NOT MEASUREMENT SENSITIVE

NOTICE OF CHANGE

MIL-STD-2045-44500
NOTICE 1
29 July 1994

MILITARY STANDARD
TACTICAL COMMUNICATIONS PROTOCOL 2
(TACO2) FOR THE NATIONAL IMAGERY
TRANSMISSION FORMAT STANDARD

TO: ALL HOLDERS OF MIL-STD-2045-44500:

1. THE FOLLOWING PAGES OF MIL-STD-2045-44500 HAVE BEEN REVISED AND SUPERSEDE THE PAGES LISTED:

NEW PAGE	DATE	SUPERSEDED PAGE	DATE
ix	18 June 1994	ix	29 July 1994
x	18 June 1994	x	REPRINTED WITHOUT CHANGE
13	18 June 1994	13	REPRINTED WITHOUT CHANGE
14	18 June 1994	14	29 July 1994
29	18 June 1994	29	REPRINTED WITHOUT CHANGE
30	18 June 1994	30	29 July 1994
37	18 June 1994	37	REPRINTED WITHOUT CHANGE
38	18 June 1994	38	29 July 1994
49	18 June 1994	49	29 July 1994
50	18 June 1994	50	REPRINTED WITHOUT CHANGE
63	18 June 1994	63	REPRINTED WITHOUT CHANGE

AMSC N/A AREA DCPS

DISTRIBUTION STATEMENT A Approved for public release; distribution is unlimited.

NOTICE OF CHANGE MIL-STD-2045-44500

NEW PAGE	DATE	SUPERSEDED PAGE	DATE
64	18 June 1994	64	29 July 1994
81	18 June 1994	81	REPRINTED WITHOUT CHANGE
82	18 June 1994	82	29 July 1994
99	18 June 1994	99	REPRINTED WITHOUT CHANGE
100	18 June 1994	100	29 July 1994
101	18 June 1994	101	REPRINTED WITHOUT CHANGE
102	18 June 1994	102	29 July 1994

2. RETAIN THIS NOTICE AND INSERT BEFORE TABLE OF CONTENTS.

3. Holders of MIL-STD-2045-44500 will verify that page changes and additions indicated above have been entered. This notice page will be retained as a check sheet. This issuance, together with appended pages, is a separate publication. Each notice is to be retained by stocking points until the military standard is completely revised or canceled.

Custodians:
DISA: DC
Army: SC
Air Force: 90
Navy: OM
DIA: DI
NSA: NS
USMC: MC
DLA: DH
Other:
Joint Staff/Architecture & Integration
USSPACECOM

Preparing Activity:
DISA

Reviewing Activity:

Army	SC	Navy	EC, CH, ND, TD, OM
Air Force	02, 13, 17, 29, 90	USMC	MC, CG
DLA	DH		
DMA	MP		
DIA	DI		
DOT	OST		
OASD	IQ, DO, MA, IR		
ODISC 4	AC		
STRICOM	PT		

(Project DCPS-008)

CONTENTS

PARAGRAPH PAGE

FIGURE

CONTENTS

FIGURE PAGE

TABLE

APPENDIX

4. GENERAL REQUIREMENTS

4.1 Overview. To exchange NITFS messages between systems, the participants must agree on the mechanism and protocols to be used to support the exchange. In some cases, connectivity and transfer protocols already may exist; for instance, Defense Information Systems Network (DISN)-connected hosts can use File Transfer Protocol (FTP) for moving standard format files. In other cases, connectivity is available, but common transfer protocols are not, or the available protocols are intolerably inefficient; for instance, DISN protocols run very slowly over slow-turnaround half-duplex circuits, and cannot run at all over simplex circuits. TACO2 provides efficient NITFS message transfer across point-to-point and point-to-multipoint links (tactical radio circuits) where neither DISN nor other current GOSIP protocols are suitable.

4.1.1 Approach. TACO2 uses a layered model, similar in philosophy to the ISO Open Systems Interconnection Reference Model. The TACO2 model is shown on figure 1; the components are described in the following subparagraphs.

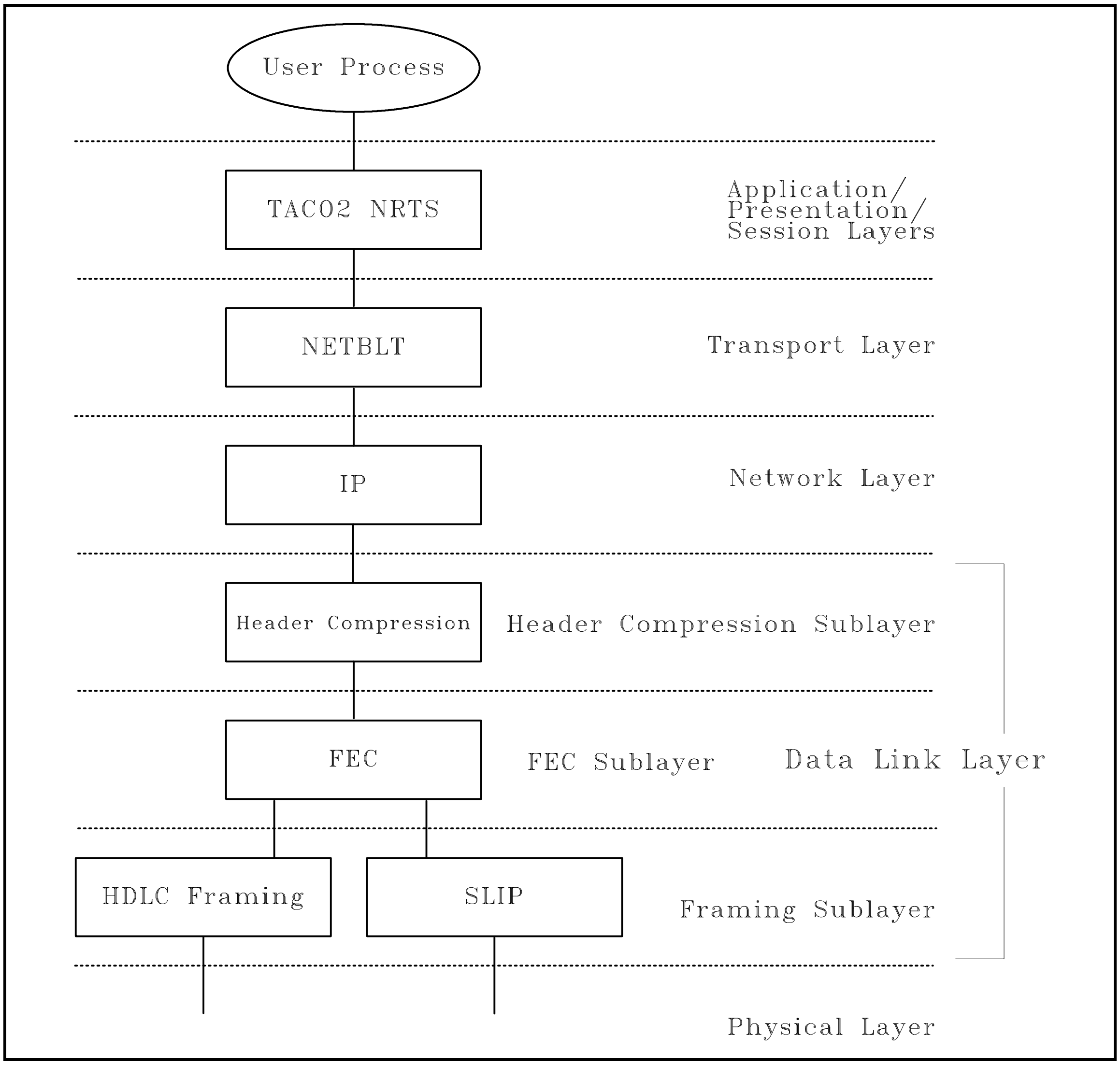

FIGURE 1. The TACO2 message transfer reference model.

4.1.2 NITFS reliable transfer server for TACO2 (TACO2 NRTS). The TACO2 NRTS controls the communications service to be used, exchanges the message and associated information with it, and acts as a session layer to allow resumption of interrupted transfers.

4.1.3 NETBLT. NETwork BLock Transfer (NETBLT) shall provide the reliable, flow-controlled transport level protocol designed to achieve high throughput across a wide variety of networks. It allows the sending client (the NRTS) to break the message being sent into a series of buffers, and to pass those buffers to NETBLT as information or space availability permits. Buffers are broken into packets for transmission. The critical element for performance is multiple buffering, so that new buffers can be sent

5.2.3.4 Flow control parameter renegotiation. The burst size and burst interval also may be re-negotiated after each buffer transmission to adjust the transfer rate according to the performance observed from transferring the previous buffer. The receiving end shall send burst size and burst interval values in its OK messages (described in 5.2.5.2.1). The sender shall compare these values with the values it can support. It then may modify either of the parameters, but only by making them more restrictive. The modified parameters shall then be communicated to the receiver in DATA, LDATA, or NULL-ACK packets.

5.2.3.5 Client-controlled flow. (Effectivity 3). A burst interval value of zero shall have a special meaning: internal flow control shall be turned off, so that only client level flow control shall be in effect. In this case, the sending NETBLT shall transmit packets without regard for the rate control mechanism. When using client-controlled flow, the receiver shall use an alternative method for data timer estimation (see 5.2.5.2.4.3).

5.2.4 Checksumming. NETBLT shall use checksums to validate the contents of packets and packet headers unless packet integrity is assured by the data link layer. The checksum value shall be the bitwise negation of the ones-complement sum of the 16-bit words being checksummed. On twos-complement machines, the ones-complement sum can be computed by means of an "end around carry"; that is, any overflows from the most significant bit are added into the least significant bits. See figure 8 for an example. If the quantity to be checksummed has an odd number of bytes, it shall be padded with a final null byte (binary 0's) to make the number of bytes even for the purpose of checksum calculation. The extra byte shall not be transmitted as part of the packet, but its existence shall be assumed at the receiving end for checksum verification.

	Word 1	0001	(Note: all numbers are in hexadecimal)
	Word 2	F203	
	Word 3	F4F5	
	Word 4	F6F7	
Sum showing carry:		2DDF0	
Sum without carry:		DDF0	
	Carry:	2	
Ones-complement sum:		DDF2	
Complement for checksum:		220D	

FIGURE 8. Example of checksumming.

5.2.5 NETBLT protocol operation. Each NETBLT transfer shall have three stages: connection setup, data transfer, and connection close. The stages are described in detail below, along with methods for ensuring that each stage completes reliably. State diagrams are provided at the end of the description for each stage of the transfer. Each transition in the diagrams is labeled with the event that causes the transition, and optionally, in parentheses, actions that occur at the time of the transition.

5.2.5.1 Connection setup. A NETBLT connection shall be set up by an exchange of two packets between the sending NETBLT and the receiving NETBLT. The sending end shall send an OPEN packet; the receiving end shall acknowledge the OPEN packet in one of two ways: it shall either send a REFUSED packet, indicating that the connection cannot be completed for some reason, or it shall complete the connection setup by sending a RESPONSE packet. After a successful connection setup, the transfer can begin. Figure 9 illustrates the opening of a connection by a sender, and figure 10 shows the same process for a receiver.

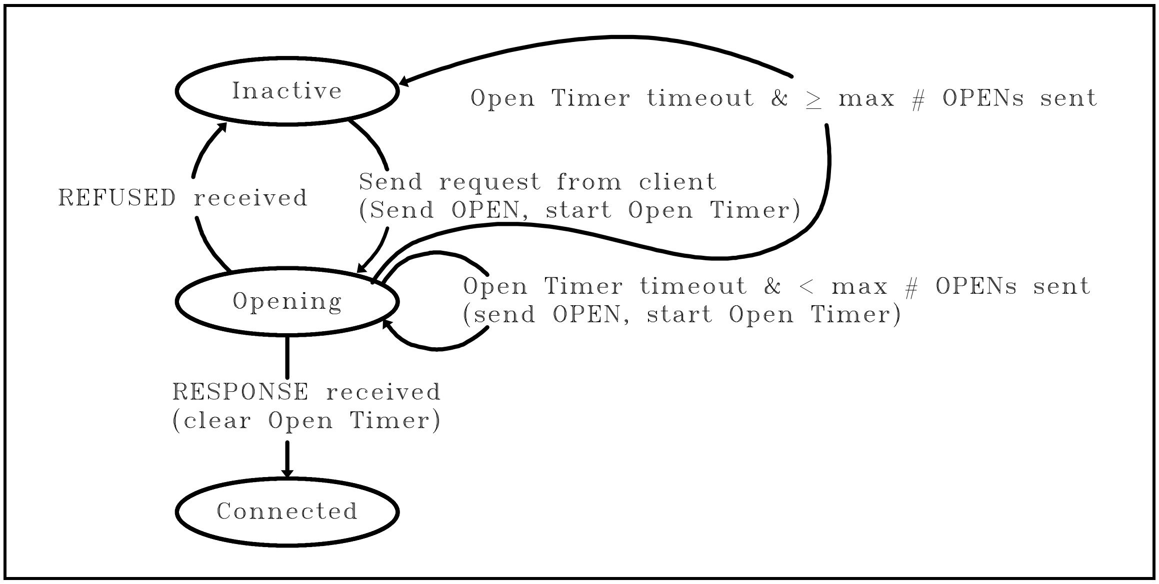

FIGURE 9. Sender open state diagram.

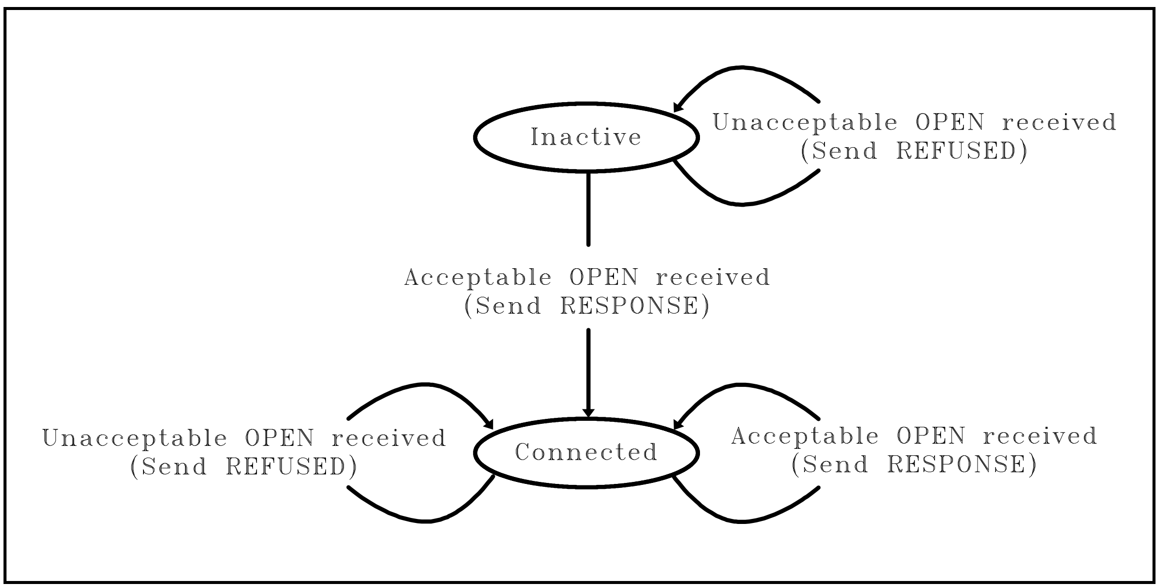

FIGURE 10. Receiver open state diagram.

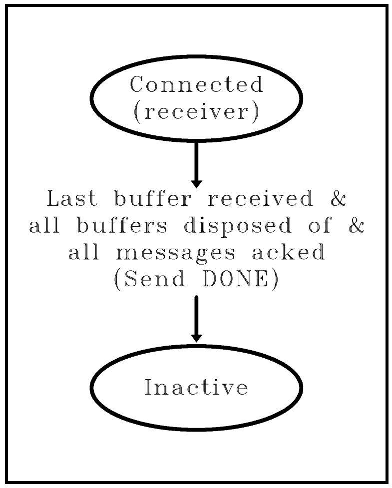

FIGURE 13. Receiver successful close state diagram.

5.2.5.3.1.2 Sender successful close. The sender shall recognize that the transfer has completed when the following are true: (1) it has transmitted DATA packets with a "last-buffer" flag set and (2) it has received OK messages for all its buffers. At that point, it shall "dally" for a predetermined period of time before closing its half of the connection. If the NULL-ACK packet acknowledging the receiver's last OK message was lost, the receiver has time to retransmit the OK message, receive a new NULL-ACK, and recognize a successful transfer. The dally timer value shall be based on the receiver's control timer value; it shall be long enough to allow the receiver's control timer to expire so that the OK message can be resent. The sender shall use the receiver's current control timer value to compute its dally timer value. A value of twice the receiver's control timer value is suitable for the dally timer. When the sender receives a DONE packet, it shall clear its dally timer and close its half of the connection. Figure 14 illustrates this sequence.

.

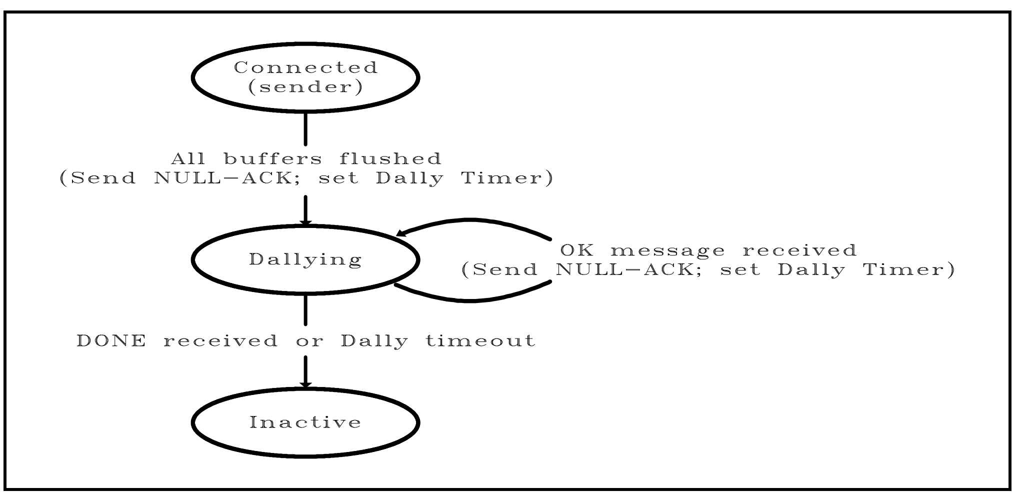

FIGURE 14. Sender successful close state diagram.

5.2.5.3.2 Client QUIT. During a NETBLT transfer, one client may send a QUIT packet to the other, to terminate the transfer prematurely. Since the QUIT occurs at a client level, the QUIT transmission shall occur only between buffer transmissions. The NETBLT receiving the QUIT packet shall take no action other than immediately notifying its client and transmitting a QUITACK packet. The QUIT sender shall time out and retransmit until a QUITACK has been received or its death timer expires. The sender of the QUITACK shall dally before quitting, so that it can respond to a retransmitted QUIT. Figure 15 illustrates this sequence.

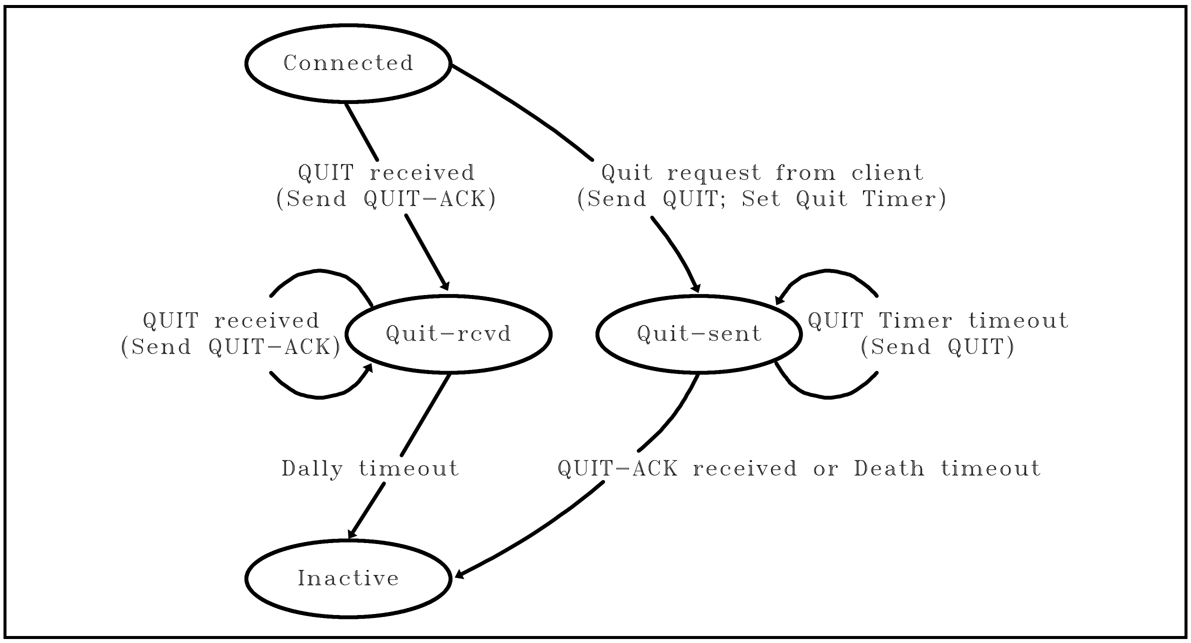

FIGURE 15. Quit state diagram.

5.2.5.3.3 NETBLT ABORT. An ABORT shall take place when an unrecoverable malfunction occurs. Since the ABORT originates in the NETBLT layer, it may be sent at any time. The ABORT implies that the NETBLT layer is malfunctioning, so no transmit reliability is expected, and the sender shall immediately close its connection. Figure 16 illustrates this sequence.

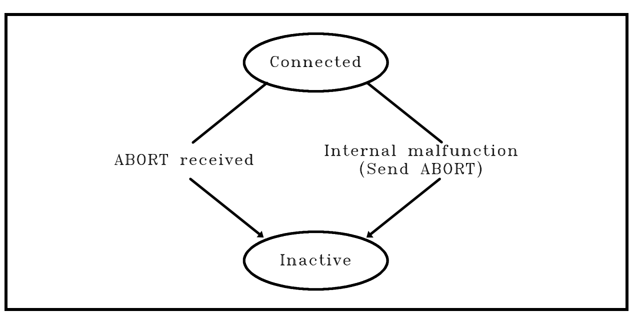

FIGURE 16. Abort state diagram.

5.2.9.5 REFUSED packets.

5.2.9.5.1 Reason for QUIT/ABORT/REFUSE. The reason shall be an appropriate ASCII string up to 80 characters long, suitable for display to the recipient. The use of REFUSED indicates that the connection cannot be completed for some reason.

(NOTE: Strings used may include:
"no service listening on port x," where x is the unacceptable port number.)

5.2.9.6 DATA and LDATA packets.

5.2.9.6.1 Packet number. The first data packet in each buffer shall be numbered 0.

5.2.9.6.2 Data area checksum value. All TACO2 DATA and LDATA packets shall be checksummed.

5.2.9.7 Timer precision. Timer precision in NETBLT shall be no worse than ±200 milliseconds (msec).

5.2.9.8 Open timer value. The open timer shall initially be set to no less than two seconds. In half-duplex and full duplex, the value of the open timer shall be increased by two seconds after each timeout.

5.2.9.9 Quit timer value. The quit timer shall be set to no less than five seconds.

5.2.9.10 Death timer value. The death timer should be set to no less than two minutes.

5.3 Network layer - IP.

5.3.1 Overview. The DOD IP forms the network layer of the TACO2 protocol suite. IP provides a mechanism for transmitting blocks of data (datagrams) from sources to destinations, which are specified by 32-bit addresses. It is a "best-effort" mechanism, which provides no assurance that a datagram is delivered, but takes appropriate steps when possible to move a datagram toward its destination. IP is specified in Internet RFC 791, as amended by RFC 950 (IP Subnet Extension), RFC 919 (IP Broadcast Datagrams), and RFC 922(IP Broadcast Datagrams with Subnets). It usually also includes the Internet Control Message Protocol (ICMP), specified in RFC 792, which provides a mechanism for communicating control and error information between hosts and other hosts or gateways. Although ICMP is an integral part of IP, it uses the support of IP as if it (ICMP) were a higher level protocol. IP is also specified in MIL-STD-1777, which formally specifies a protocol consistent with RFC 791.

5.3.1.1 IP augmentations. As used in TACO2, IP may be augmented by the revised IP Security Option (RFC 1108), and by the Host Extensions for IP Multicasting (RFC 1112). These augmentations are not required in this version of TACO2, but they may be necessary for operation in certain environments. TACO2 supports a limited form of multicasting by allowing simplex receivers to "listen in" on simplex, half-duplex, or full-duplex transmissions. (Effectivity 5: later versions of TACO2 may support acknowledged multicast).

5.3.2 Required IP components. Because TACO2 uses IP outside its normal internetworked environment, some components of IP are unnecessary or inappropriate in some cases. This section identifies the required components for each major case.

5.3.2.1 Simplex. Simplex transmission may be used to support point-to-point or broadcast communication in TACO2. The Internet Header format shall be as specified in 5.3.3. The following fields shall be correctly filled in and interpreted for simplex operation:

a. Version

b. Internet Header Length

c. Total Length

d. Fragment Offset (must be 0)

e. Protocol (30 for NETBLT)

f. Header Checksum

g. Source Address

h. Destination Address

i. IP Security Option, if required

The remaining fields shall be disregarded by a receiver in simplex operation, but shall be provided by a transmitter for the sake of consistency. Datagrams shall not be fragmented. Subnetting support is not required. ICMP shall not be used in simplex communications.

Received fields shall be filled in with values computed by combining the abbreviated header value of the corresponding field and the receiver's connection state variable value. For the Last Buffer Touched and High Consecutive Sequence Number Received fields, the computed value shall contain the abbreviated header value for the low-order portion, and the smallest value such that the result is no less than the old connection state variable value for the high-order portion. For the Buffer Number field, the computed value shall use the abbreviated header value for the low-order portion, and a value that causes minimal change between the old connection state variable value and the new computed value for the high-order portion. The connection state variable value shall be changed to correspond to the new computed value. The NETBLT Packet Number field shall be filled in with the abbreviated header Packet Number field, padded with zeros on the left. The NETBLT "L" bit shall be set according to the "LB" bit in the abbreviated header, and the NETBLT Type field shall be DATA or LDATA according to the "LP" bit in the abbreviated header. Burst Size and Burst Interval shall be filled in according to the variable values. The remaining NETBLT fields shall be filled in accordance with 5.2. The reconstructed NETBLT packet shall be passed directly to the NETBLT layer (bypassing the IP layer) for normal processing.

5.4.2 FEC sublayer. Using the FEC sublayer is optional; the inclusion of FEC-I in any compliant implementation of TACO2 is mandatory.

5.4.2.1 FEC-I code. The FEC-I encoding process takes each IP datagram to be transmitted, adds Reed-Solomon redundancy, and passes the encoded datagram to the link layer for encapsulation, generally as an HDLC frame as specified in 5.4.3.1. As a result, the HDLC implementation shall (TBR) allow received packets to be processed by the FEC sublayer even if the HDLC checksum is in error. The encoded datagram also may be encapsulated as a SLIP frame, as specified in 5.4.3.2. If the unencoded datagram contains K bytes where K is not greater than 152, the encoded datagram contains a single Reed-Solomon codeword containing K + 10 bytes. For purposes of Reed-Solomon code definition, these bytes are numbered from 0 to K + 9 as shown on figure 19 (where the left end of the figure represents the beginning of the datagram if viewed in time-sequence).

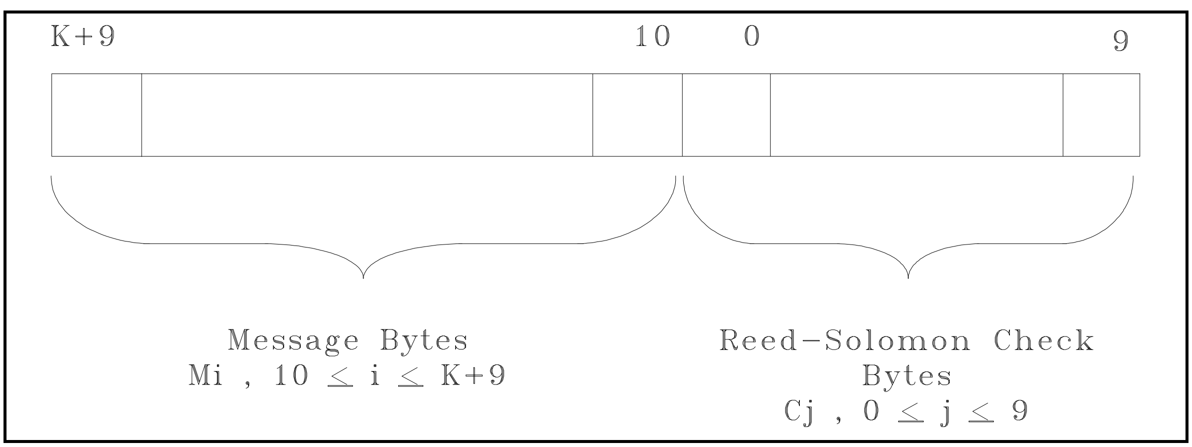

FIGURE 19. FEC-I format.

The FEC-I code uses arithmetic in the Galois field GF(256), as specified in the standard ISO 9171 for 5.25" WORM disk coding. This field has 256 elements, which are represented by 8-bit symbols ("octets" or simply "bytes"), using the generator polynomial $x^8 + x^5 + x^3 + x^2 + 1$. The primitive element α has hexadecimal value 0x02. The Reed-Solomon check bytes Cj are defined by the following:

$$C_j = \sum_{i=10}^{K+9} \frac{M_i}{\alpha^i + \alpha^j} \quad , \quad j = 0...9$$

5.4.2.1.1 Correction capability. Each codeword as defined in 5.4.2.1 has distance 11 and is fully correctable with up to five independent byte-errors. FEC-I can fully correct all patterns of five or fewer erroneous bytes in any codeword. Note that the content and sequence of the message bytes M_i remains unchanged by the encoding process.

5.4.2.1.2 Long datagrams. If the length of the datagram to be encoded is greater than 152 bytes but does not exceed 752 bytes, encoding is performed by including up to five separate concatenated Reed-Solomon codewords in the encoded datagram. The maximum encoded length is 802 bytes. As an example, figure 20 shows how a datagram with an unencoded length of 450 bytes would be encoded, giving a datagram 480 bytes long.

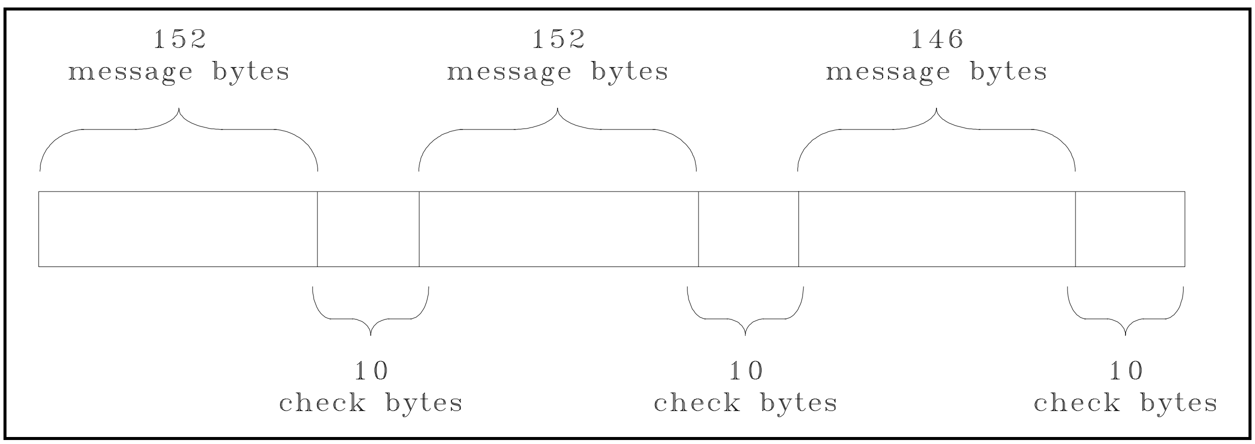

FIGURE 20. Encoding a 450 byte packet.

As shown on figure 20, only the final codeword in a multicodeword encoded datagram may be truncated to fewer than 152 message bytes. The time sequence of the message bytes is unchanged by the encoding process. FEC-I encoding is presently not specified for datagrams whose unencoded length is greater than 752 bytes. Should a FEC-I encoder be presented with such a datagram, the correct action is to transmit it without any encoding.

6.5 Effectivity summary. Some of the capabilities specified in this document are not required as of the issue date of the document. All such capabilities are marked with effectivity numbers, for example, (Effectivity 1). Each effectivity number will be replaced by a specific date in subsequent releases of this document.

6.5.1 Effectivity 1 - FEC I and Bit Error Ratio Test (BERT).

a. 4.1.6 FEC. Forward Error Correction (FEC) is a mandatory component of the TACO2 protocol stack whose use in a particular circuit is user selectable (Effectivity 1). …

6.5.2 Effectivity 2 - FEC II.

a. APPENDIX C FEC-II CODE. (The contents of this section are (Effectivity 2) pending further implementation and testing of the proposed FEC code.)

b. 5.4.2.2.3 FEC-II. FEC-II is applied to a SLIP and/or HDLC encapsulated datalink as described in appendix C (Effectivity 2).

6.5.3 Effectivity 3 - Header abbreviation and client-controlled flow.

a. 4.1.5 Header Abbreviation sublayer. TACO2 provides a mechanism for header abbreviation across point-to-point links. Use of the header abbreviation sublayer is optional: its inclusion in any compliant implementation of TACO2 shall be mandatory (Effectivity 3).

b. 5.2.3.5 Client-controlled flow. (Effectivity 3)

c. 5.4.1 Header Abbreviation sublayer. TACO2 provides a mechanism for header abbreviation across point-to-point links. Use of the header abbreviation sublayer is optional: its inclusion in any compliant implementation of TACO2 shall be mandatory (Effectivity 3).

6.5.4 Effectivity 4 - Pull vs. push.

a. 5.1 NITFS reliable transfer server for TACO2 (TACO2 NRTS). The TACO2 NRTS described here assumes an active sender and a passive receiver ("push" operation); as of the effectivity date (Effectivity 4) the TACO2 NRTS shall also support an active receiver and passive sender ("pull" operation).

b. 5.2.9.2.5 Direction. (Effectivity 4) Until the effectivity date, operation of TACO2 is defined only for "M" set to 1; that is, TACO2 allows only active sending and passive receiving. Following that date, operation with "M" set to 0 is also permissible.

6.5.5 Effectivity 5 - Multicast.

a. 5.3.1.1 IP augmentations. … TACO2 supports a limited form of multicasting by allowing simplex receivers to "listen in" on simplex, half-duplex, or full-duplex transmissions; (Effectivity 5: later versions of TACO2 may support acknowledged multicast).

6.5.6 Effectivity 6 - Medium Access Control layer.

a. 5.4 Data link layer. The Data Link layer in TACO2 is divided into three sublayers: Header Abbreviation, FEC, and Framing. (Effectivity 6: a Medium Access Control Layer, just below the Framing Sublayer, is under consideration.)

6.5.7 Effectivity 8 - Defense Information Systems Network (DISN).

a. DISA/JIEO Circular 9008

b. DISA/JIEO Specification 9137

c. DISA/JIEO Specification 9138

d. DISA/JIEO Specification 9139

e. DISA/JIEO Specification 9140

6.7 Subject term (key word) listing.

Error detection
Forward error correction (FEC)
Frames
HDLC
ICMP
IP
Message Transfer Facility
NETBLT
Packets
Secondary Imagery Dissemination Systems
SIDS
SLIP

APPENDIX C

FEC-II CODE

(The contents of this section are (Effectivity 2) pending further implementation and testing of the proposed FEC code.)

10. Scope. This appendix is not a mandatory part of the standard. The information it contains is intended for guidance only.

20. Applicable documents. This section is not applicable to this appendix.

30. FEC-II Code. FEC-II encoding applies both Reed-Solomon and BCH coding for operation in high error environments. Further, FEC-II encodes a single datagram or section of a datagram into a group of "fragments," or small packets of 12 bytes each. The BCH coding protects each fragment, and using the Reed-Solomon coding, up to eight fragments may be lost in transmission while still allowing the receiver to recover the datagrams.

(The term fragment used in this section, is unrelated to the fragments defined by the Internet Protocol.)

The individual fragments shall be encapsulated by the data link layer, generally either as a SLIP frame as specified in 5.4.3.2, or as an HDLC frame as specified in 5.4.3.1. If a FEC-II fragment is encapsulated as an HDLC frame, the 2-byte HDLC frame opening sequence, as well as the 2-byte CRC, shall not be included in the frame.

Datagrams up to length 382 bytes may be encoded by the FEC-II coding process. The first step shall be to represent the datagram by either one or two "sections." If the datagram is of length L, where L is 191 bytes or less, it shall be represented by a single "section" containing L + one bytes as follows: an initial byte containing the byte count L, followed by the L bytes contained in the datagram. If the datagram is of length L, where L is 192 through 382 bytes inclusive, it shall be represented by two sections. The first section shall consist of an initial byte with value 255 (decimal), followed by the first 191 bytes contained in the datagram. The second section shall consist of an initial byte with value (L - 191), followed by the remaining bytes of the datagram.

First we will specify the format of a single fragment, and then describe how the input bytes used to form the fragments are derived from the unencoded section.

Each 12-byte fragment is formed from an 8-byte sub-block, which is made up of the "input bytes" on figure C-1.

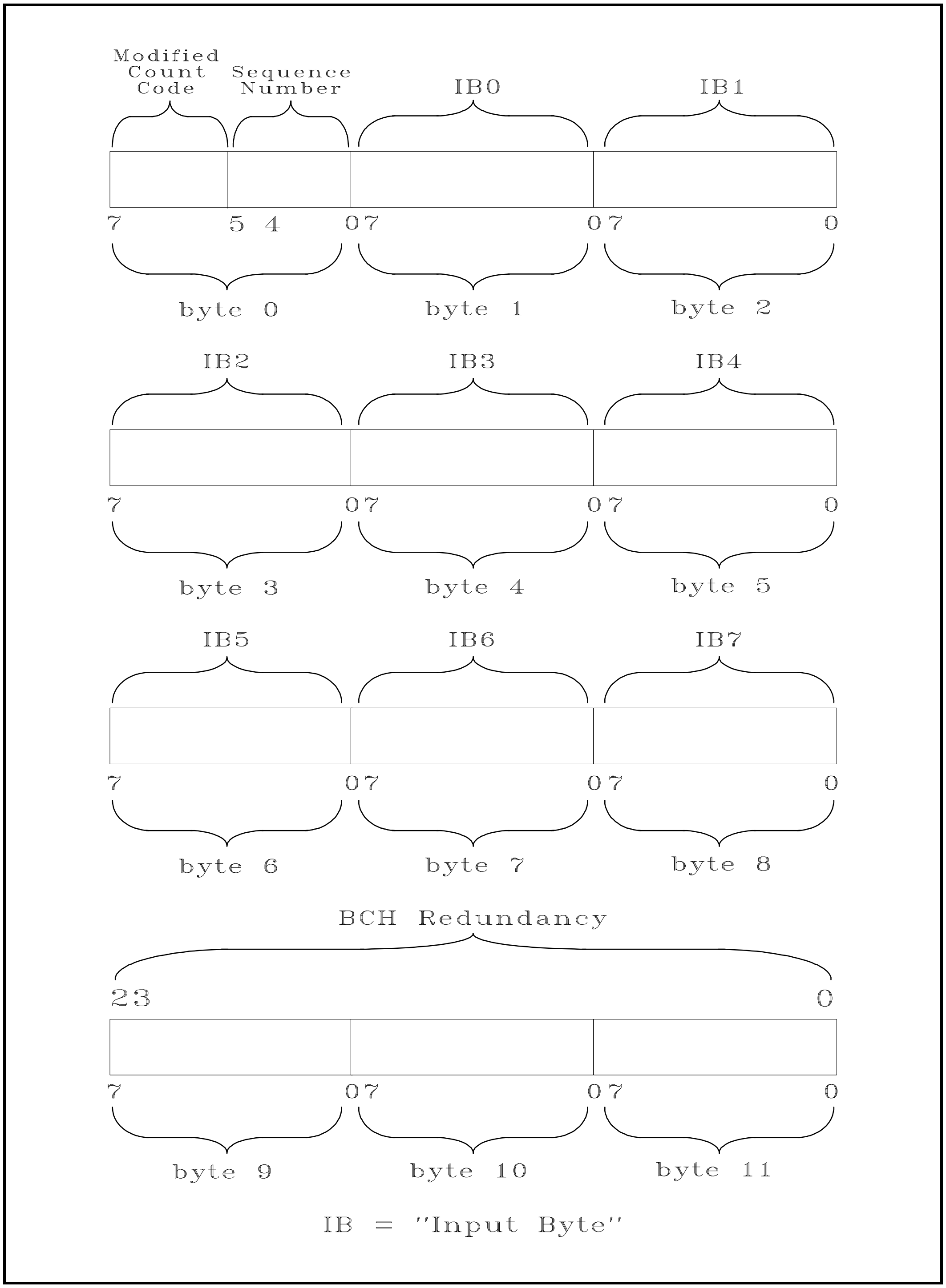

FIGURE C-1. FEC-II fragment format.

The origin of the 3-bit Modified Count Code and 5-bit sequence number will be described shortly. The 24-bit BCH redundancy field is formed using the following polynomial:

$$x^{24} + x^{23} + x^{21} + x^{20} + x^{19} + x^{17} + x^{16} + x^{15} + x^{13} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + 1$$

To calculate the BCH redundancy, feed the following bits, in the order shown, into a feedback shift register, initialized with the hexadecimal value 0x0000FF, and wired according to the above polynomial:

Modified Count Code bits 2 through 0
Sequence Number bits 4 through 0
IB0 bits 7 through 0
IB1 bits 7 through 0
.
.
.
IB7 bits 7 through 0

The above describes how to form a fragment given the input bytes, modified count code, and sequence number. We now describe how the input bytes, modified count codes, and sequence numbers are derived.

First, the unencoded section is split into a sequence of 8-byte sub-blocks. Eight more 8-byte sub-blocks containing Reed-Solomon redundancy are then created. Each sub-block contains bytes numbered 0 through 7.

Eight choices exist for the number of message sub-blocks used to represent a datagram, as determined by a 3-bit count code in table C-I.

TABLE C-I. Count codes for different length sections.

Count Code	**Number of Message Sub-blocks**	**Sequence Numbers of Message Sub-blocks**	**Nominal Section Length (bytes)**
0	4	0-3	32
1	6	0-5	48
2	8	0-7	64
3	10	0-9	80
4	13	0-12	104
5	16	0-15	128
6	19	0-18	152
7	24	0-23	192

The eight Reed-Solomon Redundancy Sub-blocks always have sequence numbers 24 through 31.

The 3-bit Modified Count Code inserted in the fragment is the bitwise-exclusive-OR formed from the above Count Code, and a residue computed from the remaining data in the fragment. This residue is defined as follows (using binary arithmetic):

$$\textit{Residue} = \left\{ (\textit{sequence number} \bmod 9) + \sum_{i=0}^{7} (\textit{Input byte } i) \bmod 9 \right\} \bmod 8$$

Each byte in a Reed-Solomon Redundancy Sub-block with sequence number *j* is computed from the correspondingly-numbered bytes M_i in each of the message sub-blocks as follows:

$$C_{T(j)} = \sum_{i} \frac{M_i}{\alpha^{T(i)} + \alpha^{T(j)}} \quad , \quad 24 \le j \le 31$$

where *i* ranges over the message sub-block sequence numbers, and the correspondence between sequence numbers and locations within the Reed-Solomon codeword is:

STANDARDIZATION DOCUMENT IMPROVEMENT PROPOSAL

INSTRUCTIONS

1. The preparing activity must complete blocks 1,2, 3, and 8. In block 1, both the document number and revision letter should be given.
2. The submitter of this form must complete blocks 4, 5, 6, and 7.
3. The preparing activity must provide a reply within 30 days from receipt of the form.

NOTE: This form may not be used to request copies of documents, nor to request waivers, or clarification of requirements on current contracts. Comments submitted on this form do not constitute or imply authorization to waive any portion of the referenced document(s) or to amend contractual requirements.

I RECOMMEND A CHANGE:	1. DOCUMENT NUMBER MIL-STD-2045-44500, NOTICE 1, 29 July 1994	2. DOCUMENT DATE *(YYMMDD)* 930618

3. DOCUMENT TITLE

Tactical Communications Protocol 2 (TACO2) for the National Imagery Transmission Format Standard

4. NATURE OF CHANGE *(Identify paragraph number and include proposed rewrite, if possible. Attach extra sheets as needed.)*

5. REASON FOR RECOMMENDATION

6. SUBMITTER

a. NAME *(Last, First, Middle Initial)*	b. ORGANIZATION	
c. ADDRESS *(Include Zip Code)*	d. TELEPHONE *(Include Area Code)* (1) Commercial (2) AUTOVON	**7. DATE SUBMITTED** ***(YYMMDD)***

8. PREPARING ACTIVITY DEFENSE INFORMATION SYSTEMS AGENCY (DISA)

a. NAME Director, Joint Interoperability and Engineering Organization	b. TELEPHONE *(Include Area Code)* (1) Commercial 908-532-7732 (2) AUTOVON DSN 992-7732
c. ADDRESS *(Include Zip Code)* ATTN: TBBF Building 283, Squier Hall Ft. Monmouth, NJ 07703	**IF YOU DO NOT RECEIVE A REPLY WITHIN 45 DAYS, CONTACT:** Defense Quality and Standardization Office 5203 Leesburg Pike, Suite 1403, Falls Church, VA 22041-3466 Telephone (703) 756-2340 AUTOVON 289-2340