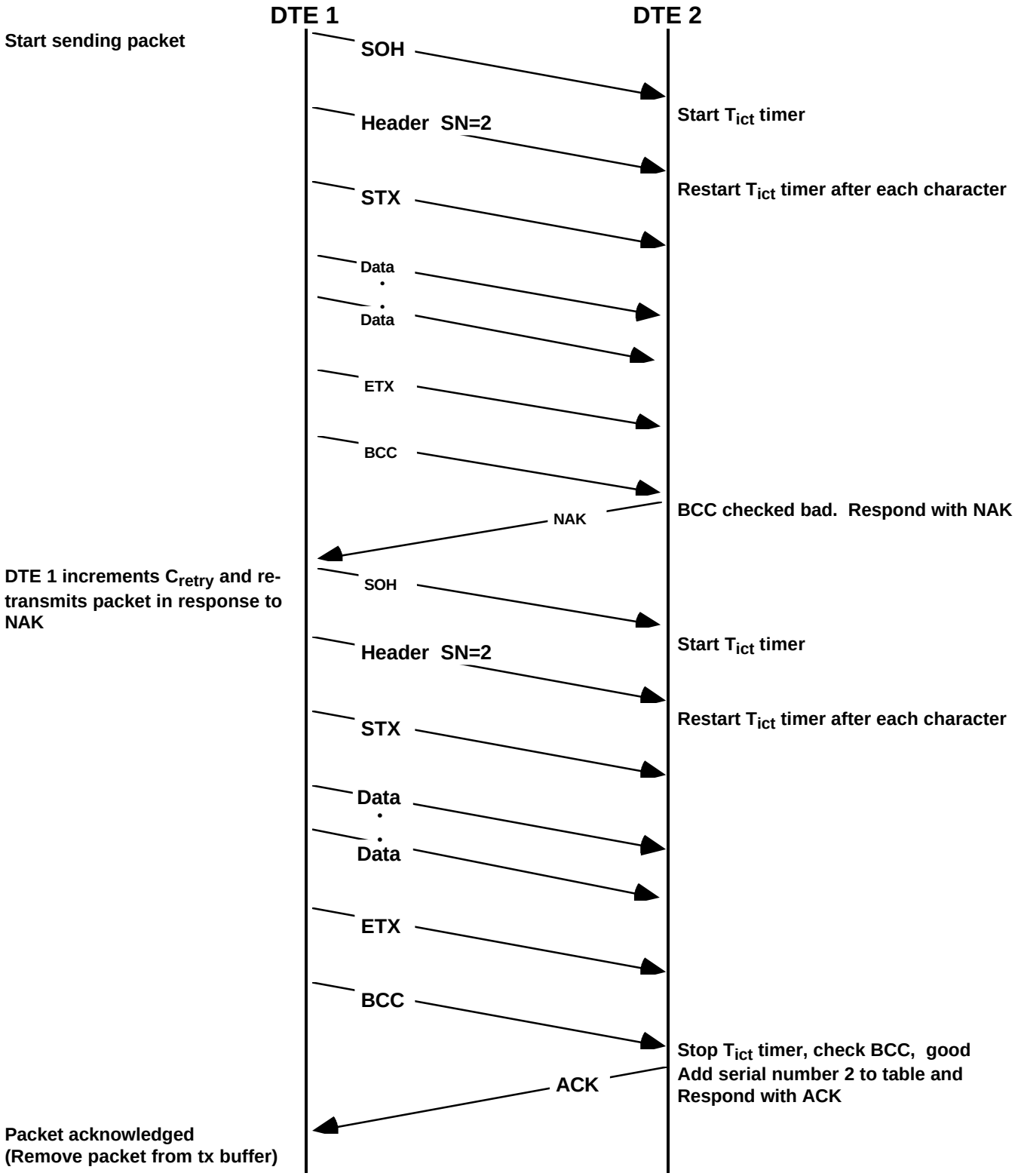

Figure 6 NAK RECEIVED AFTER PACKET IS SENT

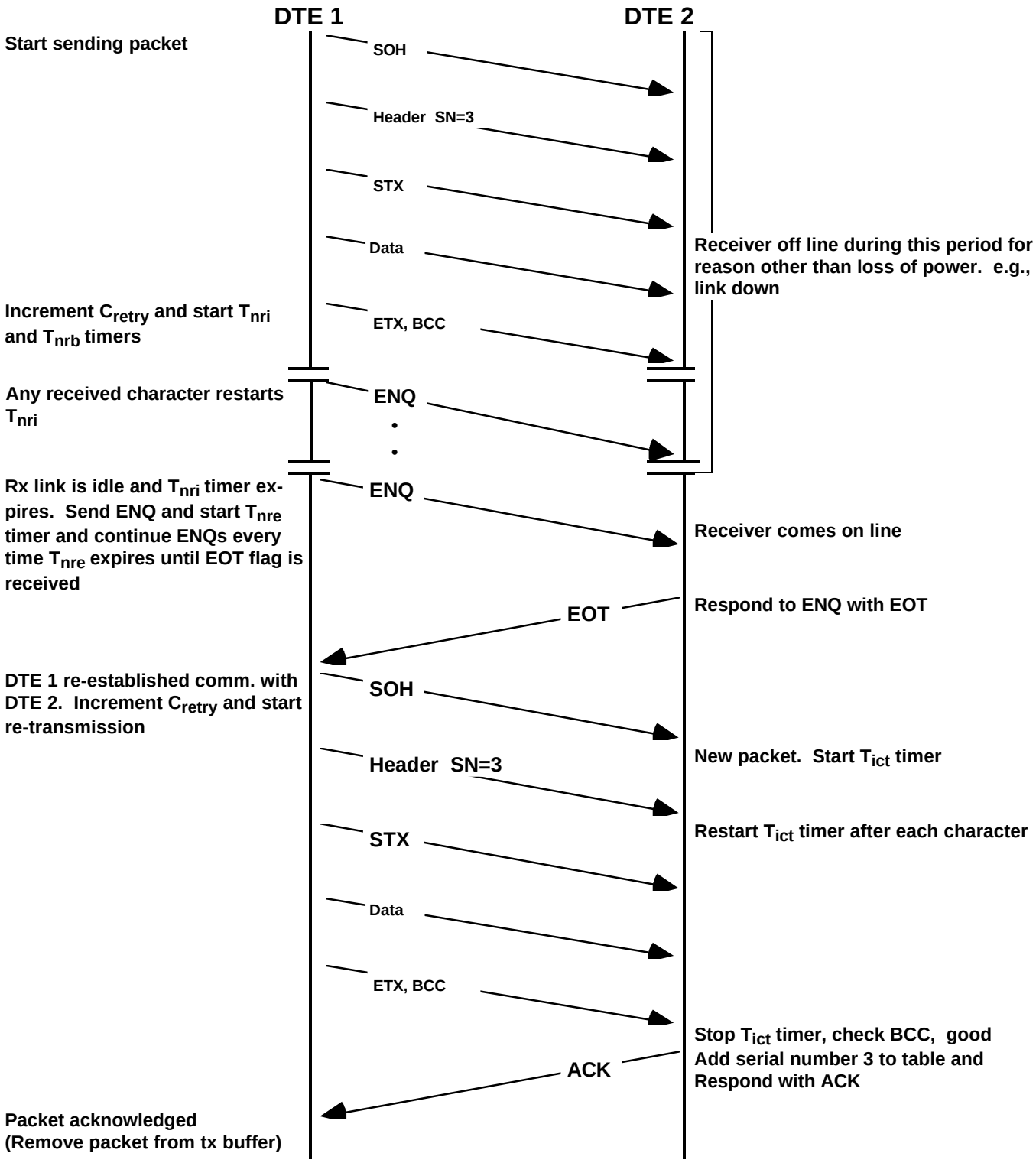

Figure 7 NO ACKNOWLEDGMENT RECEIVED BEFORE T_{nri} EXPIRES

(Rx Link Idle)

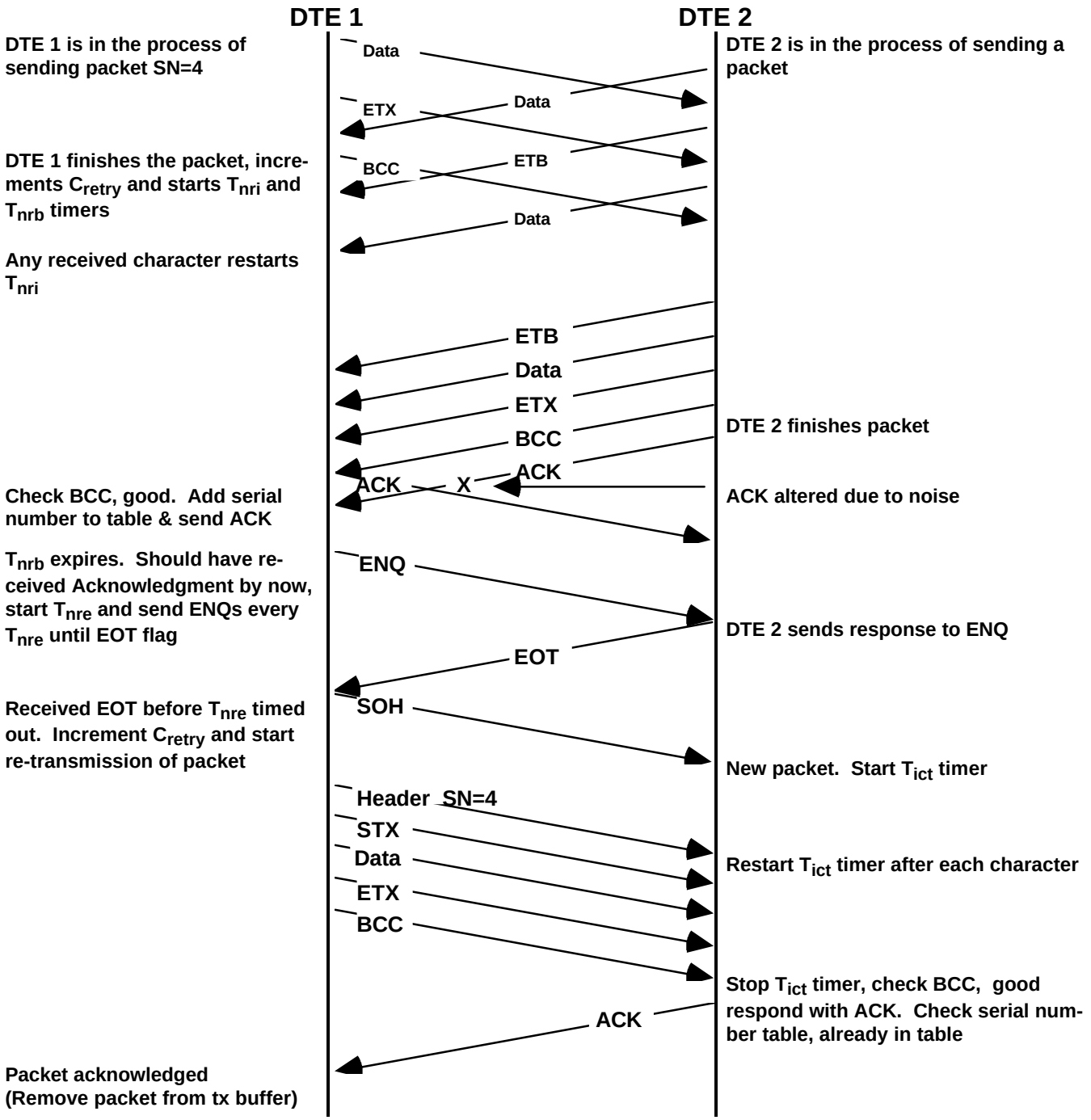

Figure 8 NO ACKNOWLEDGMENT RECEIVED BEFORE T_{nrb} EXPIRES

(RX LINK IS BUSY)

FLEX™-Augmented 12+5 Bytes

ABCDEFGHIJ	**K**	**L**	**M**	**N**	**O**	**PQ+**

'A-J' Bytes	Full FLEX™ Address
'K' Byte	Collapse Information with Optional M Field Enabler Flag
'L' Byte	Phase Information and Options Bits Flags with Optional N, O, and PQ Fields Enabler Flags
'M' Byte (Optional)	Base Frame Number and Frame Offset Enabler
'N' Byte (Optional)	RF Channel Information
'O' Byte (Optional)	Content - Information Blocking Length or (Context Switched) Content - Enhanced Fragmentation Rules
'PQ+' Bytes (Optional)	Expansion Options Bits Flags with Length to End of Expansion

Figure 9 FLEX™-AUGMENTED CAP CODE

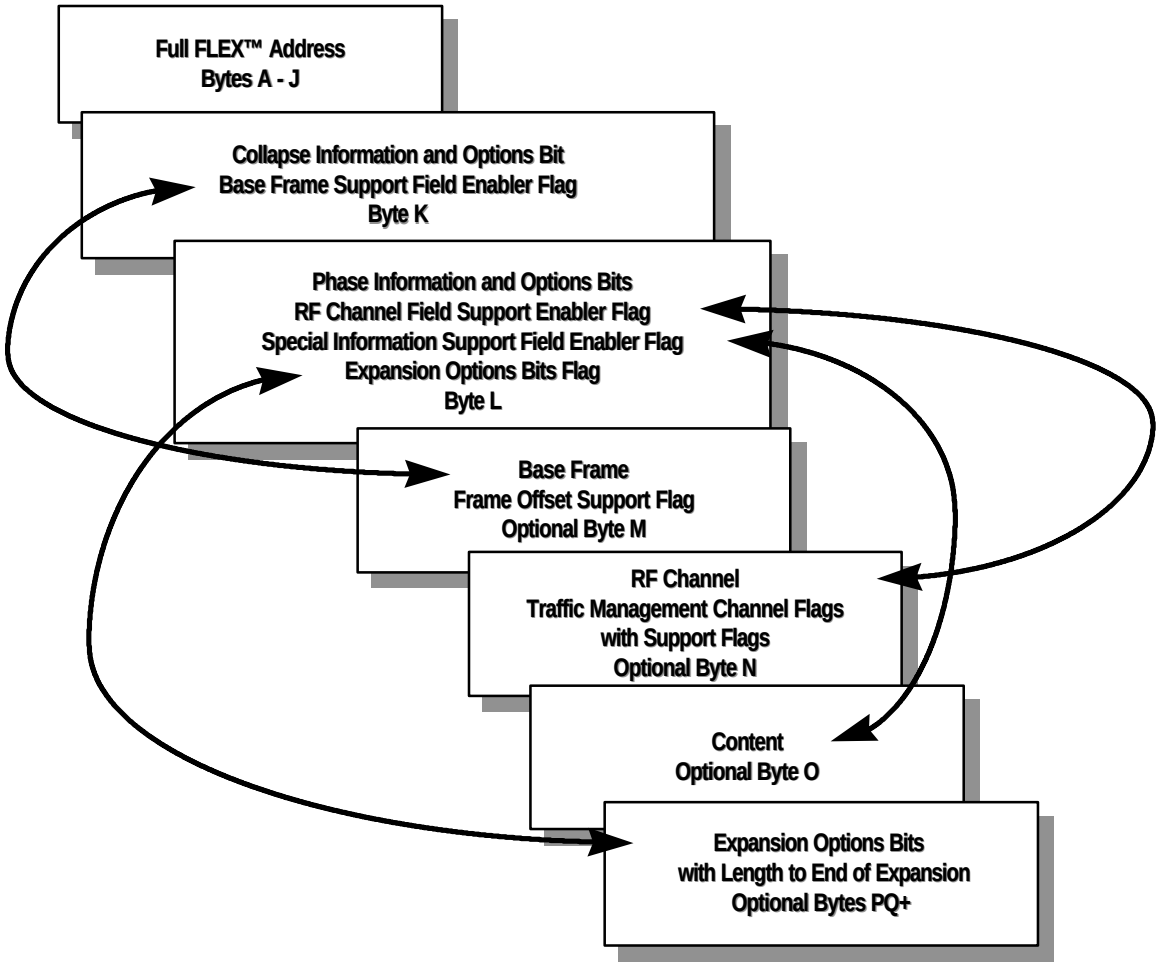

Figure 10 FLEX™-AUGMENTED CAP CODE OPTIONAL BYTES