NOTICE OF CHANGE

NOT MEASUREMENT SENSITIVE

MIL-STD-188-141B
NOTICE 1
31 AUGUST 2001

DEPARTMENT OF DEFENSE INTERFACE STANDARD

INTEROPERABILITY AND PERFORMANCE STANDARDS FOR MEDIUM AND HIGH FFREQUENCY RADIO EQUIPMENT

TO ALL HOLDERS OF MIL-STD-188-141B:

1. The following pages of MIL-STD-188-141B have been revised and supersede the pages listed:

NEW PAGE(S)	DATE	SUPERSEDED PAGE(S)	DATE
1	31 August 2001	1	1 March 1999
10-11	31 August 2001	10-11	1 March 1999
14	31 August 2001	14	1 March 1999
22-23	31 August 2001	22-23	1 March 1999
25	31 August 2001	25	1 March 1999
28	31 August 2001	28	1 March 1999
30	31 August 2001	30	1 March 1999
35	31 August 2001	35	1 March 1999
43	31 August 2001	43	1 March 1999
54-56	31 August 2001	54-56	1 March 1999
64-65	31 August 2001	64-65	1 March 1999
175	31 August 2001	175	1 March 1999
190-196	31 August 2001	190-196	1 March 1999
198-207	31 August 2001	198-207	1 March 1999
212	31 August 2001	212	1 March 1999
215	31 August 2001	215	1 March 1999
246	31 August 2001	246	1 March 1999
248	31 August 2001	248	1 March 1999
265-275	31 August 2001	265-275	1 March 1999
275a	31 August 2001	new page	
276-282	31 August 2001	276-282	1 March 1999
282a	31 August 2001	new page	
283-289	31 August 2001	283-289	1 March 1999
291-295	31 August 2001	291-295	1 March 1999
297-315	31 August 2001	297-315	1 March 1999
317-439	31 August 2001	317-439	1 March 1999
489-498	31 August 2001	489-498	1 March 1999
498a-498f	31 August 2001	new pages	
506-511	31 August 2001	506-511	1 March 1999
514-517	31 August 2001	514-517	1 March 1999
520	31 August 2001	520	1 March 1999

1. SCOPE.

1.1 Scope.

The purpose of this document is to establish technical performance and interface parameters in the form of firm requirements and optional design objectives (DO) that are considered necessary to ensure interoperability and interface standardization of new long-haul and tactical radio systems in the medium frequency (MF) band and in the high frequency (HF) band. It is also the purpose of this document to establish a level of performance for new radio equipment that is considered necessary to satisfy the requirements of the majority of users. These technical parameters, therefore, represent a minimum set of interoperability, interface, and performance standards. The technical parameters of this document may be exceeded in order to satisfy certain specific requirements, provided that interoperability is maintained. That is, the capability to incorporate features such as additional standard and nonstandard interfaces is not precluded.

1.2 Applicability.

This standard is approved for use within the Department of Defense (DoD) in the design and development of new MF and HF radio systems. It is not intended that existing equipment and systems be immediately converted to comply with the provisions of this standard. New equipment and systems, and those undergoing major modification or rehabilitation, shall conform to this standard. If deviation from this standard is required, the user should contact the lead standardization activity for waiver procedures.

1.3 Application guidance.

The terms “system standard” and “design objective” are defined in FED-STD-1037. In this document, the word “shall” identifies firm requirements. The word “should” identifies design objectives that are desirable but not mandatory.

2. APPLICABLE DOCUMENTS.

2.1 General.

The documents listed in this section are specified in sections 3, 4, and 5 of this standard. This section does not include documents cited in other sections of this standard, those recommended for additional information, or those used as examples. While every effort has been made to ensure the completeness of this list, document users are cautioned that they must meet all specified requirements documents cited in sections 3, 4, and 5 of this standard, whether or not they are listed.

2.2 Government documents.

2.2.1 Specifications, standards, and handbooks.

The following specifications, standards, and handbooks form a part of this document to the extent specified herein. Unless otherwise specified, the issues of these documents are those listed in the issue of the Department of Defense Index of Specifications and Standards (DODISS) and supplement thereto cited in the solicitation (see paragraph 6.3).

4.4.3 High-performance HF data modems.
If interoperation with NATO member nations is required, land, air, and maritime, single-channel HF radio equipment shall comply with the "Associated communications equipment" requirements of STANAG 4539.

4.4.4 QSTAGs.
If interoperation among American, British, Canadian, Australian (ABCA), and New Zealand Armies is required, HF combat net radio equipment shall comply with the applicable requirements of the current edition of QSTAG 733.

4.5 Adaptive communications.
Adaptive HF describes any HF communications system that has the ability to sense its communications environment, and, if required, to automatically adjust operations to improve communications performance. Should the user elect to incorporate adaptive features, they shall be in accordance with the requirements for those features stated in this document.

The essential adaptive features are:

a. Channel (frequency) scanning capability.

b. ALE using an embedded selective calling capability. A disabling capability and a capability to inhibit responses shall be included.

c. Automatic sounding (station-identifiable transmissions). A capability to disable sounding and a capability to inhibit responses shall be included.

d. Limited link quality analysis (LQA) for assisting the ALE function:

(1) Relative data error assessment.

(2) Relative signal-plus-noise-plus-distortion to noise-plus-distortion ratio (SINAD).

(3) Multipath/distortion assessment (DO) (optional).

e. Automatic link maintenance

f. Channel occupancy detection

4.6 Linking protection.
LP refers to the protection of the linking function required to establish, control, maintain, and terminate the radio link. Because this protection is applied to the link establishment function, LP is a data link layer function in terms of the Seven Layer Reference Model. Figure B-1, Appendix B shows a conceptual model of the MIL-STD-188-141 data link layer functions, showing the placement within the data link layer at which linking protection shall be implemented. Voice transmissions or data transmissions from external modems are not affected by the LP. The LP application levels and their corresponding protection interval (PI) are defined in Appendix B, paragraphs B 4.1.1 through B 4.1.1.5.

4.7 HF data link protocol.
See Appendix G, Second-Generation Data Link Protocol.

4.8 Networking functions.

a. MIL-STD-188-141 establishes the technology baseline needed for establishing and maintaining links among HF radio stations. Networking technology augments this direct connection capability with the ability to find and use indirect routes.

b. The functions performed at the network layer may be grouped into two broad categories: routing functions and data management functions. Routing functions select paths through the network for voice and data traffic, using stored information (provided by operators, local data link controllers, and remote networking controllers) about the quality of available links to other stations. Data management functions acquire and communicate that (and other) information.

c. Link-level error statistics directly characterize the quality of single-link paths and are used to compute end-to-end path quality for multiple-link paths through relays. These results are stored in a path quality matrix (PQM), which is organized to provide the path quality to any reachable destination via each directly-reachable relay station. From this path quality data, a routing table (RT) is formed. This table lists the best path to each reachable station for various types of communication (e.g., voice and data).

4.8.1 Indirect calling and relaying.
When a station cannot directly link with a desired destination, other stations may be employed to assist in getting the message through. The simplest option is to have the local link controller or the HF Network Controller (HFNC) establish a link with a station other than the desired destination so that the station operators can manually communicate (using either voice or data orderwire) after the fashion of a torn-tape relay. When the equipment at the intermediate station is able to automatically establish an indirect path to the destination, this is termed relaying. A variety of relaying techniques are possible, some of which are shown in Figure 3. These techniques are differentiated where the cross-connection occurs in the protocol stack. Each alternative is briefly discussed in Table I.

4.8.2 Network management.
See Appendix D, HF Radio Networking, and Appendix H, Management Information Base

4.9 HF e-mail and other application protocols for HF radio networks.
See Appendix E, Application Protocols for HF Radio Networks.

5.2 Common equipment characteristics.

These characteristics shall apply to each transmitter and to each receiver unless otherwise specified.

5.2.1 Displayed frequency.

The displayed frequency shall be that of the carrier, whether suppressed or not.

5.2.2 Frequency coverage.

The radio equipment shall be capable of operating over the frequency range of 2.0 MHz to 29.9999 MHz in a maximum of 100-Hz frequency increments (DO: 10-Hz) for single-channel equipment, and 10-Hz frequency increments (DO: 1-Hz) for multichannel equipment.

5.2.3 Frequency accuracy.

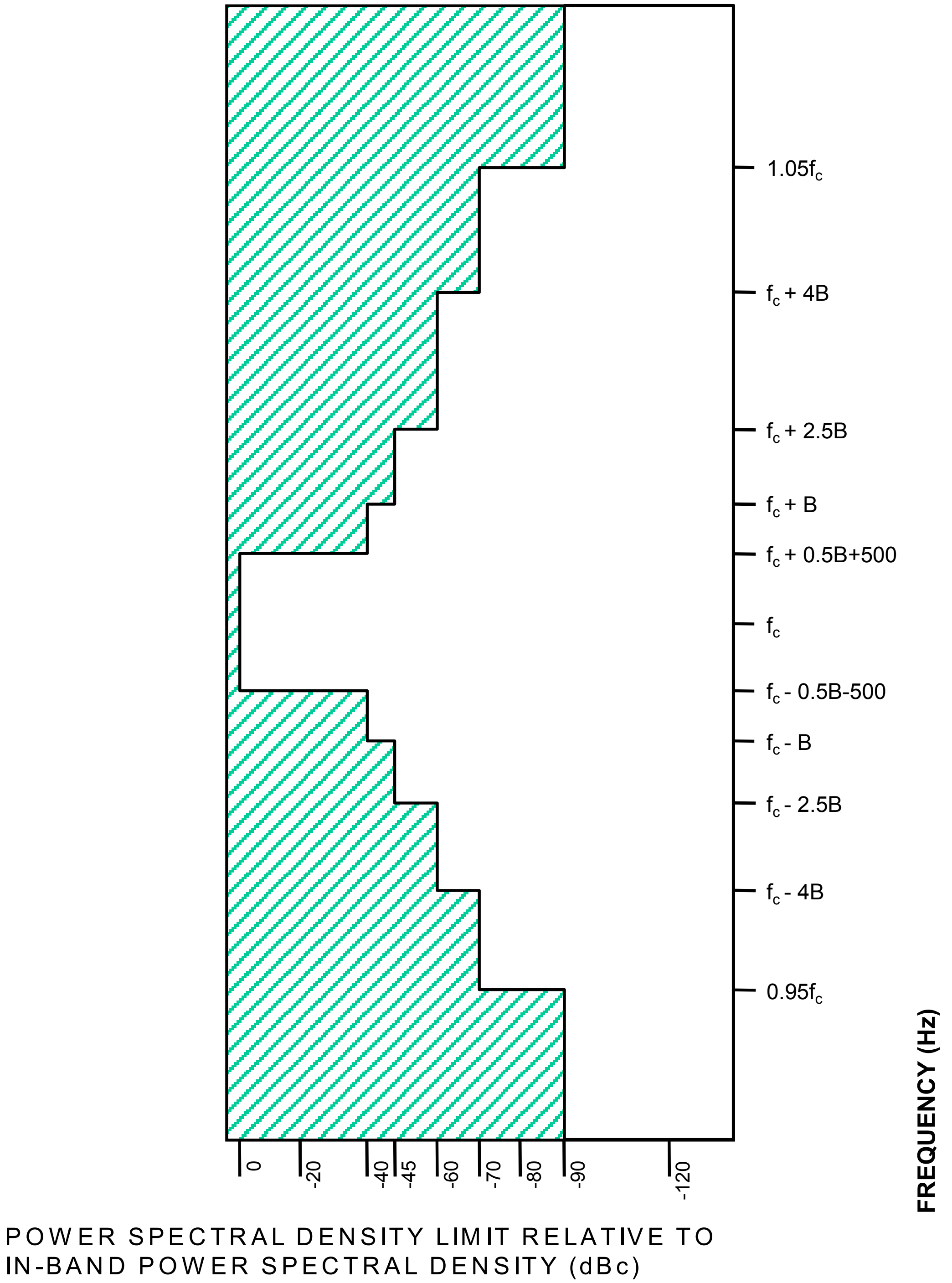
The accuracy of the radio carrier frequency, including tolerance and long-term stability, but not any variation due to doppler shift, shall be within ±30 Hz for tactical application and within ±10 Hz for all others, during a period of not less than 30 days. Tactical systems that must interoperate with long haul systems shall meet the ±10 Hz radio carrier frequency specification.

5.2.4 Phase stability.

The phase stability shall be such that the probability that the phase difference will exceed 5 degrees over any two successive 10 millisecond (ms) periods (13.33-ms periods may also be used) shall be less than 1 percent. Measurements shall be performed over a sufficient number of adjacent periods to establish the specified probability with a confidence of at least 95 percent.

5.2.5 Phase noise.

The synthesizer and mixer phase-noise spectrum at the transmitter output shall not exceed those limits as depicted in figures 4 and 5 under continuous carrier single-tone output conditions. Figure 4 depicts the limits of phase noise for cosited and non-cosited fixed-site and transportable long-haul radio transmitters. Figure 5 depicts the limits for tactical radio transmitters. Tactical systems that must interoperate with long haul systems shall meet the long haul phase noise specification in Figure 4.



NOTES:

1. B=Necessary bandwidth (Hz).
2. F_C=Center frequency of bandwidth (Hz).
3. Emissions shall fall within the unshaded portion of the curve.

FIGURE 8. Out-of-band power spectral density for HF transmitters.

5.3.2.2 Discrete frequency spurious emissions.
For HF transmitters, when driven with a single tone to produce an rf output of 25 percent rated PEP, all discrete frequency spurious emissions shall be suppressed as follows:

a. For fixed application (see Figure 9a)

- Between the carrier frequency fc and fc ± 4B (where B = bandwidth), at least 40 dBc.
- Between fc ± 4B and ± 5 percent of fc removed from the carrier frequency, at least 60 dBc.
- Beyond ±5 percent removed from the carrier frequency, at least 80 dBc.
- Harmonic performance levels shall not exceed -63 dBc.

b. For tactical application (see Figure 9b)

- Between the carrier frequency f_c and f_c ± 4B (where B = bandwidth), at least 40 dBc.
- Beyond f_c ± 4B at least 50 dBc.
- Harmonic performance levels shall not exceed -40 dBc.

5.3.3 Carrier suppression.

The suppressed carrier for tactical applications shall be at least 40 dBc (DO: 60 dBc) below the output level of a single tone modulating the transmitter to rated PEP. The suppressed carrier for fixed site applications shall be at least 50 dBc (DO: 60 dBc) below the output level of a single tone modulating the transmitter to rated PEP.

5.3.4 Automatic level control (ALC).

Starting at ALC threshold, an increase of 20 dB in audio input shall result in less than a 1 dB increase in average rf power output.

5.3.5 Attack and release time delays.

5.3.5.1 Attack-time delay.
The time interval from keying-on a transmitter until the transmitted rf signal amplitude has increased to 90 percent of its steady-state value shall not exceed 25 ms (DO: 10 ms). This delay excludes any necessary time for automatic antenna tuning.

5.3.5.2 Release-time delay.
The time interval from keying-off a transmitter until the transmitted rf signal amplitude has decreased to 10 percent of its key-on steady-state value shall be 10 ms or less.

5.3.6 Signal input interface characteristics.

5.3.6.1 Input signal power.
Input signal power for microphone or handset input is not standardized. When a line-level input is provided (see paragraph 5.3.6.2), rated transmitter PEP shall be obtainable for single tone amplitudes from -17 dBm to +6 dBm (manual adjustment permitted).

5.3.6.2 Input audio signal interface.

5.3.6.2.1 Unbalanced interface.
When an unbalanced interface is provided, it shall have an audio input impedance of a nominal 150 ohms, unbalanced with respect to ground, with a minimum return loss of 20 dB against a 150-ohm resistance over the nominal 3 kHz passband.

5.3.6.2.2 Balanced interface.
When a balanced interface is provided, the audio input impedance shall be a nominal 600 ohms, balanced with respect to ground, with a minimum return loss of 26 dB against a 600-ohm resistance over the frequency range of 300 Hz to 3050 Hz. The electrical symmetry shall be sufficient to suppress longitudinal currents at least 40 dB below the reference signal level.

5.3.7 Transmitter output load impedance.
The nominal rf output load impedance at interface point B in Figure 2 shall be 50 ohms, unbalanced with respect to ground. Transmitters shall survive any voltage standing wave radio (VSWR) at point B, while derating the output power as a function of increasing VSWR. However, the transmitter shall deliver full rated forward power into a 1.3:1 VSWR load. Figure 10 is a design objective for the derating curve. The VSWR between an exciter and an amplifier shall be less than 1.5:1. The VSWR between an amplifier and an antenna coupler shall be less than 1.5:1 for fixed applications and less than 2.0:1 for tactical application.

5.4.1.7 Receiver sensitivity.
The sensitivity of the receiver over the operating frequency range, in the sideband mode of operation (3-kHz bandwidth), shall be such that a -111 dBm (DO: -121 dBm) unmodulated signal at the antenna terminal, adjusted for a 1000 Hz audio output, produces an audio output with a SINAD of at least 10 dB over the operating frequency range.

5.4.1.8 Receiver out-of-band IMD.
Second-order and higher-order responses shall require a two-tone signal amplitude with each tone at -30 dBm or greater (-36 dBm or greater for tactical applications), to produce an output SINAD equivalent to a single -110 dBm tone. This requirement is applicable for equal-amplitude input signals with the closest signal spaced 30 kHz or more from the operating frequency.

5.4.1.9 Third-order intercept point.
Using test signals within the first IF passband, the worst-case third-order intercept point shall not be less than +10 dBm (+1 dBm for tactical applications).

5.4.2 Receiver distortion and internally generated spurious outputs.

5.4.2.1 Overall IMD (in-channel).
The total of IMD products, with two equal-amplitude, in-channel tones spaced 110 Hz apart, present at the receiver rf input, shall meet the following requirements. However, for frequency division multiplex (FDM) service, the receiver shall meet the requirements for any tone spacing equal to or greater than the minimum between adjacent tones in any FDM library. The requirements shall be met for any rf input amplitude up to 0 dBm PEP (-6 dBm/tone) at rated audio output. All IMD products shall be at least 35 dB (DO: 45 dB) below the output level of either of the two tones.

5.4.2.2 Adjacent-channel IMD.
For multiple-channel equipment, the overall adjacent-channel IMD in each 3 kHz channel being measured shall not be greater than -35 dBm at the 3 kHz channel output with all other channels equally loaded with 0 dBm unweighted white noise.

5.4.2.3 Audio frequency total harmonic distortion.
The total harmonic distortion produced by any single-frequency rf test signal, which produces a frequency within the frequency bandwidth of 300 Hz to 3050 Hz shall be at least 25 dB (DO: 35 dB) below the reference tone level with the receiver at rated output level. The rf test signal shall be at least 35 dB above the receiver noise threshold.

5.4.2.4 Internally generated spurious outputs.
For 99 percent of the available 3 kHz channels, internally generated spurious signals shall not exceed -112 dBm. For 0.8 percent of the available 3 kHz channels, spurious signals may exceed –112 dBm but shall not exceed -100 dBm for tactical applications and -106 dBm for fixed applications. For 0.2 percent of the available 3 kHz channels, spurious signals may exceed these levels.

5.5 ALE.

5.5.1 Basic ALE (2G).

If ALE is to be implemented, it shall be in accordance with appendix A. The ALE requirements include selective calling and handshake, link quality analysis and channel selection, scanning, and sounding. These requirements are organized in Appendix A as follows:

a. Requirements for ALE implementation are given in sections A-1 through A-4.

b. Detailed requirements on ALE waveform, signal structure, protocols, and ALE control function (orderwire messages) are contained in section A-5.

5.5.2 3G ALE.

This improved, more capable ALE may be implemented in addition to, but not in lieu of, Basic ALE. The technical requirements for 3G ALE are contained in Appendix C.

5.6 LP.

If linking protection is required to be implemented, it shall be in accordance with appendix B. These requirements are organized in Appendix B as follows:

a. General requirements for LP implementation are given in sections B-1 through B-4.

b. Detailed requirements on how to implement LP are given in section B-5.

c. The unclassified application level (AL-1) is the lowest level of LP and is mandatory for all protected radios implementing LP.

d. The unclassified enhanced application level (AL-2) is the highest level of LP covered in Appendix B. The algorithms for the higher levels of LP, application levels AL-3 and AL-4, are defined in National Security Agency (NSA) classified documents.

e. The 24-bit Lattice algorithm for linking protection applies to 2nd generation systems (Appendix B, section B.5.6) and the SODARK algorithm applies to 3rd generation systems (Appendix B, section B.5.7).

5.7 ALE control functions (orderwire functions).

See Appendix A, paragraphs A 5.6 and A 5.7.

5.8 Networking functions.

See Appendix D.

5.9 Network management.

See Appendix D.

5.10 HF application interface.

See Appendix E.

ALE
Automatic sounding
Baseline mode
Deep interleaving
Forward error correction
Golay coding
Leading redundant word
Linking protection
LQA
Network functions
Network management
Protection interval
Radio frequency scanning
Selective calling
Slotted responses
Star net and group
Triple redundant words
Word phase

6.5 International standardization agreements.
Certain provisions of this standard in paragraphs 4.2, 4.4, 5.2, 5.3, and 5.4 are the subject of international standardization agreements, STANAGs 4203, 4539, and 5035, and QSTG 733. When change notice, revision, or cancellation of this standard is proposed that will modify the international agreement concerned, the preparing activity will take appropriate action through international standardization channels, including departmental standardization offices, to change the agreement or make other appropriate accommodations.

6.6 Electromagnetic compatibility (EMC) requirements.
All services and agencies are responsible for their own EMC programs, which are driven by their user requirements and doctrine.

HF radio has significant inherent EMC implications that require serious consideration by designers, users, and acquisition personnel. It is strongly recommended that all users of this standard refer to the following documents prior to design or acquisition of HF radio systems or equipment:

a. MIL-STD-461, Requirements for the Control of Electromagnetic Interface Emissions and Susceptibility.

b. MIL-STD-462, Measurement of Electromagnetic Interference Characteristics.

c. MIL-HDBK-237, Electromagnetic Compatibility Management Guide for Platform, Systems and Equipment.

The applicable portions of these documents should be included in any acquisition actions for HF radio systems or equipment.

TABLE OF CONTENTS (continued)

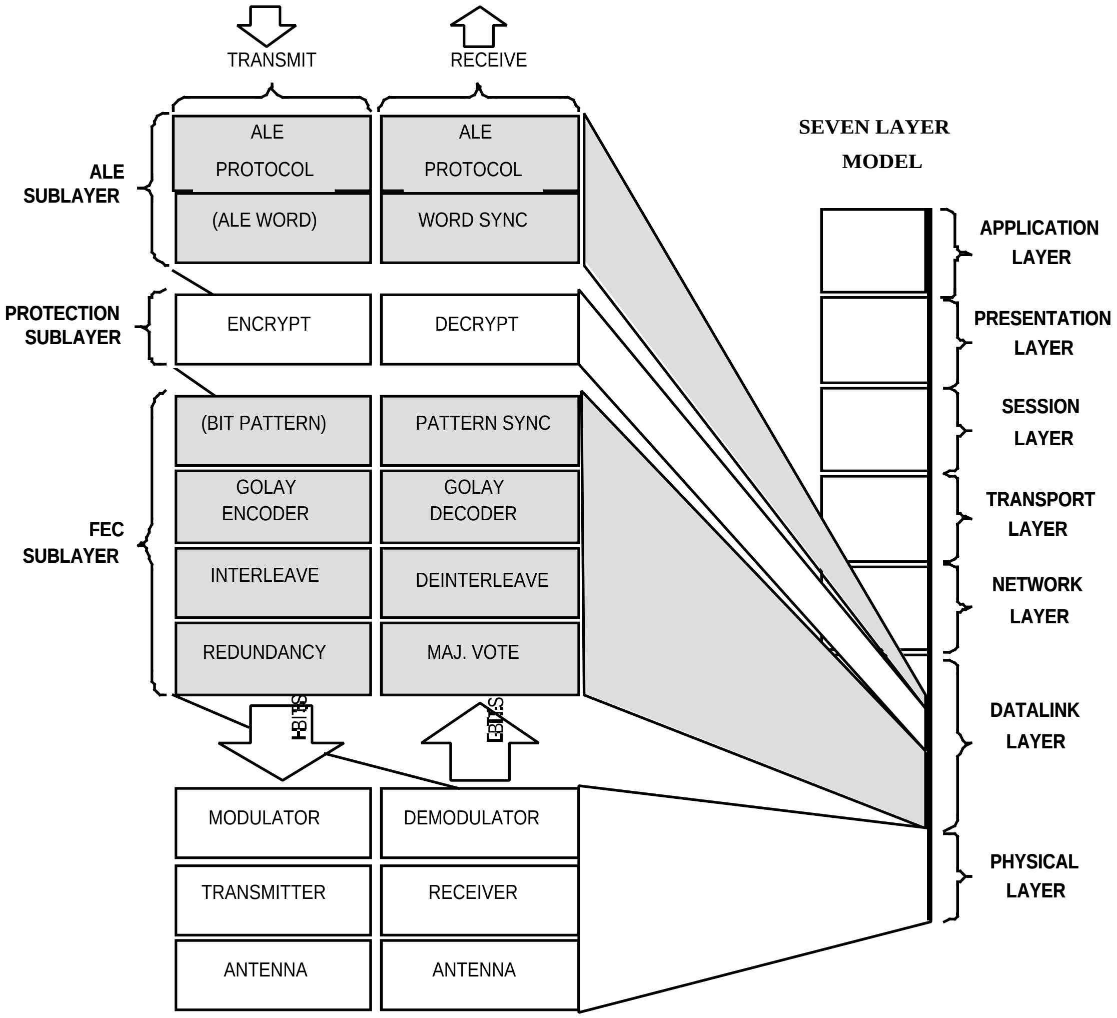


FIGURE A-1. Data link with ALE and FEC sublayers.

A.4.1.3 Calling.
Upon request by the operator or an external automated controller, the radio system shall execute the appropriate calling protocol specified in A.5.5.

A.4.1.4 Channel evaluation.
The radio system shall be capable of automatically transmitting ALE sounding transmissions in accordance with A.5.3, and shall automatically measure the signal quality of ALE receptions in accordance with A.5.4.1.

A.4.1.5 Channel quality display.
If an operator display is provided, the display shall indicate the signal-plus-noise-plus-distortion to noise-plus-distortion (SINAD) ratio in 1-dB steps in the range 0 through 30 dB, with off-scale SINAD shown as the nearest value (0 or 30). Unknown SINAD shall be indicated as 31.

A.4.2 System performance requirements.
Stations designed to this appendix shall demonstrate an overall system performance equal to or exceeding the following requirements.

A.4.2.1 Scanning rate.
Stations designed to this appendix shall incorporate selectable scan rates of two and five channels per second, and may also incorporate other scan rates (design objective (DO): 10 channels per second).

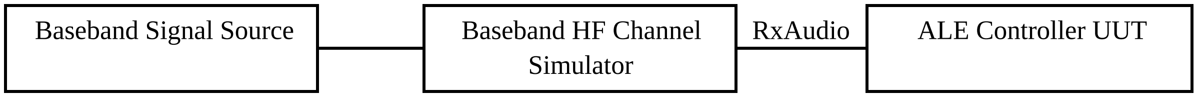
A.4.2.1.1 Alternative Quick Call (AQC) .
In the optional AQC-ALE protocol, the system shall be capable of variable dwell rates while scanning such that traffic can be detected in accordance with table A-II Probability of Linking.

A.4.2.1.2 Recommendation.
Radios equipped with the optional AQC-ALE shall provide scanning at scan rates of two channels per second or five channels per second for backward compatibility to non-AQC-ALE networks.

A.4.2.2 Occupancy detection.
Stations designed to this appendix shall achieve at least the following probability of detecting the specified waveforms (See A.5.4.7) under the indicated conditions, with false alarm rates of no more than 1 percent. The channel simulator shall provide additive white gaussian noise (AWGN) without fading or multipath (MP). See table A-I.

TABLE A-I. Occupancy detection probability (2G and 3G).

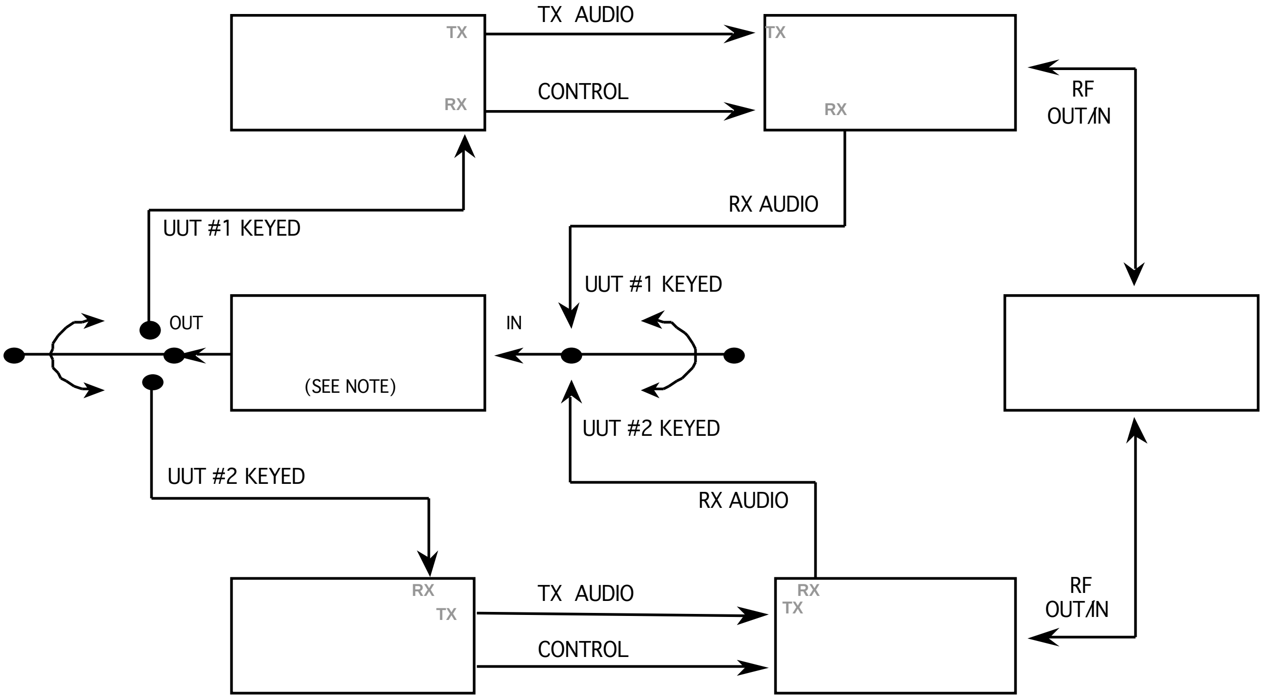
Waveform	SNR (dB in 3 kHz)	Dwell Time (s)	Detection Prob
ALE	0	2.0	0.80
	6	2.0	0.99
SSB Voice	6	2.0	0.80
	9	2.0	0.99
MIL-STD-188-110	0	2.0	0.80
(Serial Tone PSK)	6	2.0	0.99
STANAG 4529	0	2.0	0.80
	6	2.0	0.99
STANAG 4285	0	2.0	0.80
	6	2.0	0.99



NOTES:

1. The single side-band (SSB) voice test signal shall be taken from the Wireless Network Samples CD NMSU-EE-CD-021.
2. The PSK test signal shall be taken from the Wireless Network Samples CD NMSU-EE-CD-021.

FIGURE A-2. Occupancy detection test setup.



THE SIMULATOR INCLUDES EITHER INTERNAL OR EXTERNAL CAPABILITY TO ADJUST/MONITOR SIGNAL/NOISE/DOPPLER-OFFSET SETTINGS AND SHALL INCORPORATE APPROPRIATE FILTERING TO LIMIT THE AUDIO PASSBAND TO 300 - 3050 Hz.

FIGURE A-3. System performance measurements test setup.

A.4.2.3 Linking probability.

Linking attempts made with a test setup configured as shown in figure A-3, using the specified ALE signal created in accordance with this appendix, shall produce a probability of linking as shown in table A-II.

A.4.4 ALE operational rules.
The ALE system shall incorporate the basic operational rules listed in table A-V. Some of these rules may not be applicable in certain applications. For example, "always listening" is not possible while transmitting with a transceiver or when using a common antenna with a separate transmitter and receiver.

TABLE A-V. ALE operational rules.

1) Independent ALE receive capability (in parallel with other modems and simular audio receivers) (critical).
2) Always listening (for ALE signals) (critical).
3) Always will respond (unless deliberately inhibited).
4) Always scanning (if not otherwise in use).
5) Will not interfere with active channel carrying detectable traffic in accordance with table A-I (unless this listen call function is overriden by the operator or other controller).
6) Always will exchange LQA with other stations when requested (unless inhibited), and always measures the signal quality of others.
7) Will respond in the appropriate time slot to calls requiring slotted responses.
8) Always seek (unless inhibited) and maintain track of their connectivities with others.
9) Linking ALE stations employ highest mutual level of capability.
10) Minimize transmit and receive time on channel.
11) Automatically minimize power used (if capable).
NOTE : Listed in order of precedence.

A.4.5 Alternate Quick Call ALE (AQC-ALE).

A.4.5.1 Introduction.
This feature may be implemented in addition to the basic ALE functionality described in this appendix. The AQC-ALE provides a link establishment technique that requires significantly less time to link than the baseline ALE system. This is accomplished by some additional technology and trading-off some of the lesser used functions of the baseline system, for a faster linking process. The AQC-ALE shall always be listening for the baseline ALE call and shall automatically respond and operate in that mode when called.

A.4.5.2 General signaling strategies.
The AQC-ALE format employs the following characteristics:

a. Packs three address characters (21 bits) into a 16-bit value

b. Addresses are reduced from a maximum of 15 characters to 6 characters

c. Six (6) address characters are sent in every transaction

d. Replaces two seldom used preambles as follows:

- FROM preamble becomes PART2 indicating the 2nd address word
- THRU preamble becomes INLINK indicating a linked transaction

e. Isolates station addresses from message portion of the signaling structure:

- TO, TIS, TWAS, INLINK, PART2 preambles used for addressing
- CMD, DATA, and REP are used for messaging

f. Easy separation of second generation basic ALE and AQC-ALE protocols:

- Fixes 1 bit of any address word
- Prevents legitimate addresses in AQC-ALE from being legitimate addresses in second generation basic ALE.

g. Provides at least eight information bits per transmission

A.4.5.3 Features supported by AQC-ALE.
The following basic ALE features are fully implemented using the AQC-ALE protocol.

NOTE: A station operating in AQC-ALE can respond to any call type, but a station equipped with only second generation basic ALE will not respond to AQC-ALE protocol forms.

a. Linking protection levels 0, 1, 2, 3

b. Unit calls

c. Star Net calls

d. Allcalls

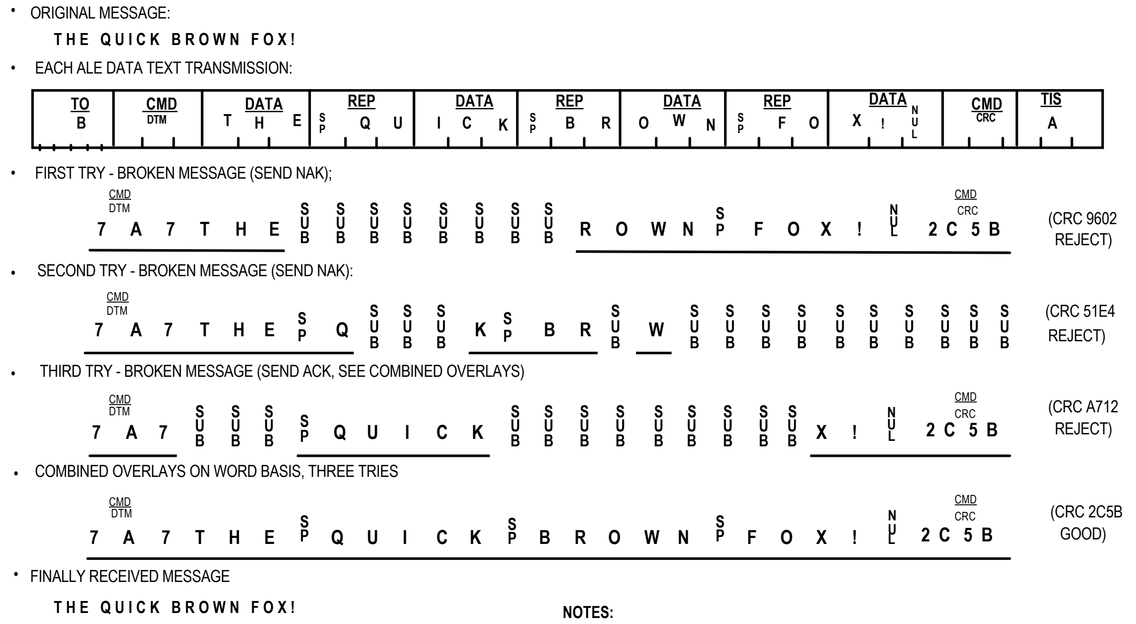
e. AnyCalls

f. LQA Exchange as part of the call handshake

g. Supports Orderwire and Relay features while in a link:

- automatic message display (AMD), data text message (DTM) or DBM
- User Unique Functions (UUF) when in a link
- Call Relay features
- Time of day and Network Management

h. Sounds are shortened to include scan time + 50percent


ORIGINAL MESSAGE:
THE QUICK BROWN FOX!
EACH ALE DATA TEXT TRANSMISSION:
TO B
CMD DTM
DATA T H E
REP SP Q U
DATA I C K
REP SP B R
DATA O W N
REP SP F O
DATA X ! NUL
CMD CRC
TIS A
FIRST TRY - BROKEN MESSAGE (SEND NAK);
CMD DTM 7 A 7 T H E SUB SUB SUB SUB SUB SUB SUB SUB R O W N SP F O X ! NUL CMD CRC 2 C 5 B
(CRC 9602 REJECT)
SECOND TRY - BROKEN MESSAGE (SEND NAK):
CMD DTM 7 A 7 T H E SP Q SUB SUB SUB K SP B R SUB W SUB SUB SUB SUB SUB SUB SUB SUB SUB SUB
(CRC 51E4 REJECT)
THIRD TRY - BROKEN MESSAGE (SEND ACK, SEE COMBINED OVERLAYS)
CMD DTM 7 A 7 SUB SUB SUB SP Q U I C K SUB SUB SUB SUB SUB SUB SUB SUB SUB X ! NUL CMD CRC 2 C 5 B
(CRC A712 REJECT)
COMBINED OVERLAYS ON WORD BASIS, THREE TRIES
CMD DTM 7 A 7 T H E SP Q U I C K SP B R O W N SP F O X ! NUL CMD CRC 2 C 5 B
(CRC 2C5B GOOD)
FINALLY RECEIVED MESSAGE
THE QUICK BROWN FOX!

NOTES:

1. CMD DTM IN THIS EXAMPLE INDICATES SEVEN WORDS WITH ASCII CHARACTERS AND SEVEN STUFF BITS IN THE LAST WORD.

2. SUB INDICATES SUBSTITUTE CHARACTERS WHEN BAD AND REJECTED.

3. SP INDICATES SPACE FUNCTION.

4. CMD CRC CONTAINS FOUR HEXADECIMAL CHARACTERS CONSTITUTING THE 16-BIT FRAME CHECK SEQUENCE.

FIGURE A-48. Data text message reconstruction (overlay).

During reception of ALE frames and DTM data blocks, it is expected that fades, interferences, and collisions will occur. The receiving station shall have the capability to maintain synchronization with the frame and the DTM data block transmission, once initiated. It shall also have the capability to read and process any colliding and significantly stronger (that is, readable) ALE signals without confusing them with the DTM signal (basic ALE reception in parallel, and always listening). Therefore, useful information that may be derived from readable collisions of ALE signals should not be arbitrarily rejected or wasted. The DTM structures, especially the DTM EXTENDED, can tolerate weak signals, short fades, and short noise bursts. For these cases and for collisions, the DTM protocol can detect DTM words that have been damaged and "tag" them for error correction or repeats. The DTM constructions are described herein. Within the DTM data block structure, the CMD DTM word shall be placed ahead of the DTM data block itself. The DTM word shall alert the receiving station that a DTM data block is arriving, how format or pattern; and when KB3 is set to “1” the message data is 7-bit ASCII characters. For DBM ARQ, flow control bit KB3 is set to “0” to indicate that the DBM transfer flow should continue or resume; and when KB3 is set to “1” it indicates that the sending station should pause (until another and identical DBM ARQ is returned, except that KB3 shall be “0”).

For DBM BASIC, EXTENDED, and NULL, when the “message” control bit KB2 (W13) is set to the same value as the KB2 in any sequentially adjacent DBM data block, the message data contained within those adjacent blocks (after individual error control) shall be recombined with the message data within the present DBM data block to reconstitute (segment-by-segment) the original whole message; and when KB2 is set opposite to any sequentially adjacent DBM data blocks, those data blocks contain separate message data and shall not be combined. For DBM ARQ, “message” control bit KB2 shall be set to match the referenced DBM data block KB2 value to provide message confirmation.

For DBM BASIC, EXTENDED, and NULL, the sequence control bit KB1 (W14) shall be set opposite to the KB1 value in the sequentially adjacent DBM BASIC, EXTENDED, or NULLs be sent (the KB1 values therefore alternate, regardless of their message dependencies). When KB1 is set the same as any sequentially adjacent DBM sent, it indicates a duplicate. For DBM ARQ, sequence control bit KB1 shall be set to match the referenced DBM data block or NULL KB1 value to provide sequence confirmation.

When used for the DBM protocols, the ten DBM data code (BC) bits BC10 through BC1 (W15 through W24) shall indicate the DBM mode (BASIC, EXTENDED, ARQ, or NULL). They shall also indicate the size of the message data and the length of the data block. The DBM NULL BC value shall be “0” (0000000000), and it shall designate the single CMD DBM NULL word. The DBM EXTENDED BC values shall range from “1” (0000000001) to “445” (0110111101), and they shall designate the CMD DBM EXTENDED word and the data block multiple (of 49 INTERLEAVER DEPTH) which defines the variable data block sizes, in increments of 588 binary bits or 84 ASCII characters. The DBM BASIC BC values shall range from “448” (0111000000) to “1020” (1111111100), and they shall designate the CMD DBM BASIC word and the exact size of the message data in a fixed size (INTERLEAVER DEPTH = 49) data block, with up to 572 binary bits or 81 ASCII characters. The DBM ARQ BC value shall be “1021” (1111111101), and it shall designate the single CMD DBM ARQ word.

NOTES:
1. The values “446” (0110111110) and “447” (0110111111) are reserved.
2. The values “1022” (1111111110) and “1023” (1111111111) are reserved until standardized (see table A-XXXIV).

A.5.8 AQC (optional) .
AQC-ALE is designed to use shorter linking transmissions than those of baseline second generation ALE (2G ALE) described previously in this appendix. AQC-ALE uses an extended

version of the 2G ALE signaling structure to assure backward compatibility to already fielded radios. Special features of AQC-ALE include the following:

- The signaling structure separates the call attempt from the inlink-state transactions. This allows radios that are scanning to detect and exit a channel that is carrying traffic that is of no interest.
- The address format is a fixed form to allow end of address detection without requiring the last word wait timeout.
- Control features distinguish call setup channels from traffic carrying channels.
- Local Noise Reports are inherent in the sound and call setup frames to minimize the need to sound as frequently.
- Resources that are needed during the linked state can be identified and bid for during the link setup. This provides a mechanism to bid for needed resources during linking.

A.5.8.1 Signaling structure .

The AQC-ALE signaling structure is identical to that described previously in this appendix, except as provided below and in the remaining subsections of this section:

- The AQC-ALE word is encoded differently (see A.5.8.1.1).

A.5.8.1.1 AQC-ALE word structure.

The AQC-ALE word shall consist of a three-bit preamble, an address differentiation flag, a 16-bit packed address field, and a 4-bit Data Exchange field. These fields shall be formatted and used as described in the following paragraphs. Every AQC-ALE word shall have the form shown in figure A-52, AQC-ALE Word. The data values associated with a particular AQC-ALE word are defined by the context of the frame transmission (see A.5.8.2).

A.5.8.1.1.1 Packed address.

AQC-ALE packs the 21 bits representing three address characters in the 38-character ASCII subset into 16 bits. This is performed by assigning an ordinal value between 0 and 39 to each member of the 38-character subset. Base 40 arithmetic is used to pack the mapped data into a 16-bit number. The ASCII characters used for addressing shall be mapped to the values defined in table A-XXXVI, Address Character Ordinal Values, with character 1's value multiplied by 1600, Character 2’s value multiplied by 40, and Character 3’s value multiplied by 1. The sum of the three values shall be used as the 16-bit packed address (see example below).

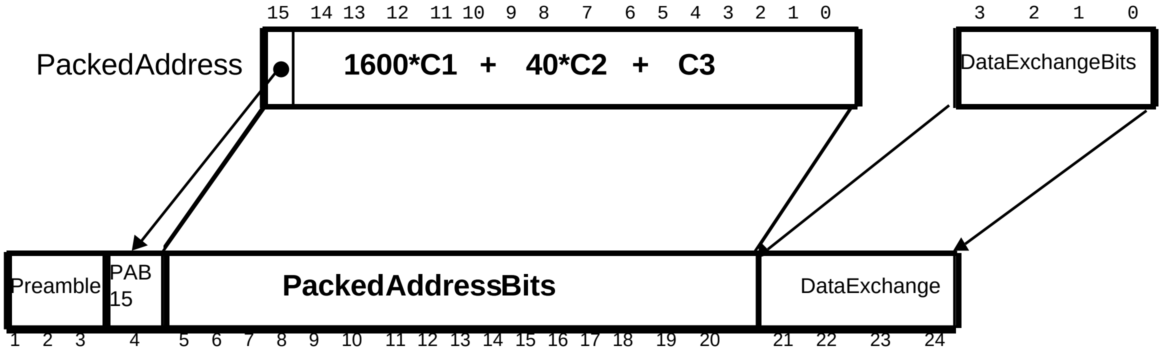


FIGURE A-52. AQC-ALE data exchange word.

TABLE A-XXXVI. AQC address character ordinal value.

Character	Value
*	0
0 to 9	1 to 10
?	11
@	12
A to Z	13 to 38
_ (Underscore)	39

Note: The "*" and "_" characters are not part of the standard ALE ASCII-38 character set. These characters shall not be used in station addresses in any network that is required to interoperate with stations that support only baseline 2G ALE.

Example:

Using table A-XXXVI, the address 'ABC' would be computed as:

(Value('A') * 1600) + (Value('B')* 40) + Value('C')
which is
(13 * 1600) + (14 * 40) + 15 = 21,375

The smallest valued legal address is "000" for a packed value of ➔	1,641
A legal address such as "ABC" would have a packed value of ➔	21,375
The largest valued legal address is "ZZZ" for a packed value of ➔	62,358

A.5.8.1.1.2 Address differentiation flag.
Bit 4 of the AQC-ALE word shall be a copy of the most significant bit of the 16-bit packed address. This combination results in no legal address in AQC-ALE being legal in baseline 2G

ALE and vice versa. The packed address shall occupy the next 16 bits of the 21-bit data portion of the address.

A.5.8.1.2 Preambles .
The preambles shall be as shown in table A-XXXVII AQC-ALE word types (and preambles)

TABLE A-XXXVII. AQC-ALE word types (and preambles).

Word Type	Code Bits	Functions	Significance
INLINK	001	direct routing	Transaction for linked members
TO	010	--	See table A-VIII
CMD	110	--	See table A-VIII
PART2	100	direct routing	indicates this is the second part of the full AQC-ALE address
TIS	101	--	See table A-VIII
TWAS	011	--	See table A-VIII
DATA	000	extension of information	Used only in message section to extend information being sent
REP	111	duplication and extension of information	Used only in message section to extend information being sent

A.5.8.1.2.1 TO.
This preamble shall have a binary value of 010 and is functionally identical to the TO preamble in A.5.2.3.2.1. The AQC-ALE TO preamble shall represent the first of two words identifying the address of the station or net.

A.5.8.1.2.2 THIS IS (TIS).
This preamble shall have a binary value of 101. The preamble is functionally identical to the TIS preamble in A.5.2.3.2.2. The AQC-ALE TIS preamble identifies the AQC-ALE word as containing the first three characters of the of the calling or sounding station address.

A.5.8.1.2.3 THIS WAS (TWAS).
This preamble shall have a binary value of 011. This preamble is functionally identical to the TWAS preamble in A.5.2.3.2.3. The AQC-ALE TWAS preamble identifies the AQC-ALE word as containing the first three characters of the of the calling or sounding station address.

A.5.8.1.2.4 PART2.
This preamble shall have a binary value of 100. This preamble is shared with the baseline 2G ALE preamble of FROM. This preamble identifies the second set of three characters in an AQC-ALE address. This preamble shall be used for the second word of every AQC-ALE packed address transmission.

A.5.8.1.2.5 INLINK.
This preamble shall have a binary value of 001. This preamble is shared with the baseline 2G ALE preamble of THRU. This preamble shall be used by AQC-ALE whenever a transmission to stations already in an established link is required. This preamble identifies the AQC-ALE word

as containing the first three characters of the transmitting station address. This preamble may also be used in the acknowledgement frame of a three-way handshake as described in A.5.8.2.3.

A.5.8.1.2.6 COMMAND.
No Change to A.5.2.3.3.1

A.5.8.1.2.7 DATA.
See A.5-2.3.4.1. In the AQC-ALE word, this preamble never applies to a station address.

A.5.8.1.2.8 REPEAT.
See A.5-2.3.4.2. In the AQC-ALE word, this preamble never applies to a station address.

A.5.8.1.3 AQC-ALE address characteristics .

A.5.8.1.3.1 Address size.
Addresses shall be from 1 to 6 characters.

A.5.8.1.3.2 Address character set.
The address character set shall be the same ASCII-38 character set as for baseline 2G ALE.

A.5.8.1.3.3 Support of ISDN (option) (NT).
To support an ISDN address requirement, the station shall be capable of mapping any 15 character address to and from a 6 character address for displaying or calling. This optional mapping shall be available for at least one Self Address and all programmed Other Addresses in the radio.

A.5.8.1.3.4 Over-the-air address format.
A two AQC-ALE word sequence shall be broadcast for any AQC-ALE address. The "@" shall be used as the stuff character to complete an address that contains fewer than six characters. The sequence shall be an AQC-ALE word with the preamble TO, TIS, TWAS, or INLINK for the first three characters of the address followed by an AQC-ALE word with the preamble PART2 for the last three address characters.

A.5.8.1.4 Address formats by call type.

A.5.8.1.4.1 Unit addresses.
A unit or other address shall be from one to six characters.

A.5.8.1.4.2 StarNet addresses.
A StarNet address shall be from one to six characters.

A.5.8.1.4.3 Group addresses.
This feature is not applicable to AQC-ALE.

A.5.8.1.4.4 AllCall address.
AQC-ALE AllCall address shall be six characters. The second three characters of the AllCall address shall be the same as the first three characters. Thus, a global AllCall sequence would look like:

TO-**@?@**|PART2-**@?@**.

A.5.8.1.4.5 AnyCall address.
AQC-ALE AnyCall address shall be six characters. The second three characters of the AnyCall address shall be the same as the first three characters. Thus, a global AnyCall sequence would look like:

TO-**@@?**|PART2-**@@?**.

A.5.8.1.5 Data exchange field.
The 4-bit data exchange field shall be encoded as described in Table A-XXXVIII and the following paragraphs. The use of the various encodings DE(1) through (9) shall be as shown in the figures for the Sound, Unit call, Starnet call, All call, and Any call in the respective subsections of A.5.8.2.

NOTE: A station may use the contents of the data exchange field to further validate the correctness of a given frame.

TABLE A-XXXVIII. Data exchange definitions.

	Bit 3	Bit 2	Bit 1	Bit 0	Description
DE(1)	1	1	1	1	No Data Available
DE(2)	x	x	x	x	Number of TOs Left in Calling Cycle Section
DE(3)	x	x	x	x	Inlink Resource List Expected
DE(4)	x	x	x	x	Local Noise Index
DE(5)	0	< BER Range>			BER estimate
DE(6)	x	x	x	x	LQA Measurement Index
DE(7)	x	x	x	x	Number of Tis/Twas left in Sound
DE(8)	Ack This	<# of Command Preambles>			Most Significant Bits of the Inlink Transaction Code
DE(9)	I'm Inlink	< Transaction Code >			Least significant 4 bits of Inlink

A.5.8.1.5.1 DE(1) no data available.
DE(1) shall be sent in the TIS word in the conclusion of a Call frame. All data bits shall be set to 1s.

A.5.8.1.5.2 DE(2) number of TO words left in calling cycle.
DE(2) shall be sent in every AQC-ALE word that contains a TO preamble. In a Call frame, the DE(2) field shall indicate the remaining number of TO preambles that remain in the frame. This is an inclusive number and when set to a value of 1 the next address shall be the caller’s address

using a TIS or TWAS preamble. When the remaining call duration would require a count greater than 15, a count of 15 shall be used.

A value of 0 shall be used in in the Response frame and Acknowledgement frame when a single address in required. DE(2) shall count down to 1 whenever multiple addresses are transmitted in an address section.

A.5.8.1.5.3 DE(3) Inlink resource list.

DE(3) shall be sent in the PART 2 word that follows each TO word. The DE(3) field shall indicate the type of traffic to be conveyed during the Inlink state, using the encodings in table AXXXIX. Values not specified in the table are reserved, and shall not be used until standardized.

Upon receipt of the INLINK Resource List in the Call, the called station shall determine whether the station can operate with the desired resource. When responding to the call, the called station shall honor the requested resource whenever possible. If the resource requested is unavailable, the called unit shall respond with an alternate resource that is the best possible alternative resource available to the receiver. This information is provided in the Response frame of a handshake.

By definition, when the calling station enters an Inlink state with the called station, the calling station accepted the Inlink resource that the called station can provide.

TABLE A-XXXIX. Inlink resource list.

Value	Meaning	Alternate Resource
0	Clear Voice	15
1	Digital Voice	0
2	High Fidelity Digital (HFD) Voice	1 or 0
3	Reserved	NA
4	Secure Digital Voice	2, 1, 0
5	Secure HFD Voice	4, 2, 1, 0
6	Reserved	NA
7	Reserved	NA
8	ALE Messaging	15
9	PSK Messaging	0 or 15
10	39 Tone Messaging	0 or 15
11	HF Email	9, 8, 0
12	KY-100 Data Security Active	9
13	Reserved	NA
14	Reserved	NA
15	Undeclared Traffic. Usually a mixture.	Always Acceptable

A.5.8.1.5.4 DE(4) local noise report.

DE(4) shall be sent in the PART 2 word that concludes a Call frame and in every PART 2 word in a Sounding frame. The Local Noise Report contains information which describes the type of

A.5.8.1.5.5 DE(5) BER Range.
DE(5) shall be sent in the TIS or TWAS word in the conclusion of AQC-ALE Response and Acknowledgement frames. It shall report the signal quality variation measured on the immediately preceding transmission of the handshake.

Whenever an AQC-ALE or ALE word is received, a bit error ratio (BER) estimate shall be computed by counting non-unanimous votes in accordance with paragraph A.5.4.1.1. This measurement can be used to determine the capacity of the channel to handle traffic. The DE(5) LQA Data Exchange word shall report the average number of non-unanimous votes in the preceding transmission; i.e., the DE(5) in the AQC-ALE Response shall report the average number of non-unanimous votes in the AQC-ALE Call, and the DE(5) in the Acknowledgement shall report the average number of non-unanimous votes in the Response.

	Bit 3	Bit 2	Bit 1	Bit 0	Description
DE(5)	0	<BER Range>			one bit spare, 3 bits of BER variation data

The average number of non-unanimous votes shall be encoded in accordance with Table A-XLI for transmission in DE(5).

TABLE A-XLI. DE(5) Encoding of BER Range.

Average Non-Unanimous Votes	Bit index
0	000
1	001
2 - 3	010
4 - 8	011
9 - 13	100
14 - 19	101
20 - 25	110
>25	111

A.5.8.1.5.6 DE(6) LQA measurement.
DE(6) shall be sent in the PART 2 word in the conclusion of AQC-ALE Response and Acknowledgement frames. The Link Quality Measurement contains the predicted quality of the channel to handle traffic. This value may be used as a first approximation to setting data rates for data transmission, determining that propagation conditions could carry voice traffic, or directing the station to continue to search for a better channel. (See A.5.8.1.5.5 for a description of the LQA.) This can also be used to determine which channels are more likely to provide sufficient propagation characteristics for the intended Inlink state traffic. Table A-XLII shall be used to

encode the measured mean SNR value. An additional column is provided suggesting possible channel usage for the given SNR value.

TABLE A-XLII. LQA scores.

Value	Measured SNR	Potential Channel Usage
0	SNR <= -6	Choose another channel
1	-6 < SNR <= -3	use 50 to 75 bps data
2	-3 < SNR <= 0	use 50 to 75 bps data
3	0 < SNR <= 3	use 150 bps data
4	3 < SNR <= 6	use 300 bps data
5	6 < SNR <= 9	use 300 bps data
6	9 < SNR <= 12	use 1200 bps data, could carry voice, digital voice, KY-100 data, secure digital voice
7	12 < SNR <= 15	use 1200 bps data, could carry voice
8	15 < SNR <= 18	use 2400 bps data, could carry voice
9	18 < SNR <= 21	use 2400 bps data, could carry good quality voice, HFD Voice, Secure HFD Voice
10	21 < SNR <= 24	use 4800 bps data, could carry high quality voice
11	24 < SNR <= 27	use 4800 bps data, could carry poor quality voice
12	27 < SNR <= 30	Very high data rates can be supported (9600 baud)
13	30 < SNR <= 33	
14	SNR > 33	
15	No Measurement Taken	Value in DE(5) shall be ignored

A.5.8.1.5.7 DE(7) number of Tis/Twas left in sounding cycle.
While transmitting the sounding frame, DE(7) shall be sent in each TIS/TWAS word to identify the remaining number of TIS/TWAS words that will be transmitted in the frame. This is an inclusive number and when set to a value of 1, only one PART2 word remains in the frame.

When the sound duration would require an initial count greater than 15, a count of 15 shall be used until the count can correctly decrement to 14. From this point, DE(7) shall count down to 1.

A.5.8.1.5.8 DE(8) inlink data definition from INLINK.
Inlink Event transaction definitions are defined by 2 data exchange words. DE(8) shall be used when the INLINK preamble is used, while DE(9) shall be used for the second half of the address begun with the INLINK preamble.

	Bit 3	Bit 2	Bit 1	Bit 0	Description
DE(8)	AckThis	<# of Command Preambles>			Most Significant Bits of the Inlink Transaction Code

A.5.8.1.5.8.1 Acknowledge this frame.
Data Bit3, ACK-THIS, when set to 1, shall indicate that the stations which are linked to the transmitting station are to generate an ACK Inlink message in response to this frame. If the address section of an Inlink transaction is present, then only the addressed stations in the link are to respond. The responding station Inlink event shall return a NAK if any CRC in the received

message fails, otherwise the Inlink event shall be an ACK. When Data Bit3 is set to 0, the transmitting station is broadcasting the information and no response by the receiving stations is required.

A.5.8.1.5.8.2 Identify command section count.
Data Bits 0-2 represent the number of command sections that are present in the frame. A value of 0 indicates no command sections are present, i.e., the frame is complete when the immediately following PART2 address word is received. A value of 1 indicates that 1 command section is present. Up to seven command sections can be transmitted in one Inlink event transaction.

A.5.8.1.5.9 DE(9) Inlink data definition from PART2.
Inlink Event transaction definitions are defined by 2 data exchange words. DE(9) is used for the second half of the address begun with the INLINK preamble.

	Bit 3	Bit 2	Bit 1	Bit 0	Description
DE(9)	I'm Inlink	<	Transaction Code	>	Least significant 4 bits of Inlink

A.5.8.1.5.9.1 I AM remaining in a link state.
Data Bit3, I'mInlink, when set to 1, shall indicate that the transmitting station will continue to be available for Inlink transactions. When set to 0, the station is departing the linked state with all associated stations. It shall be the receiver 's decision to return to scan or perform other overhead functions when a station departs from a link state. All Inlink event transactions should set this to '1' when the members of the link are to remain in the linked state.

Valid combinations of data bit ACK-THIS and I'mInlink are defined in table A-XLIII.

TABLE A-XLIII. Valid combinations of ACK-This and I'm Inlink.

Ack This Value	I'm Inlink Value	Description
0	0	Station departing linked state
0	1	Station remaining in linked state
1	0	Not valid. A station cannot leave a link and expect a response
1	1	Acknowledge this transmission.

A.5.8.1.5.9.2 Inlink event transaction code.
Data Bits 0-2 represent the type of Inlink event that is being transmitted. Table A-XLIV shall be used to encode the types of Inlink events. The Operator ACK/NAK and AQC-ALE Control Message sections are described in A.5.8.3.

TABLE A-XLIV. DE(9) inlink transaction identifier.

Value	Notes	Meaning	Message Section Count
0		Reserved	0
1		MS_141A Section Definition. Each section shall be terminated with a CRC	1 to 7
2		ACK'ng Last Transaction	0
3		NAK'ng Last Transaction	0
4	(1)	Directed Link Terminate	0
5	(1) (2)	Operator ACK/NAK	1
6	(1) (2)	AQC-ALE Control Message	1 to 7
7		Reserved	0

1. Requires that an address section (To,Part2) was received in the frame.
2. Optional Transaction Code.

A.5.8.2 AQC-ALE frame structure and protocols.

A.5.8.2.1 Calling cycle.

The calling cycle frame is used when the caller is attempting to reach a station that is scanning. Sufficient address words are repeated continuously until the scanning radio has had ample opportunity to stop on the channel. Other receivers, upon hearing an address, may recognize the presence of an ongoing call and skip processing the channel until the handshake is completed.

The calling cycle shall be composed of the target address broadcast for at least the period defined as the call duration for the radio, followed by the target address followed by the caller's (source) address. Data exchange values shall be per the specific type of call being attempted. When the call duration is not evenly divisible by 2 Trw, then an additional full address may be transmitted. When an entire address is not used to complete a fractional portion of the call duration, the caller shall begin the transmission with the second half of the target address using the PART2 preamble. In this case, the LP word number shall be 1.

When the radio is programmed to automatically derive the call duration, the equation shall be:

Number of Channels * 0.196

Table A-XLV specifies minimum and maximum number of words used for the scanning cycle section of a call. The total number of words used for calling is four additional words. The unit call time column presents the maximum time to complete a unit call as measured from the first tone transmitted by the caller to the last tone transmitted by the caller in the Acknowledgement frame. Users will see times greater than these due to call setup time, caller tune time, listen before call, and link notification delay; these may add several seconds to the response time seen by a user.

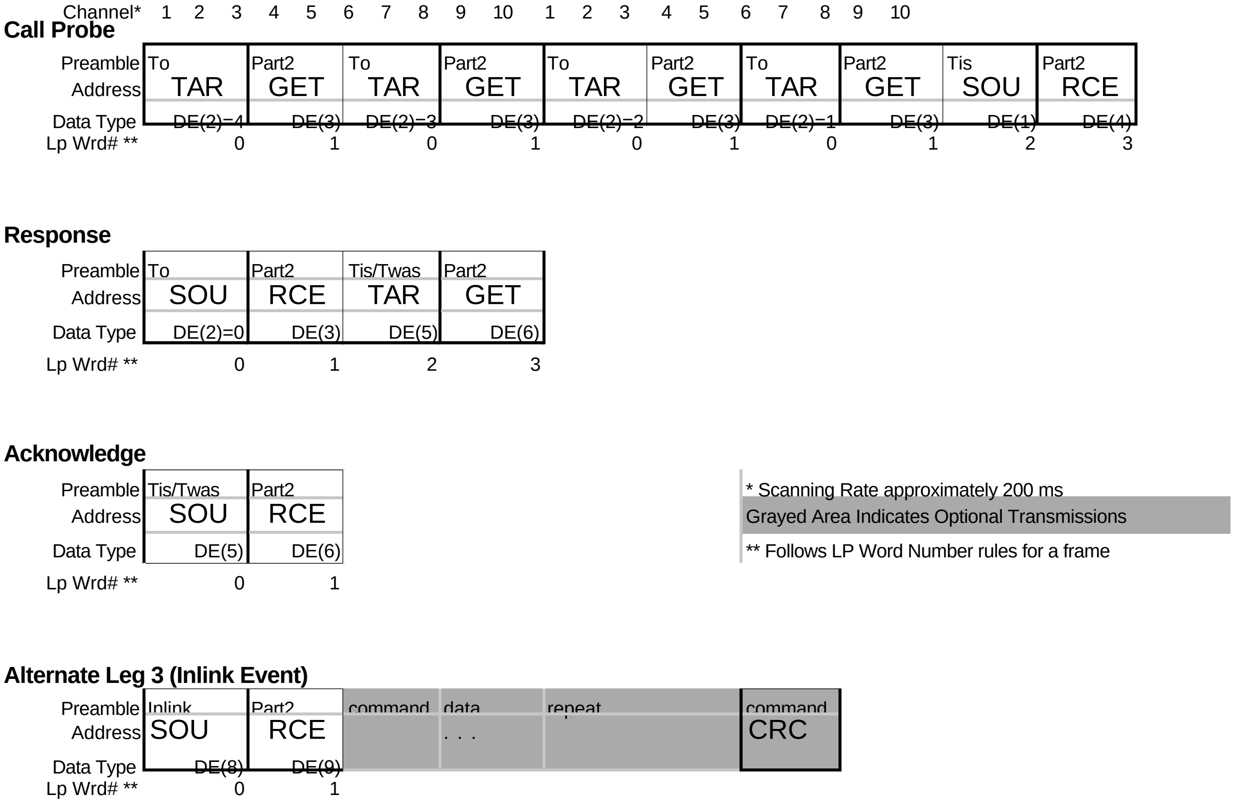
TABLE A-XLV. Scanning part duration using automated calculation.

Channels	AQC-ALE Minimum Scan Trw	AQC-ALE Maximum Scan Trw	Call Time in Seconds
1	0	0	4.8
2	1	2	5.6
3	2	2	5.6
4	2	2	5.6
5	3	4	6.4
6	3	4	6.4
7	4	4	6.4
8	4	4	6.4
9	5	6	7.2
10	5	6	7.2
11	6	6	7.2
12	6	6	7.2
13	7	8	8.0
14	7	8	8.0
15	8	8	8.0
16	8	8	8.0
17	9	10	8.8
18	9	10	8.8
19	10	10	8.8
20	10	10	8.8

A.5.8.2.2 Unit call structure.

A unit call in AQC-ALE follows the same principles as a standard ALE unit call with the following changes. In the Leading Call section of the Call and Response, the address shall appears once instead of twice. In the Acknowledgement frame, only the conclusion section shall be sent. See figure A-53 for an example of a unit call sequence from SOURCE to TARGET.

- See A.5.8.2.1, Calling Cycle to determine the maximum number of words to send during the scanning call portion of the Call.
- An Inlink Event Transaction shall be used in lieu of the Acknowledgement frame when ALE data traffic is available for the Inlink State in AQC-ALE.



Channel* 1 2 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9 10

Call Probe

Preamble	To	Part2	To	Part2	To	Part2	To	Part2	Tis	Part2
Address	TAR	GET	TAR	GET	TAR	GET	TAR	GET	SOU	RCE
Data Type	DE(2)=4	DE(3)	DE(2)=3	DE(3)	DE(2)=2	DE(3)	DE(2)=1	DE(3)	DE(1)	DE(4)
Lp Wrd# **	0	1	0	1	0	1	0	1	2	3

Response

Preamble	To	Part2	Tis/Twas	Part2
Address	SOU	RCE	TAR	GET
Data Type	DE(2)=0	DE(3)	DE(5)	DE(6)
Lp Wrd# **	0	1	2	3

Acknowledge

Preamble	Tis/Twas	Part2
Address	SOU	RCE
Data Type	DE(5)	DE(6)
Lp Wrd# **	0	1

* Scanning Rate approximately 200 ms
Grayed Area Indicates Optional Transmissions
** Follows LP Word Number rules for a frame

Alternate Leg 3 (Inlink Event)

Preamble	Inlink	Part2	command	data	repeat	command
Address	SOU	RCE		. . .		CRC
Data Type	DE(8)	DE(9)				
Lp Wrd# **	0	1				

FIGURE A-53. Example of unit call format.

A.5.8.2.3 Star net call structure.
The call probe shall be identical to a Unit call where the star net address replaces the unit address. The Slotted Response portion shall always use a two word address for the TO and TIS addresses. Just as in Baseline 2G ALE, the slotted response shall be 5 Tw wider than the 6 Tw needed to transmit the TIS/TWAS address. Slot 0 shall be 17 Tw to accommodate a non-net member participating in the call. Slot 1 and all remaining slots shall be 11 Tw wide. No LQA information shall be emitted in the Acknowledgement portion of the Start Net Call except as provided through the data exchange bits.

The Data Exchange values shall be per figure A-54.

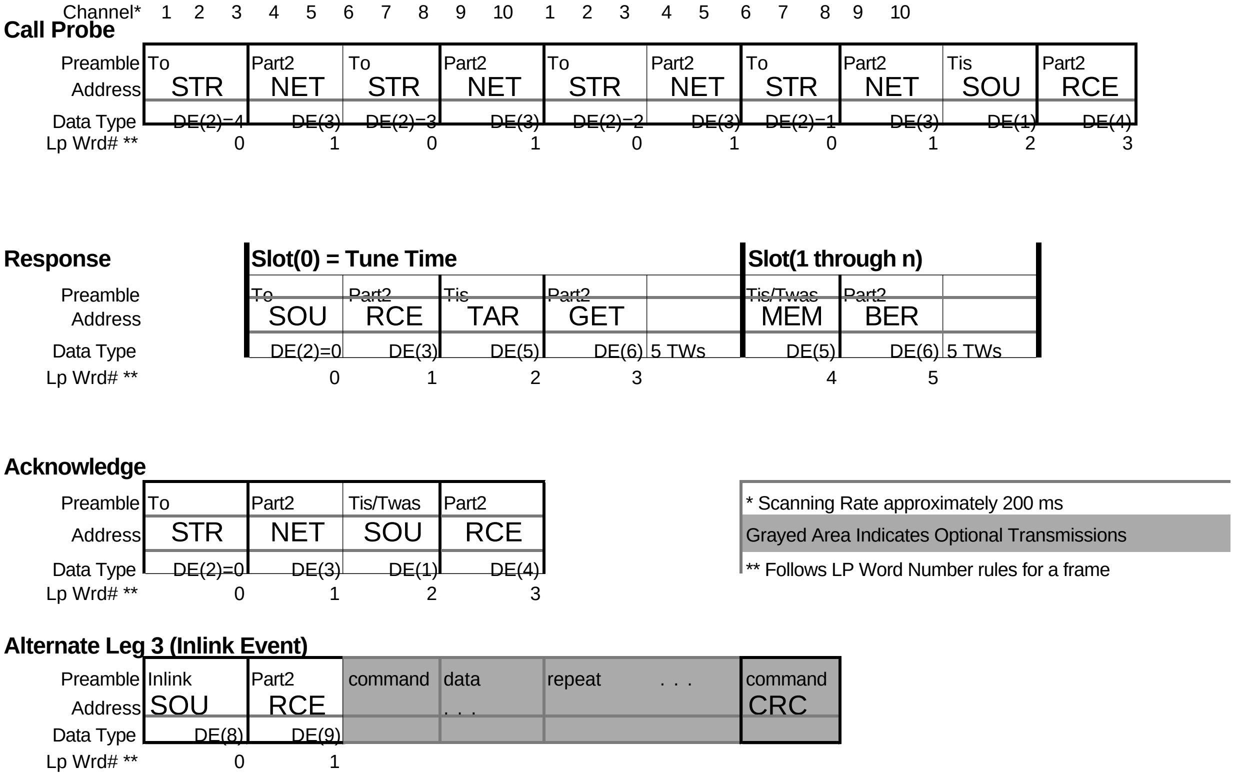


FIGURE A-54. Example of StarNet format.

An Inlink Event frame may be used for the Acknowledgement frame. Slots 1 and beyond may be expanded by fixed number of Trw for certain types of AQC-ALE Inlink Messages.

A.5.8.2.4 <u>AllCall frame formats</u>.
A station placing an AllCall shall issue the call using the calling cycle definition in A.5.8.2.1. The actions taken shall be as described for baseline 2G ALE AllCalls. The Data Exchange values shall be per figure A.-55, AllCall Frame Format. Selective AllCall shall be supported.

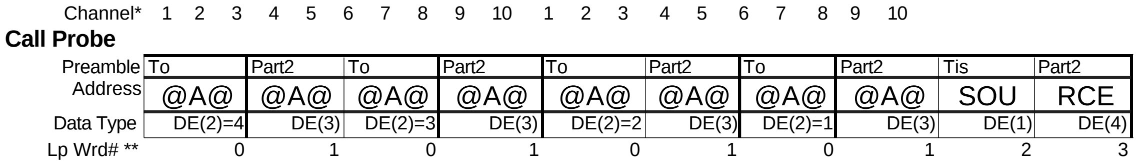
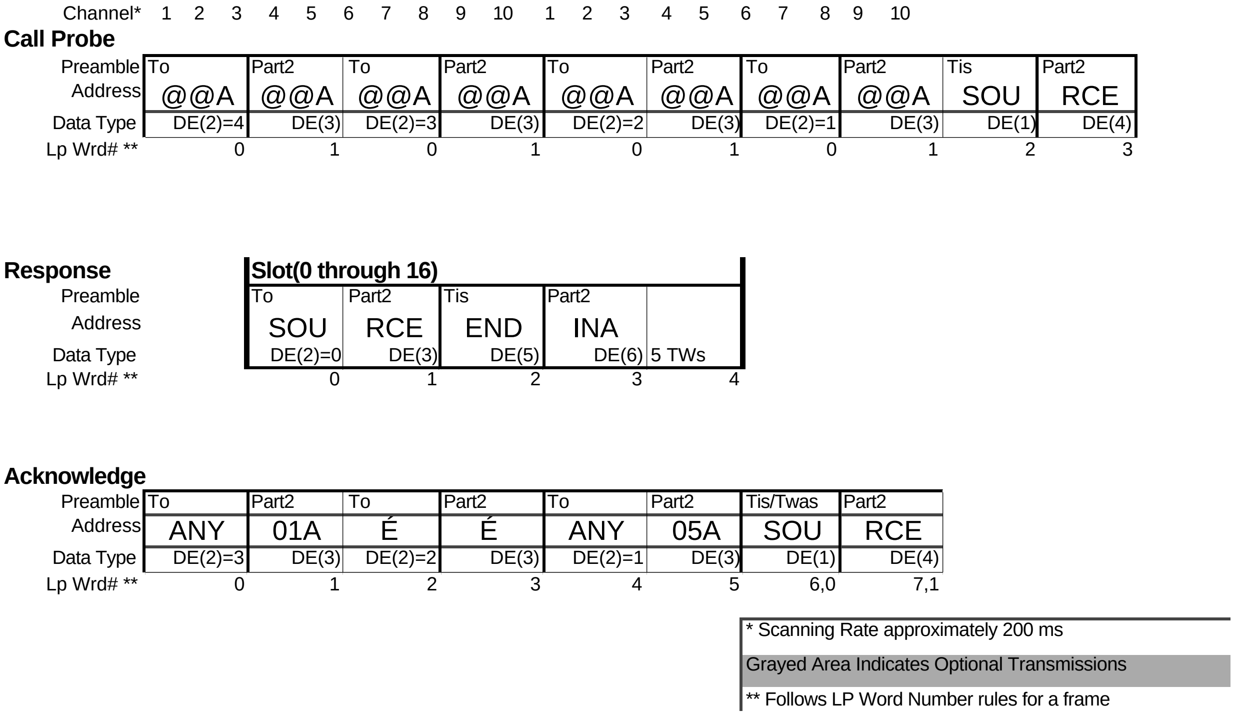


FIGURE A-55. <u>Example AllCall frame format</u>.

A.5.8.2.5 <u>AnyCall frame formats</u>.
A station placing an AnyCall shall issue the call using the calling cycle definition in A.5.8.2.1. The actions taken shall be a described for baseline 2G ALE AnyCalls except that the Slot width shall be fixed at 17 Tw. The leading address section and conclusion shall be used for each slotted

response. The Data Exchange values shall be per figure A-56. Selective AnyCall and Double Selective AnyCall shall be supported.



Channel* 1 2 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9 10

Call Probe

Preamble	To	Part2	To	Part2	To	Part2	To	Part2	Tis	Part2
Address	@@A	@@A	@@A	@@A	@@A	@@A	@@A	@@A	SOU	RCE
Data Type	DE(2)=4	DE(3)	DE(2)=3	DE(3)	DE(2)=2	DE(3)	DE(2)=1	DE(3)	DE(1)	DE(4)
Lp Wrd# **	0	1	0	1	0	1	0	1	2	3

Response

	Slot(0 through 16)				
Preamble	To	Part2	Tis	Part2	
Address	SOU	RCE	END	INA	
Data Type	DE(2)=0	DE(3)	DE(5)	DE(6)	5 TWs
Lp Wrd# **	0	1	2	3	4

Acknowledge

Preamble	To	Part2	To	Part2	To	Part2	Tis/Twas	Part2
Address	ANY	01A	É	É	ANY	05A	SOU	RCE
Data Type	DE(2)=3	DE(3)	DE(2)=2	DE(3)	DE(2)=1	DE(3)	DE(1)	DE(4)
Lp Wrd# **	0	1	2	3	4	5	6,0	7,1

* Scanning Rate approximately 200 ms

Grayed Area Indicates Optional Transmissions

** Follows LP Word Number rules for a frame

FIGURE A-56. Example AnyCall frame formats.

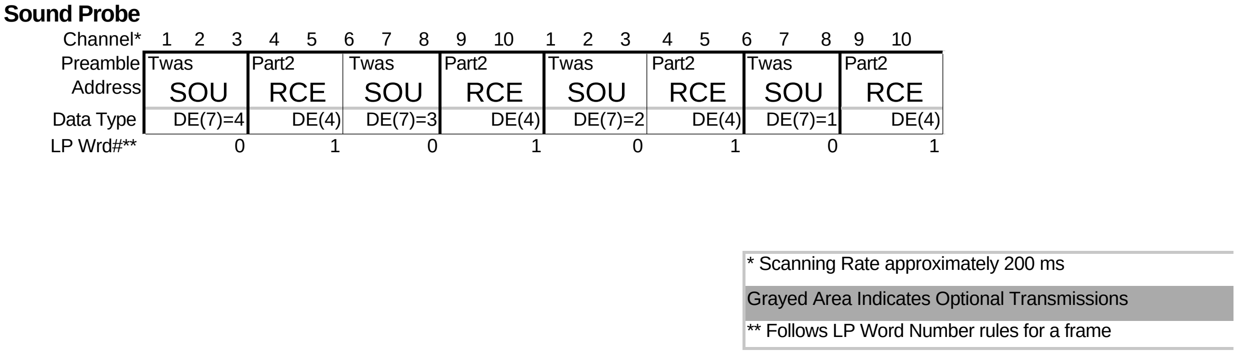
An Inlink Event frame shall not be used for the Acknowledgement frame.

A.5.8.2.6 Sounding.
The sounding cycle shall be composed of the station's address broadcast for at least the period defined as the sound duration for the radio. Data exchange values shall be as denoted in figure A-57. When the call duration is not evenly divisible by 2 triple-redundant word times, then the an additional full address may be transmitted. When an entire address is not used to complete a fractional portion of the sound duration, the caller shall begin the transmission with the second half of the target address using the PART2 preamble. In this case, the LP word number shall be 1. As shown in figure A-57, the LP word number shall toggle between 0 and 1.

When the radio is programmed to automatically derive the sound duration, the equation shall be:

Number of Channels * 0.196 + 0.784

See table A-58 for the minimum and maximum number of Trw to broadcast automatically.



Sound Probe

Channel*	1 2 3	4 5	6 7 8	9 10	1 2 3	4 5	6 7 8	9 10
Preamble	Twas	Part2	Twas	Part2	Twas	Part2	Twas	Part2
Address	SOU	RCE	SOU	RCE	SOU	RCE	SOU	RCE
Data Type	DE(7)=4	DE(4)	DE(7)=3	DE(4)	DE(7)=2	DE(4)	DE(7)=1	DE(4)
LP Wrd#**	0	1	0	1	0	1	0	1

* Scanning Rate approximately 200 ms

Grayed Area Indicates Optional Transmissions

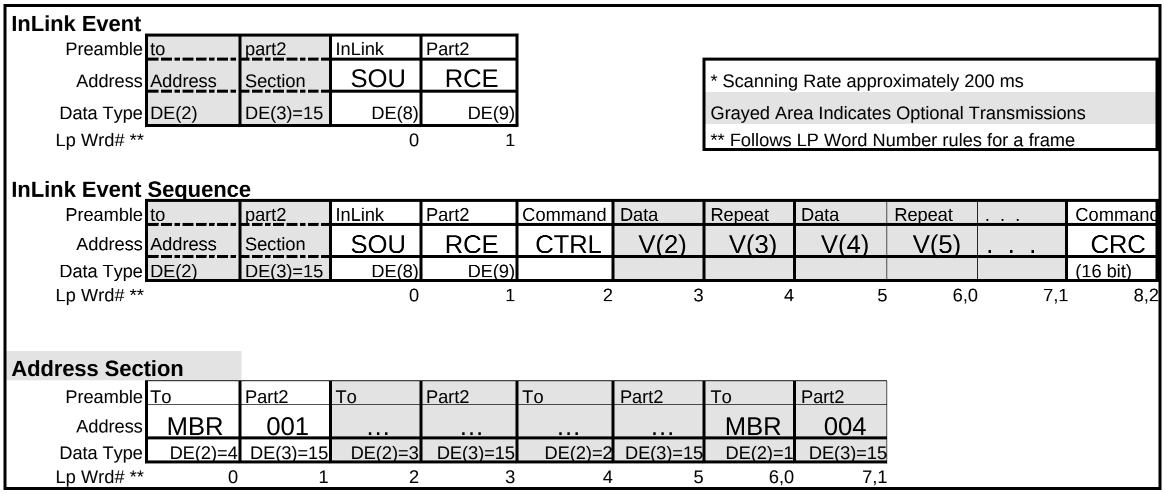
** Follows LP Word Number rules for a frame

FIGURE A-57. Example sounding frame format.

A.5.8.2.7 Inlink transactions.
AQC-ALE stations shall have the capability to transfer information within the Inlink state of the radio. A special purpose frame is defined for the purpose of separating link establishment transactions from transactions that occur during the Inlink state. Two types of Inlink transactions are defined, Inlink Event and Inlink Event Sequence. Either transaction can have an optional address section appended to the beginning of the frame. This optional address section indicates that the transaction is targeted at the addresses defined in this section of the frame.

The Inlink frame uses Data Exchange DE(8) and DE(9). DE(8) informs the recipient of the type of transaction and whether this frame needs to be acknowledged. See A.5.8.3.8. DE(9) data content indicates to the caller the exact form of the data and identifies if the sender intends to remain in the linked state with all those represented in the address section of the frame. When the address section is omitted, the frame shall be targeted to all stations currently linked with the transmitting station. See A.5.8.3.9.

The data Exchange values shall be per figure A-58. This figure outlines the general format of both types of Inlink transaction events.



InLink Event

Preamble	to	part2	InLink	Part2
Address	Address	Section	SOU	RCE
Data Type	DE(2)	DE(3)=15	DE(8)	DE(9)
Lp Wrd# **			0	1

* Scanning Rate approximately 200 ms
Grayed Area Indicates Optional Transmissions
** Follows LP Word Number rules for a frame

InLink Event Sequence

Preamble	to	part2	InLink	Part2	Command	Data	Repeat	Data	Repeat	. . .	Command
Address	Address	Section	SOU	RCE	CTRL	V(2)	V(3)	V(4)	V(5)	. . .	CRC
Data Type	DE(2)	DE(3)=15	DE(8)	DE(9)							(16 bit)
Lp Wrd# **			0	1	2	3	4	5	6,0	7,1	8,2

Address Section

Preamble	To	Part2	To	Part2	To	Part2	To	Part2
Address	MBR	001	…	…	…	…	MBR	004
Data Type	DE(2)=4	DE(3)=15	DE(2)=3	DE(3)=15	DE(2)=2	DE(3)=15	DE(2)=1	DE(3)=15
Lp Wrd# **	0	1	2	3	4	5	6,0	7,1

FIGURE A-58. Example inlink transaction TRW sequences.

A.5.8.2.7.1 Inlink transaction as an acknowledgement (NT).

The Inlink Event or the Inlink Event Sequence shall be used as the Acknowledgement frame of a handshake whenever the calling radio has a message for the radios entering the Inlink state. If the INLINK preamble is replacing a TIS preamble indicating that the radios were to remain in an Inlink state, then the I'M LINKED bit shall be set to 1. If a TWAS preamble would normally be used for this transmission, the I'M LINKED bit shall be set to 0. Thus, the calling station can minimize over the air time for any transaction by judicious use of Inlink state and associated control bits.

A.5.8.2.7.2 CRC for Inlink event sequences.

As seen in figure A-58, a command section of an Inlink event sequence shall consist of the COMMAND preamble, followed by the data associated with the command using the preambles DATA and REPEAT. The Inlink event sequence frame shall be terminated with a COMMAND preamble containing the CRC of the data contained in all words starting with the first COMMAND preamble. This CRC shall be computed exactly as the CRC for a standard ALE DTM (See A.5.6.1). The receiver shall maintain a history of failed CRC. The history may be displayed to the operator or used in channel selection algorithms for follow-on traffic.

A.5.8.2.7.3 Use of address section.

The address section of a Inlink transaction, when present, shall indicate that the addressed stations in the link are to react to the information contained in the message section.

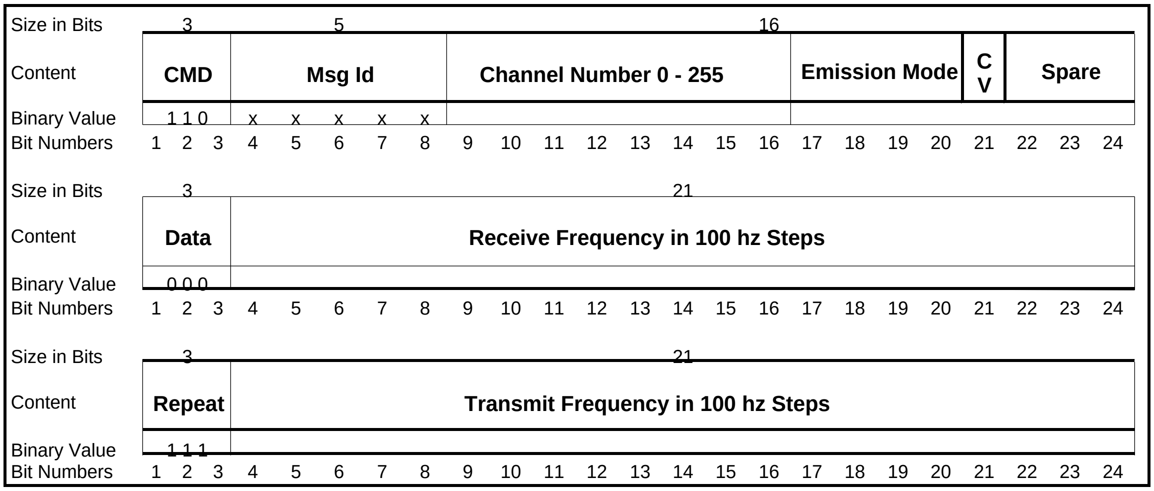
TABLE A-IL. Adding spaces during AMD unpacking.

	Message Value is in a Dictionary	Message Value is in ASCII-64 and not Alphanumeric	Message is Value is Alphanumeric
First Character of Message	No Leading Space	No Leading Space	No Leading Space
Last Expanded Character from Lookup	Add Leading Space	No Leading Space	Add Leading Space
Last Expanded Character is ASCII-64	Add Leading Space	No Leading Space	No Leading Space

A.5.8.3.2.2 Channel definition (NT).

The channel definition provides a system to reprogram the radio with a different frequency or to cause stations in a link to move to a traffic channel. This allows the radios to listen for general propagation characteristics in a common area and then move to a nearby channel to manage the inlink state transactions. By allowing a channel to be reprogrammed, the radio can adapt to a wide variety of conditions that may occur on a mission. If congestion is experienced on the assigned frequency, the stations shall return to the normal scan list and reestablish the call.

The channel index number is specified from a range of 0 to 255. A radio shall have at least 100 channels available for reprogramming. A channel index of 0 shall indicate that the receive and transmit frequencies are to be used for the remainder of this link. Other channel index numbers indicate that the new assignment shall be entered into the channel table.



Size in Bits	3	5	16			
Content	CMD	Msg Id	Channel Number 0 - 255	Emission Mode	C V	Spare
Binary Value	1 1 0	x x x x x				
Bit Numbers	1 2 3	4 5 6 7 8	9 10 11 12 13 14 15 16	17 18 19 20	21	22 23 24

Size in Bits	3	21
Content	Data	Receive Frequency in 100 hz Steps
Binary Value	0 0 0	
Bit Numbers	1 2 3	4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

Size in Bits	3	21
Content	Repeat	Transmit Frequency in 100 hz Steps
Binary Value	1 1 1	
Bit Numbers	1 2 3	4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

FIGURE A-61. Channel definition and meet-me function.

Frequencies shall be specified as a 21-bit values with each step being 100 Hz. See figure A-61 for an example format of this message. A 2-bit value 0 for emission mode shall indicate upper side band and a value of 1 shall indicate a value of lower side band. Bits 17-18 refer to the receive frequency, bits 19-20 to the transmit frequency. A value of "1" in bit 21 or the Channel Verification bit indicates that the called station will initiate an inlink transmission requesting an acknowledgement from the calling station upon going to the new channel. This bit is only valid in the event that the Channel Number was specified to be "0".

Word 2 of the message shall consists of:

1. 3 bits of LP Level number. Values range from 0 through 4.
2. 1 bit for Lower Level Linking. When set to 1, the radio shall honor lower level link attempts.
3. 3 bits for LP Key number identification. A value of 0 indicates no key assignment. When an LP level greater than 0 exists, this would be an non-operational condition. If more than one type of key is used between LP levels, they must use the same key index. When a radio does not have a key present for a given LP Key, a value of NOKEY shall be used.
4. 5 bits for the number of channels. Immediately following this word shall be (number_of_Channels/2) words containing the channel numbers to use. Earlier commands defining channel numbers or a preprogrammed value define the actual frequencies used.
5. 6 bits for defining the words from a dictionary into the 64 words. The mapping of a dictionary into a database dictionary allows a specific set of words that yield a higher frequency hit rate to the dictionary. A value of 0 indicates using the orginal programmed dictionary. The mapping of the dictionary is contained in the Trw that follow the channel association.

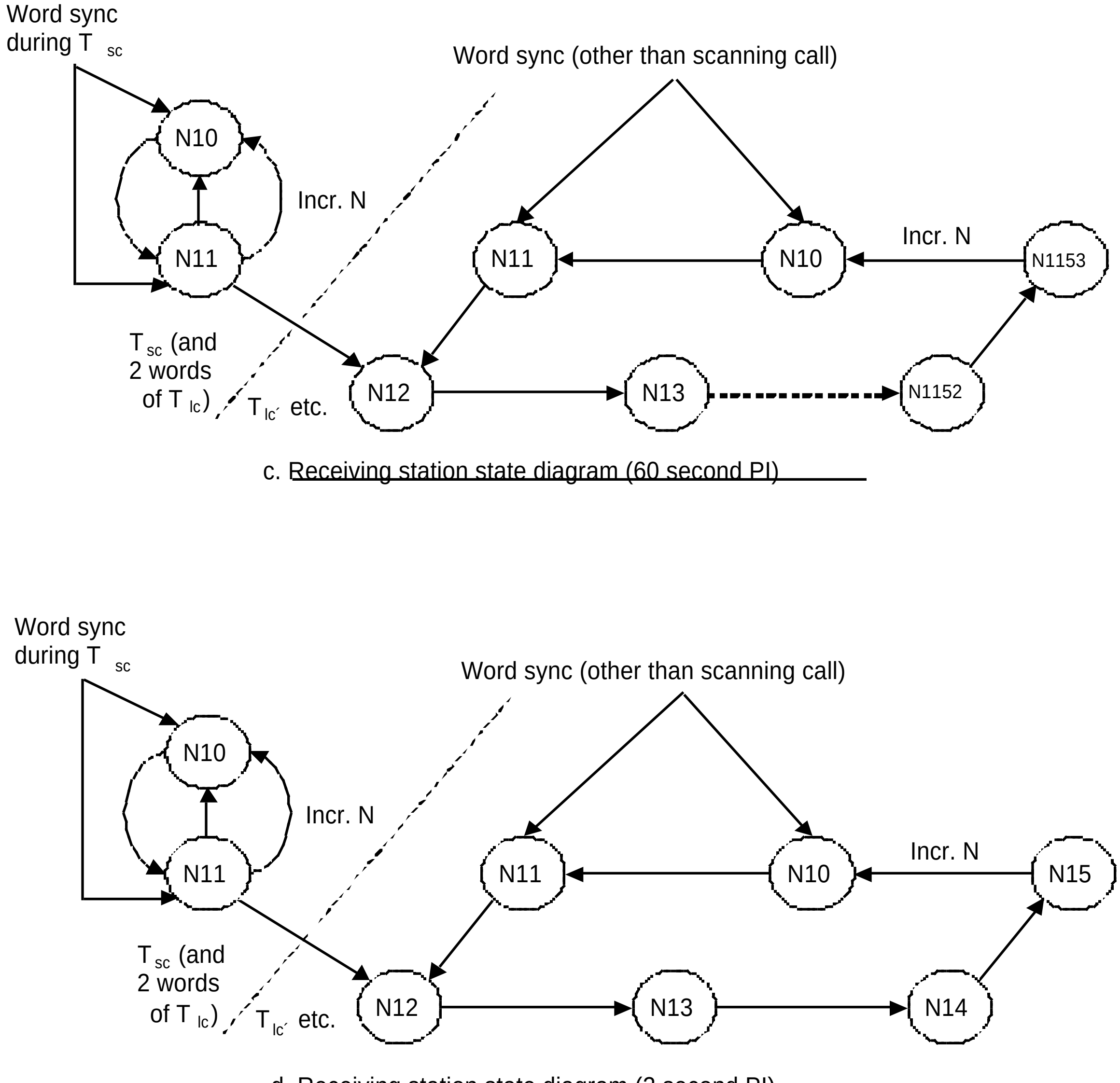
A.5.8.3.2.8 Database content listing (NT)

This command shall have the same format as the Define Database Content.

A.5.8.4 AQC-ALE linking protection.

When operating in LP with AQC-ALE, every 24-bit AQC-ALE word shall be scrambled in accordance with Appendix B. The same rules for LP in baseline 2G ALE shall be applied to AQC-ALE with the following exceptions:

- The word number for all TO AQC-ALE words during the scanning call shall be 0, and the word number for all PART 2 AQC-ALE words during the scanning call shall be 1. The TIS or TWAS word that concludes a scanning call shall use word number 2 and the following PART 2 word shall use word number 3.
- The AQC-ALE response frame shall use word numbers 0, 1, 2, and 3.
- A 2-word AQC-ALE acknowledgement shall use word numbers 0 and 1. The TOD shall be later than that used at the end of the scanning call.



c. Receiving station state diagram (60 second PI)

d. Receiving station state diagram (2 second PI)

FIGURE B-4. Transmitting and receiving stations state diagram (continued).

B.5.3.2 Receiving station.
Because of the possibility of acceptable decodes under multiple TOD/word number combinations, receivers shall attempt to decode received words under all allowed combinations (the current and adjacent PIs (future and past), and both w = 0 and w = 1) when attempting to achieve word synchronization with a calling station (six combinations). Stations prepared to accept time requests (see B.5.5.2.2) shall also attempt to decode received words using coarse TOD (fine time = all 1s, correct coarse time only) with both w = 0 and w = 1 (eight combinations total). All valid combinations shall be checked while seeking word sync. After achieving word sync, the number of valid combinations is greatly reduced by the linking protection

one slot (900 ms). The PI field in the seed shall be used as a 17-bit integer rather than as an 11-bit coarse time and a 6-bit fine time field. This 17-bit PI field shall contain the number of 900 ms slots that have elapsed since midnight (network time). The word number field in the seed shall always be 00000000. The date fields shall reflect the current network date. The frequency field shall indicate the frequency on which the protected PDU is sent. Synchronous-mode 3G ALE nodes shall ignore any synchronous-mode Probe PDU (i.e., a Probe PDU that is not preceded by Scanning Call PDUs) which is not encrypted using the current PI number.

B.5.4.3 Procedure for asynchronous-mode 3G ALE.
Asynchronous 3G handshakes shall be protected using the procedure in B.5.3 that has been modified as follows.

B.5.4.3.1 Protected 3G asynchronous-mode scanning call.
The probe PDU that concludes a 3G asynchronous-mode call shall be encrypted using word number = 2. Scanning call PDUs shall be encrypted using alternating word numbers 0 and 1. The word number used in encrypting the first scanning call PDU shall be selected so that the scanning call PDU sent immediately before the probe PDU is encrypted using word number = 1.

B.5.4.3.2 Protected 3G asynchronous-mode response.
The handshake PDU that follows an asynchronous-mode call shall be encrypted using the current TOD with word number = 3.

B.5.4.4 Protected 3G PI progression.
3G ALE nodes shall not accept PDU sequences in which the TOD used to encrypt a PDU is earlier than the TOD used to encrypt a preceding PDU of that sequence.

B.5.5 Time protocols.
The following shall be employed to synchronize LP time bases. The time service protocols for active time acquisition, both protected (B.5.5.2) and non-protected (B.5.5.3), are mandatory for all implementations of LP.

B.5.5.1 Time exchange word format.
See Appendix A, A.5.6.4.3.

B.5.5.2 Active time acquisition (protected).
A station that knows the correct date and time to within 1 minute may attempt to actively acquire time from any station with which it can communicate in protected mode by employing the protocol in the following paragraphs. The quality of time so acquired is necessarily at least one grade more uncertain than that of the selected time server. A station that does not know the correct date and time to within 1 minute may nevertheless employ this protected protocol by repeatedly guessing the time until it successfully communicates with a time server.

B.5.5.2.1 Time Request call (protected).
A station requiring fine time shall request the current value of the network time by transmitting a Time Request call, formatted as follows. (In principle, any station may be asked for the time,

TABLE OF CONTENTS

TABLE OF CONTENTS
(continued)

PARAGRAPH PAGE

TABLE OF CONTENTS (continued)

TABLES

TABLES
(continued)

SUPERSEDES PAGE 268 OF MIL-STD-188-141B.

TABLES
(continued)

FIGURES

FIGURES
(continued)

C.1 GENERAL.

C.1.1 Scope.

This appendix contains the requirements for the prescribed protocols and directions for the implementation and use of third generation (3G) high frequency (HF) radio technology including advanced automatic link establishment (ALE), automatic link maintenance, and high-performance data link protocols. The inter-relationship of the technology specified in this appendix to other HF automation standards is shown in figure C-1.

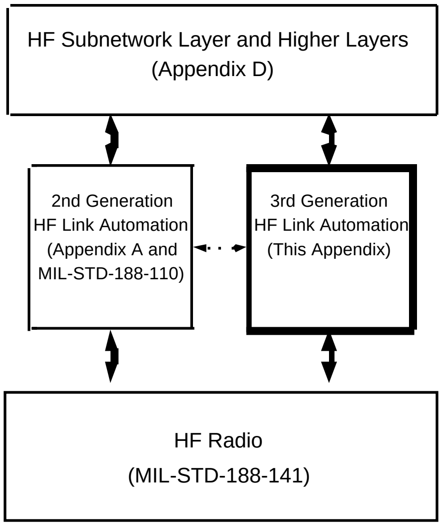


FIGURE C-1. Scope of 3G technology.

C.1.2 Applicability.

3G technology provides advanced technical capabilities for automated HF radio systems. This advanced technology improves on the performance of similar techniques described elsewhere in this standard. Thus, 3G technology may not be required by some users of HF radio systems. However, if the user has a requirement for the features and functions described herein, they shall be implemented in accordance with the technical parameters specified in this appendix.

C.2 APPLICABLE DOCUMENTS.

C.2.1 General.

The documents listed in this section are specified in C.4 and C.5 of this appendix. This section does not include documents cited in other sections of this standard or recommended for additional information or as examples. While every effort has been made to ensure the completeness of this list, document users are cautioned that they must meet all specified requirements documents cited in C.4 and C.5 of this appendix, whether or not they are listed here.

C.2.2 Government documents.

C.2.2.1 Specifications, standards, and handbooks.

The following specifications, standards, and handbooks form a part of this document to the extent specified herein. Unless otherwise specified, the issues of these documents are those listed in the issue of the Department of Defense Index of Specifications and Standards (DODISS) and supplement thereto, cited in the solicitation.

STANDARDS

FEDERAL

FED-STD-1037	Telecommunications: Glossary of Telecommunication Terms

MILITARY

MIL-STD-188-110	Interoperability and Performance Standards for HF Data Modems

Unless otherwise indicated, copies of federal and military specifications, standards, and handbooks are available from the Naval Publications and Forms Center, ATTN: NPODS, 5801 Tabor Avenue, Philadelphia, PA 19120-5099.

C.2.2.2 Other Government documents, drawings, and publications.

The following other Government documents, drawings, and publications form a part of this document to the extent specified herein. Unless otherwise specified, the issues are those cited in the solicitation.

None.

C.2.3 Non-Government publications.

The following documents form a part of this document to the extent specified herein. Unless otherwise specified, the issues of the documents which are DoD adopted are those listed in the issues of the DODISS cited in the solicitation. Unless otherwise specified, the issues of the documents not listed in the DODISS are the issues of the documents cited in the solicitation (see 6.3).

INTERNATIONAL STANDARDIZATION DOCUMENTS

North Atlantic Treaty Organization (NATO) Standardization Agreements (STANAGs)

STANAG 4285	Characteristics of 1200/2400/3600 bits per second Single Tone modulators/demodulators for HF Radio Links
STANAG 4197	Modulation and Coding Characteristics that Must be Common to Assure Interoperability of 2400 BPS Linear Predictive Encoded Digital Speech Transmitted Over HF Radio Facilities

STANAG 4198	Parameters and Coding Characteristics That Must be Common to Assure Interoperability of 2400 BPS Linear Predictive Encoded Digital Speech
STANAG 4538	Technical Standards For An Automatic Radio Control System For HF Communication Links

International Telecommunications Union (ITU)

Radio Regulations ITU-R F.520-2	Recommendtion for Fixed Service, Use of High Frequency Ionospheric Chanel Simulators

C.2.4 Order of precedence.

In the event of a conflict between the text of this document and the references cited herein, the text of this document takes precedence. Nothing in this document, however, supersedes applicable laws and regulations unless a specific exemption has been obtained.

C.3 DEFINITIONS.

C.3.1 Standard definitions and acronyms.

C.3.2 Abbreviations and acronyms.

The abbreviations and acronyms used in this document are defined below. Those listed in the current edition of FED-STD-1037 have been included for the convenience of the reader.

2G	second generation
2G ALE	second generation automatic link establishment
3G	third generation
3G ALE	third generation automatic link establishment
ACK	acknowledgment
ACQ-ALE	alternative quick call –automatic link establishment
AGC	automatic gain control
ALE	automatic link establishment
ALM	automatic link maintenance
ARQ	automatic repeat request
ASCII	American Standard Code for Information Interchange
AWGN	additive white gaussian noise
bps	bits per second
BW0	Burst Waveform 0
BW1	Burst Waveform 1
BW2	Burst Waveform 2
BW3	Burst Waveform 3
BW4	Burst Waveform 4
CLC	circuit link controller
CM	Connection Manager
CMD	ALE preamble word COMMAND
CONF	confirm
CRC	cyclic redundancy check
CSU	Call SetUp

dB	decibel
DO	design objective
EMCON	Emission Control
EOM	End of Message
FEC	forward error correction
FSK	frequency shift keying
GPS	Global Positioning System
HF	high frequency
HDL	high-rate data link protocol
HNMP	HF Network Management Protocol
Hz	Hertz
LDL	low-rate data link protocol
lsb	least-significant bit
kHz	kiloHertz
MHZ	megahertz
ms	millisecond
msb	most-significant bit
NAK	negative acknowledgment
PDU	protocol data unit
PN	pseudo noise
PU	participating unit (usually a radio station)
REQ	request
rx	receive
s	second
SDU	service data unit
SNMP	simple network management protocol
SSB	Single SideBand
TERM	Terminate
TLC	Transmit Level Control
TM	traffic management
TOD	time of day
TRF	Traffic
TSU	Traffic SetUp
TWAS	ALE preamble word THIS WAS
tx	transmit
UNL	unlink
UTC	coordinated universal time

C.3.3 Operating parameters.

The operating parameters used in this appendix are collected here for the convenience of the reader.

Symbol	Parameter Name	Default Value
T_{sym}	PSK symbol time	1/2400 s _ 417 µs
T_{slot}	Slot time	900 milliseconds (ms)
C	Number of scanned channels	
M	Number of repetitions of protocol data units (PDUs) per channel in asynchronous networks	1.3
T_{sc}	Time for an asynchronous mode scanning call	
T_{tlc}	Time for transmit level control settling	256/2400 s _ 106.7 ms
$T_{BW0\ pre}$	Time for Burst Waveform 0 preamble	384/2400 s = 160.0 ms

$T_{BW0\ data}$	Time for Burst Waveform 0 data	832/2400 s _ 346.7 ms
D	Current dwell channel	
T	Seconds since midnight (network time)	
G	Dwell group number	

Also see table C-XXVIII 3G-ALE Protocol Data for additional operating parameters.

C.4 GENERAL REQUIREMENTS.

The third-generation automatic link establishment (3G-ALE) protocol, the Traffic Management (TM) protocol, the High-Rate Data Link (HDL) and Low-Rate Data Link (LDL) protocols, and the circuit link management (CLC) protocol form a mutually-dependent protocol suite (see figure C-2). Compliance with this appendix requires compliant implementations of all of the protocols defined in this appendix (shown in shaded box in figure C-2).

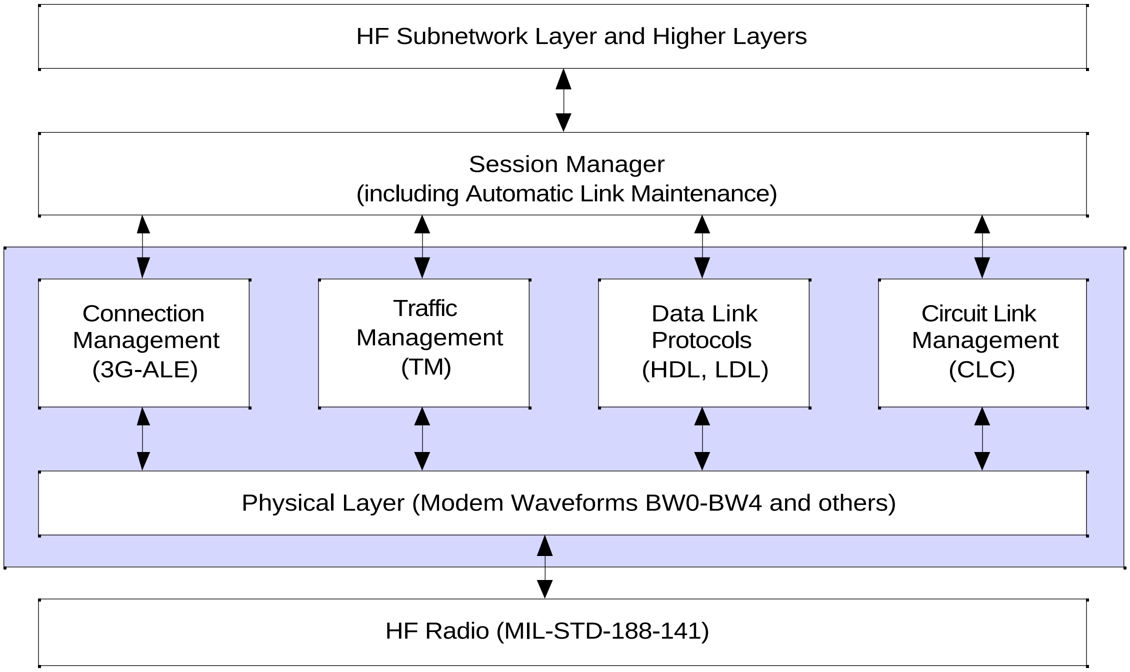

HF Subnetwork Layer and Higher Layers
Session Manager
(including Automatic Link Maintenance)
Connection Management (3G-ALE)
Traffic Management (TM)
Data Link Protocols (HDL, LDL)
Circuit Link Management (CLC)
Physical Layer (Modem Waveforms BW0-BW4 and others)
HF Radio (MIL-STD-188-141)

FIGURE C-2. 3G HF protocol suite.

C.4.1 Connection Management.

The central entity in 3G link management is a process called the Connection Manager. This process coordinates the Automatic Channel Selection (ACS), Automatic Link Establishment (ALE), and Automatic Link Maintenance (ALM) processes to establish and maintain links requested by the Session Manager process or remote stations. The state diagrams for the 3G-ALE protocols in this appendix have been simplified by moving all of the control functions not directly involved in link establishment handshakes into the CM process. This is intended to more clearly elucidate the interoperability requirements in the protocols.

NOTE: Although this function is necessary in any implementation of 3G ALE, it does not exchange PDUs over the air with other stations, and bears no direct interoperability requirements. It is described here to assist in understanding the other 3G processes.

In response to requests for links from the local Session Manager process, the CM process generally proceeds as follows:

- CM requests that the ACS process identify a channel for placing a call to the desired destination(s).
- CM computes the time at which the ALE process will need to begin sending its Call PDU on that channel. It then computes the time at which a link request must be sent to the ALE process by deducting from the call time the time needed for sending a link request to ALE, time to tune and otherwise prepare the RF equipment for placing the call, and time for Listen Before Transmit on the selected channel.
- CM sends a link request to the ALE process at the appropriate time.
- Should the ALE process fail to establish a link for any reason, the CM process repeats this entire sequence until either the link is established or a maximum number of attempts has been reached. In the latter case, the CM process reports failure to the Session Manager.

Note that this procedure leaves the ALE process in its scanning state (searching for incoming calls) at all times when it is not actively linking or linked. This is a key requirement for achieving high availability and minimizing linking delays.

C.4.2 Frequency management.

C.4.2.1 Calling and traffic channels.

Frequencies assigned for use in 3G networks will be designated for use in calling, traffic, or both. Network managers should observe the following principles in assigning frequencies to channels in these networks:

- Use of a channel for both calling and traffic reduces performance in networks subject to heavy traffic loads.
- Traffic channels should be assigned near calling channels so that the propagation characteristics of traffic channels are similar to those of the calling channels.
- Calling channels should be assigned to scan lists (see Scanning below) in non-monotonic frequency order so that the available frequency range is covered several times during a single scan. For example, frequencies 3, 4, 5, 6, 8, 10, 11, 13, 18, and 23 MHz might be scanned in the order 3, 6, 11, 23, 5, 10, 18, 4, 8, 13.

Channels are numbered for reference, starting with channel 0. Calling channels shall be assigned to the lowest-numbered channels: when C calling channels are scanned, the highest-numbered calling channel shall be C-1. Calling channels shall be scanned in ascending channel number order.

C.4.2.2 External frequency management.

Systems shall provide for management of frequency use via the network management interface (see Section C.4.9). This capability shall include at least the following:

- Assignment of frequencies to channels
- Enabling and disabling of calling and traffic on each channel
- Assignment of channels to scan list
- Entry of channel quality data

NOTE: The network manager must assign the first three items uniformly network-wide.

C.4.3 Network synchronization.

3G systems shall include mechanisms to maintain synchronization among all station time bases in a network. When 3G ALE is operating in synchronous mode, the difference between the earliest time and the latest time among the stations must not exceed 50 ms. In asynchronous networks, the permissible range of network times is determined by the current level of linking protection, if any.

C.4.3.1 External synchronization.

A means shall be provided to set the local time from a source such as a Global Positioning System (GPS) receiver. The internal time base shall differ by no more than 1 ms from the external source immediately after such a time update. Time base drift shall not exceed 1 part per million.

C.4.3.2 Over-the-air synchronization.

When an external source of synchronization is not available, 3G systems shall maintain synchronization using the synchronization management protocol of C.5.2.7.

C.4.4 Scanning.

When not engaged in any of the 2G or 3G protocols, 3G systems shall continuously scan assigned channels, listening for 2G and 3G calls. They shall leave the scanning state when called or when placing a call, in accordance with the protocol behaviors specified in C.5.2.4 and C.5.2.4.3.

C.4.4.1 Synchronous mode scanning.

3G-ALE synchronous-mode receivers shall scan at a synchronized rate of 5.4 seconds per channel. Stations shall be assigned to dwell groups by the network manager. Each dwell group shall listen on a different channel during each dwell period, in accordance with the following formula:

$$D = ((T / 5.4) + G) \bmod C$$

where D = Dwell channel number

T = Seconds since midnight (network time)

G = Dwell group number

C = Number of channels in scan list

Note that this yields channel numbers in the range 0 to C-1 in accordance with C.4.2.1.

C.4.4.2 Asynchronous mode scanning.

3G systems using asynchronous mode 3G ALE shall scan assigned calling channels at a rate of at least 1.5 channels per second. (design objective (DO): scan at 10 channels per second, in which case the corresponding dwell period of 100 ms may be extended to up to 667 ms as required when evaluating received signals. If a BW0 preamble has not been detected within 667 ms, the system shall resume scanning.)

C.4.5 3G addresses.

3G systems use 11-bit binary addresses in the over-the-air protocols.

These 3G addresses shall be translated to and from second-generation addresses (call signs of up to 15 characters) for operator use. Only the 26 capital letters (A through Z) and ten digits (0 through 9) from the American Standard Code for Information Interchange (ASCII) may be used in second-generation addresses. This ASCII subset is termed the ASCII-36 subset in this appendix.

C.4.5.1 Synchronous mode address structure.

The synchronous mode 3G-ALE protocol defines further structure within the 11-bit address space: the 5 least-significant bits (LSBs) of the address shall contain the dwell group number of the node, and the 6 most-significant bits (MSBs) shall contain the node's member number within that group (see figure C-3).

6 bits	5 bits
Member #	Group #

FIGURE C-3. Synchronous mode address structure.

C.4.5.2 Net entry addresses.

The member numbers from 111100 through 111111 (addresses 11110000000 through 11111111111) are reserved for temporary use by stations calling into a network, and shall not be assigned to any network member.

C.4.5.3 Multicast addresses.

Multicast addresses form a distinct 6-bit address space, and shall be distinguished from individual addresses by their use only in multicast calls. When computing link IDs for use in multicast calls, the multicast address shall be placed in the most-significant six bits (member number portion in figure C-1), and the group number bit positions shall be filled with five bits set to 1.

C.4.5.4 Node address assignments.

Each node in a network shall be assigned a single 11-bit address that is distinct from all other node addresses in the network. This address shall be recognized by that node in individual and unicast calls.

> NOTE: When it is desired to be able to reach all network members with a single call, and network traffic is expected to be light, up to 60 network member stations may be assigned to one dwell group. However, this arrangement is subject to calling channel congestion. To support heavier call volume than the single-group scheme will support, the network members should be distributed into multiple dwell groups.

C.4.5.5 Multicast address assignments.

A 3G system shall be programmable to subscribe to (recognize) at least 10 multicast addresses in addition to its individual node address. Multicast addresses have network-wide scope.

C.4.5.6 NATO-mode addressing.

When interoperation with NATO Automatic Radio Control System (ARCS) networks is required, the addressing scheme described above shall be modified as specified in this section. An ARCS address includes a 13-bit network number and a 10-bit station address.

- The six most significant bits of the 10-bit ARCS station address shall be used as the member number portion of the 3G address.
- The four least-significant bits of the 10-bit ARCS station address shall be used as the four most-significant bits of the group number portion of the 3G address.
- The least-significant bit of the group number portion of the 3G address shall be used as a "cross net" (XN) flag, set to 1 if the 13-bit ARCS network numbers of the calling and called addresses differ. Otherwise the XN flag shall be 0.

The 13-bit ARCS network number is not sent explicitly, but shall be used to apply linking protection to 3G-ALE PDUs sent in NATO mode (see section C.5.2.8).

C.4.6 3G ALE.

3G ALE provides functionality similar to second-generation ALE (2G ALE) as described in Appendix A, but with improved ability to link in stressed channels, to link more quickly, and to operate efficiently in large, data-oriented networks.

3G-ALE systems shall be capable of operation in both asynchronous and synchronous modes in accordance with C.5.2.4.1 and C.5.2.4.2. A system operating in synchronous mode shall recognize asynchronous-mode scanning calls addressed to it and respond to such calls in accordance with the asynchronous-mode protocol.

After a link is established using 3G ALE, the system shall wait no more than a programmable time, (T_{traf_wait} Time or trafWaitTimeMcast as appropriate), for traffic setup to begin, and shall return to scanning if neither traffic setup nor traffic has begun within that time.

C.4.6.1 System performance requirements.

Requirements for linking probability and occupancy detection are specified in the following paragraphs.

C.4.6.1.1 Linking probability.

3G-ALE systems shall meet or exceed the linking probability requirements of table C-I while operating in synchronous or asynchronous mode. The test procedure of A.4.2.3 shall be employed, with the following modifications:

- The multipath delay settings shall be 0.5 ms for the Good channel and 2.0 ms for the Poor channel.
- Units under test shall scan 3 calling channels (C = 3).
- The requested traffic type shall be packet data.
- A link will be declared successful if, in response to the first Call PDU sent, the 3G-ALE controllers complete an individual call handshake and both tune to the traffic channel specified in the handshake PDU to begin traffic setup.

TABLE C-I. Linking probability requirements versus signal to noise ratio (SNR) decibels (dB) in 3 kHz channels.

Probability of Linking Success	Gaussian	ITU-R F.250-2 Good	ITU-R F.250-2 Poor
25%	-10	-8	-6
50%	-9	-6	-3
85%	-8	-3	0
95%	-7	1	3

C.4.6.1.2 Occupancy detection.

3G-ALE systems shall detect occupied channels as specified below for synchronous or asynchronous operation, and shall not send ALE PDUs on channels that appear to be occupied without operator intervention. The probability of declaring a channel occupied when it carries only additive white gaussian noise (AWGN) shall be less than 1 percent.

C.4.6.1.2.1 Occupancy detection for synchronous operation.

3G-ALE systems operating in synchronous mode shall correctly recognize that a channel is occupied at least as reliably as indicated in table C-II during the Listen portion of Slot 0 (see C.5.2.3, Synchronous dwell structure). The test procedure of A.4.2.2 shall be used. Systems shall also meet or exceed the requirements of table C-II for detecting calling channels in use while listening before calling.

C.4.6.1.2.2 Occupancy detection for asynchronous operation.

3G-ALE systems operating in asynchronous mode shall meet the occupancy detection requirements of A.4.2.2, using the test procedure specified in A.4.2.2. Such systems shall also meet the 3G-ALE and 3G-HDL occupancy detection requirements of table C-II.

TABLE C-II. **Synchronous-mode occupancy detection requirements (3 kHz SNR dB).**

Waveform	AWGN 3 kHz SNR (dB)	Minimum Required Detection Probability
2G-ALE	0	50%
	6	90%
3G-ALE (BW0)	-9	50%
	-6	95%
3G-HDL (BW2)	0	30%
	6	70%
single sideband (SSB) Voice	6	50%
	9	75%
MIL-STD-188-110 or	0	30%
FED-STD-1052 PSK modem	6	70%
STANAG 4285 or	0	30%
STANAG 4529 PSK modem	6	70%

C.4.6.2 Automatic channel selection (ACS).

The 3G-ALE calling protocols inherently evaluate channels during link establishment. However, informed selection of the initial calling channel can reduce calling overhead (in both synchronous and asynchronous modes) and result in faster linking (in asynchronous mode).

C.4.6.2.1 Calling channel selection.

3G-ALE systems should use all available channel quality data to select the initial channel for calling:

- Calling channel link quality measurements collected from all received PDUs.
- Occupancy of traffic channels monitored during Slot 0 of each scanning dwell.
- Data from prediction programs and other external sources stored in the channel quality data in the 3G-ALE Station table (e.g., via the network management interface).

C.4.6.2.2 Traffic channel selection.

Traffic channel selection for 3G ALE requires finding a suitable channel in the ACS database for the requested type of traffic. The traffic channel may be on any traffic frequency in the database (i.e., it need not be in the same band as the search channel that was used to set up the link). Traffic channel characteristics may be estimated from recent measurements of the traffic channel or nearby search channels, predictions of traffic channel characteristics, and the most recent results of monitoring the traffic channel for use by other stations.

C.4.6.3 Interoperability with 2G systems.

A 3G-ALE system shall always listen for 2G signalling when it is listening for 3G calls. Any 2G sounds received shall be evaluated, and the results shall be stored for use in placing 2G calls.

C.4.6.4 MIL-STD-188-148A functionality.

When establishing a link while operating in MIL-STD-188-148A frequency-hopping mode, a 3G-ALE system shall spread each PDU over multiple hops in accordance with Appendix F. Linking performance when linking while hopping shall meet or exceed the requirements of Appendix F.

C.4.6.5 NATO-mode link setup (for information only).

STANAG 4538 specifies two link setup modes for use in NATO networks:

- The Robust Link Setup (RLSU) protocol is identical to 3G ALE as specified in this appendix.
- The Fast Link Setup (FLSU) protocol is intended for small, lightly loaded networks. It is not interoperable with RLSU or 3G ALE.

Both LSU modes are mandatory for implementations of STANAG 4538. This is intended to provide interoperability between U.S. 3G-ALE networks and NATO networks via the RLSU mode.

C.4.7 Data link protocol.

When a link has been established for packet data transfer, using 3G ALE or other means, the TM protocol in accordance with C.5.3 shall be used to coordinate use of the HDL and LDL protocols in accordance with C.5.4 and C.5.5 to transfer data messages. When a link has been established for data virtual circuit operation, the CLC protocol (C.5.6) shall be used.

C.4.8 Automatic link maintenance.

After a link is established, the traffic channel may be found unsuitable. The Relink and Restart commands of C.5.3 shall be implemented for use when a change in frequency or data link operating mode performance.is necessary to address such conditions.

C.4.9 Network management interface.

3G systems should provide a network management interface in accordance with Appendix D to facilitate interoperability with common network management systems.

C.4.10 Order of transmission.

Unless otherwise specified, all PDUs shall be serialized as follows:

- Fields within a PDU shall be sent in left-to-right order as shown in figures in this appendix.
- Bits within fields shall be sent most-significant bit (MSB) first.

NOTE: The MSB of each field is shown as the leftmost bit in each figure in this appendix.

C.4.11 3G-ALE data structures.

3G systems shall implement the following data structures at the network management interface (if provided).

C.4.11.1 Station self address.

The "self address" of each 3G-ALE station table shall be an index into the station table (see below).

C.4.11.2 Station table.

The 3G-ALE station table shall be capable of storing at least 128 entries. Each entry shall contain at least the following fields:

- Station call sign in accordance with 2G format (up to 15 ASCII-36 characters)
- 3G address (11 bits, including dwell group number)
- Multicast subscription flag (indicates whether the associated address of this entry is a multicast to which this station listens)
- Channels on which address is valid
- Link quality measurements from that station (if not a self address or multicast address) on each calling and traffic channel including time of measurement
- Current station status (if not a self address or multicast address).

Entries for all network members shall be locked in the table. Other table entries shall store data obtained from received PDUs, with the oldest such entry replaced when new data is available and the table is full.

C.4.11.3 Channel table.

The channel table shall provide storage for at least 128 channel entries. Individual flags for each channel shall indicate whether that channel may be used for 3G link establishment, for 2G link establishment, and for traffic. Each entry shall also include transmit and receive frequencies, antenna selection and settings, power limits, and modulation type.

C.4.12 Cyclic Redundancy Check (CRC) computation procedure.

A CRC (Cyclic Redundancy Check) is a sequence of bits computed in a specific manner from a sequence of input bits. The CRC is concatenated with the string of input bits and the entire sequence is transmitted over a channel. At the receive side of the channel, the CRC is used to attempt to determine whether the channel caused any errors in the concatenated sequence. The input sequence of bits is said to be covered by the CRC. A suitably chosen method for generating the CRC sequence can reduce the probability of undetected random channel errors to approximately $(1/2)^K$ where K is the number of bits composing the CRC.

All of the CRCs used in the protocols defined in this Appendix shall be computed using the procedure defined below.

When a CRC is to be computed from the non-CRC bits of a given PDU, the following must be known:

U_0, U_1, … U_{N-1} the N non-CRC bits contained in the PDU, in the order in which they will be coded, modulated, and transmitted. U_0 is the first bit input to the PDU coding and modulation processing.

g(X) A K^{th} order polynomial with binary coefficients of form:

$$1 + g_1*X + g_2*X^2 + \ldots + g_{K-2}*X^{K-2} + g_{K-1}*X^{K-1} + X^K$$

NOTE: 1. The order K of this polynomial indicates the number of bits composing the CRC.

NOTE: 2. The $zero^{th}$ and K^{th} coefficients, g_0 and g_k, are equal to one.

The following diagram indicates the operations necessary to compute the CRC. The addition operation pictured is binary addition (exclusive-or). The multipliers pictured represent binary multiplication (or binary and); specifically, each circle containing the name of one of the polynomial coefficients multiplies its input by the coefficient value (0 or 1) to produce its output.

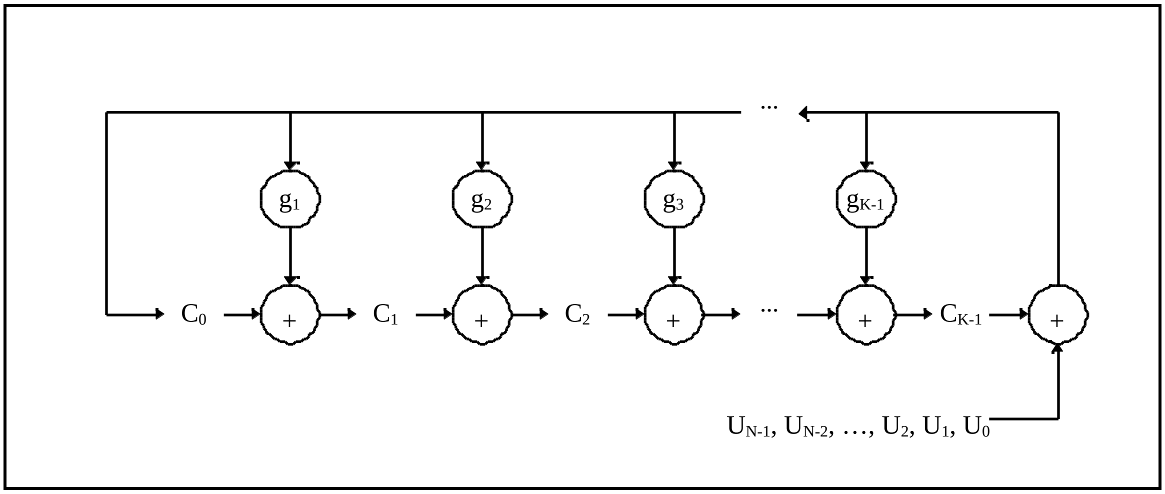


FIGURE C-4. CRC computation structure.

The above structure is used by the transmitter to produce a CRC sequence from the N user bits U_0 through U_{N-1} via the following procedure:

1. Initialize binary memory elements C_0 through C_{K-1} with 0.
2. Apply each of the N user bits in order, starting with U_0, to the binary adder at the far right of the diagram, and perform the other indicated binary additions and multiplications.
3. After each of the N user bits has been applied to the indicated adder, the memory elements C_0 through C_{K-1} contain the CRC.

4. The K bit CRC is read out and appended to bits U_0 through U_{N-1} of the PDU in right-to-left order, starting with C_{K-1} and finishing with C_0, so that the entire PDU with CRC is the bit-sequence (U_0, …, U_{N-1}, C_{K-1}, …, C_0), with U_0 being the first bit and C_0 the last.

NOTE: The structure can be viewed as a feedback shift register with feedback connections corresponding to the coefficients of the polynomial: the feedback connection labeled 'g_i' is present if and only if the coefficient g_i is equal to one.

C.5 DETAILED REQUIREMENTS.

C.5.1 Constituent waveforms.

This section defines the constituent waveforms used by the Third Generation HF Automation protocols. Burst waveforms are defined for the various kinds of signalling required in the system, so as to meet their distinctive requirements as to payload, duration, time synchronization, and acquisition and demodulation performance in the presence of noise, fading, and multipath. All of the burst waveforms use the basic 8-ary PSK serial tone modulation of an 1800 hertz (Hz) carrier at 2400 symbols per second that is also used in the MIL-STD-188-110 serial tone modem waveform.

Table C-III summarizes the characteristics of the waveforms and their uses within this standard. Note that all transmissions begin with a guard sequence (part of the preamble for BW3) to allow settling time for transmit level control (TLC) and receiver automatic gain control (AGC). These symbols are included in the indicated burst durations.

TABLE C-III. Burst waveform characteristics.

Wave form	used for	burst duration	payload	preamble	FEC coding	inter-leaving	data format	effectrive code rate[1]
BW0	3G-ALE PDUs	613.33 ms 1472 PSK symbols	26 bits	160.00 ms 384 PSK symbols	rate = 1/2, k = 7 convolutional (no flush bits)	4x13 block	16-ary orthogonal Walsh function	1 / 96
BW1	Traffic Manage-ment PDUs; HDLacknow-ledgement PDUs	1.30667 seconds 3136 PSK symbols	48 bits	240.00 ms 576 PSK symbols	rate = 1/3, k = 9 convolutional (no flush bits)	16x9 block	16-ary orthogonal Walsh function	1 / 144
BW2	HDLtraffic data PDUs	640 + (n*400) ms 1536 + (n*960) PSK symbols, n = 3, 6, 12, or 24	n*1881 bits	26.67 ms 64 PSK symbols (for equalizer training)	rate = 1/4, k = 8 convolutional (7 flush bits)	none	32 unknown/ 16 known	variable: 1 / 1 to 1 / 4
BW3	LDL traffic data PDUs	373.33 + (n*13.33) ms 32n + 896 PSK symbols, n = 32*m, m = 1, 2, ... , 16	8n+25 bits	266.67 ms 640 PSK symbols	rate = 1/2, k = 7 convolutional (7 flush bits)[2]	convolutional block	16-ary orthogonal Walsh function	variable: 1 / 12 to 1 / 24
BW4	LDL acknow-ledgement PDUs	640.00 ms 1536 PSK symbols	2 bits	none	none	none	4-ary orthogonal Walsh function	1 / 1920

Notes:

1. Reflects Forward Error Correction (FEC) and Walsh-function coding only; does not include known data or convolutional encoder flush bits.

2. In this case, the number of flush bits exceeds by one the minimum number required to flush the convolutional encoder; this makes the number of coded bits a multiple of four as is required for the Walsh-function modulation format.

Other waveforms, including the MIL-STD-188-110 serial tone modem waveform and high data rate waveform, can be used to deliver data and digitized voice signalling on circuit links established using the 3G-ALE and TM protocols.

C.5.1.1 Service primitives.

Table C-IV defines the service primitives exchanged between the Burst Waveform (physical layer) entities and the higher-layer user processes that use Burst Waveform services. Note that there is no requirement that implementations of the waveforms and protocols defined in this Appendix contain precisely these service primitives; nor are the services primitives defined below necessarily all of the service primitives that would be required in an implementation of these waveforms and protocols.

TABLE C-IV. Burst Waveform (BWn) service primitives.

Primitive name	Attribute	Values	Description
BW0_Send	Overview	Invoked by a user process to send a 26-bit data payload using the BW0 robust burst signalling format	
	Parameters	payload	an ordered sequence of 26 bits of data to be modulated and transmitted using the BW0 signalling format
	Originator	Connection Management (CM)	
	Preconditions		
BW0_Receive	Overview	Issued by BW0 Receiver when it has received a BW0 transmission.	
	Parameters	payload	the 26 bits of payload data received in the incoming BW0 transmission. The payload value can contain undetected errors due to channel noise, fading, multipath, etc.; however, on a perfect channel, the payload value would be identical to the payload parameter-value of the original BW0_Send primitive at the remote station.
	Originator	BW0 Receiver	
	Preconditions	BW0 Receiver is active.	
BW0_Pre_Detect	Overview	Issued by BW0 Receiver when it has detected a BW0 acquisition preamble.	
	Parameters	none	
	Originator	BW0 Receiver	
	Preconditions	BW0 Receiver is active.	
BW1_Send	Overview	Invoked by a user process to send a 48-bit data payload using the BW1 robust burst signalling format	
	Parameters	payload	an ordered sequence of 48 bits of data to be modulated and transmitted using the BW1 signalling format
	Originator	· HDL protocol · TM	
	Preconditions		
BW1_Receive	Overview	Issued by BW1 Receiver when it has received a BW1 transmission.	
	Parameters	payload	the 48 bits of payload data received in the incoming BW1 transmission. The payload value can contain undetected errors due to channel noise, fading, multipath, etc.; however, on a perfect channel, the payload value would be identical to the payload parameter-value of the original BW1_Send primitive at the remote station.
	Originator	BW1 Receiver	
	Preconditions	BW1 Receiver is active.	
BW1_Pre_Detect	Overview	Issued by BW1 Receiver when it has detected a BW1acquisition preamble.	
	Parameters	none	
	Originator	BW1 Receiver	
	Preconditions	BW1 Receiver is active.	

TABLE C-IV. Burst Waveform (BWn) service primitives (continued).

Primitive name	Attribute	Values	Description
BW2_Send	Overview	Invoked by a user process to send a sequence of data packets to a remote station, using the BW2 high-rate burst signalling format	
	Parameters	tx frame	a *BW2 tx frame*, consisting of an ordered sequence of NumPkts data packets to be modulated and transmitted using the BW2 signalling format, where NumPkts = 3, 6, 12, or 24. Each data packet contains an ordered sequence of 1881 bits of payload data.
		reset	boolean value; reset = TRUE indicates to the BW2 transmitter that it should reset its Forward Transmission counter, *FTcount*.
	Originator	HDL protocol	
	Preconditions	A previous signalling exchange has established the time at which transmission of the current BW2 burst is to start, and the time at which the receiver should expect it to arrive.	
		If reset = FALSE, the value of NumPkts for the current invocation of BW2_ Send (i.e., the number of data packets in the forward transmission frame, payload) must be equal to the value of NumPkts in the preceding invocation of BW2_Send. (reset = TRUE in the first invocation of BW2_Send for a new datagram, at which time the value of NumPkts for all invocations of BW2_Send throughout the duration of the datagram transfer is determined.)	
BW2_Receive	Overview	Issued by the BW2 Receiver when it has received a BW2 transmission.	
	Parameters	rx frame	a *BW2 rx frame* containing NumRcvd indexed data packets, where 0 ≤ NumRcvd ≤ NumPkts. An indexed data packet contains · payload: a data packet containing 1881 bits of payload data; identical to the corresponding data packet in the transmitted *tx* frame. · index: the position at which the indexed data packet occurred in the forward transmission (BW2 tx frame) in which it was received, where 0 ≤ index ≤ NumPkts. index = 0 indicates that the packet was in the first packet-slot in the forward transmission. Only data packets received with no errors (as indicated by checking the 32-bit CRC added to each packet by BW2) are passed to the user process in a BW2 rx frame.
	Originator	BW2 Receiver	
	Preconditions	BW2 Receiver is active.	
		The arrival time of the incoming BW2 burst has been estimated, based on the observed arrival time of previous received signalling from the remote station.	

TABLE C-IV. Burst Waveform (BWn) service primitives (continued).

Primitive name	Attribute	Values	Description
BW3_Send	Overview	Invoked by a user process to send a data packet to a remote station, using the BW3 low-rate burst signalling format.	
	Parameters	payload	an ordered sequence of data bits to be modulated and transmitted using the BW3 signalling format. (Note: the payload lengths are chosen so as to accommodate the various possible forward transmission lengths of the LDL; see section C.5.5.)
		reset	boolean value; reset = TRUE indicates to the BW3 modulator that it should reset its Forward Transmission counter, FTcount.
	Originator	LDL protocol	
	Preconditions	A previous signalling exchange has established the time at which transmission of the current BW3 burst is to start, and the time at which the receiver should expect it to arrive.	
BW3_Receive	Overview	Issued by the BW3 Receiver when it has successfully received a BW3 transmission without errors.	
	Parameters	payload	BW3 data demodulated and received without errors by the Burst Waveform Modem, as determined by the CRC check performed by the BW3 receiver; identical to the payload parameter-value of the original BW3_Send primitive at the remote station.
	Originator	BW3 Receiver	
	Preconditions	BW3 Receiver is active.	
		The arrival time of the incoming BW3 burst has been estimated, based on the observed arrival time of previous received signalling from the remote station.	
BW4_Send	Overview	Invoked by a user process to send two bits of payload data using the BW4 robust burst signalling format.	
	Parameters	payload	two bits of data to be modulated and transmitted using the BW4 signalling format.
	Originator	LDL protocol	
	Preconditions	A previous signalling exchange has established the time at which transmission of the current BW4 burst is to start, and the time at which the receiver should expect it to arrive.	
BW4_Receive	Overview	Issued by the BW4 Receiver when it has received a BW4 transmission.	
	Parameters	payload	two bits of BW4 data received and demodulated by the Burst Waveform Modem. The payload value can contain undetected errors due to channel noise, fading, multipath, etc.; however, on a perfect channel, the payload value would be identical to the payload parameter-value of the original BW4_Send primitive at the remote station.
	Originator	BW4 Receiver	
	Preconditions	BW4 Receiver is active.	
		The arrival time of the incoming BW4 burst has been estimated, based on the observed arrival time of previous received signalling from the remote station.	

C.5.1.2 Burst waveform interleaving.

A block interleaver is used to improve FEC performance for certain of the burst waveforms described below. This interleaver is based on a single interleave matrix having R rows and C columns, and hence accommodating up to (R * C) bits.

The particular interleaver used in each burst waveform is defined by the values assigned to the following set of interleaver parameters:

R	Number of rows
C	Number of columns
i_{rs}	Row increment, stuff
i_{cs}	Column increment, stuff
Δi_{rs}	Delta row increment, stuff. Applied only when stuff count is an integer multiple of the number of columns.
Δi_{cs}	Delta column increment, stuff. Applied only when stuff count is an integer multiple of the number of rows.
i_{rf}	Row increment, fetch
i_{cf}	Column increment, fetch
Δi_{rf}	Delta row increment, fetch. Applied only when fetch count is an integer multiple of the number of columns.
Δi_{cf}	Delta column increment, fetch. Applied only when fetch count is an integer multiple of the number of rows.

The parameter-values for each burst waveform are given in the sections of this document describing the individual burst waveforms. Irrespective of the particular values assigned to these parameters, each of the interleavers is operated in the following way. Starting with the matrix empty, (R * C) input bits are stuffed one by one into the matrix using the algorithm:

```
initialize s (stuff count), $r_s$, and $c_s$ to zero
while s < (R * C)
    matrix[$r_s$,$c_s$] = input bit
    increment s
    if (s mod R) == 0
        $c_s$ = ($c_s$ + $i_{cs}$ + $\Delta i_{cs}$) mod C
    else
        $c_s$ = ($c_s$ + $i_{cs}$) mod C
    end if
    if (s mod C) == 0
        $r_s$ = ($r_s$ + $i_{rs}$ + $\Delta i_{rs}$) mod R
    else
        $r_s$ = ($r_s$ + $i_{rs}$) mod R
    end if
end while
```

NOTE: using ‘=’ to denote assignment, and ‘ ==’ to denote the equality predicate.

Once the matrix has been filled, the (R * C) output bits are fetched one by one from the matrix in interleaved order, using the algorithm:

The description of the BW0 waveform generation will proceed as follows:

- C.5.1.3.1 and C.5.1.3.2 discuss generation of raw tribit values for the first two waveform components: Gain control loop compensation and Preamble.

  NOTE: A tribit number can take on the values 0,1,…,7.

- C.5.1.3.3, C.5.1.3.4, and C.5.1.3.5 discuss the mapping of input bits to the raw tribit values for the data waveform component via FEC, interleaving, and orthogonal Walsh symbol formation.
- C.5.1.3.6 discusses generation of tribit values for the pseudo noise (PN) spreading sequence and the combining of raw tribit values and PN spreading sequence tribit values.
- C.5.1.3.7 discusses carrier modulation using combined tribit values.

C.5.1.3.1 BW0 TLC/AGC guard sequence.

The TLC/AGC guard sequence portion of the BW0 waveform provides an opportunity for both the transmitting radio’s Transmit Level Control process (TLC) and the receiving radio’s Automatic Gain Control process (AGC) to reach steady states before the BW0 preamble appears at their respective inputs, minimizing the distortion to which the preamble can be subjected by these processes. The TLC/AGC guard sequence is a sequence of 256 pseudo-random tribit symbols having the values shown in table C-V. The tribit symbols are transmitted in the order shown in table C-V, starting at the top left and moving from left to right across each row, and from top to bottom through successive rows.

TABLE C-V. TLC/AGC guard sequence symbol values.

2,6,1,6,	1,6,3,0,	6,0,1,1,	5,0,0,6,	2,6,2,1,	6,2,3,2,	7,6,4,3,	0,2,3,5,
2,7,5,1,	5,1,7,6,	1,7,1,5,	4,4,0,7,	2,2,6,2,	2,2,6,3,	3,3,7,7,	3,2,4,5,
0,7,4,7,	7,7,2,3,	1,6,7,6,	5,7,0,5,	1,0,7,6,	2,4,0,2,	7,5,5,4,	1,5,1,5,
6,7,3,0,	2,7,6,6,	4,0,7,4,	3,2,2,6,	6,7,4,7,	2,0,2,7,	2,1,5,4,	6,2,3,2,
1,6,0,7,	1,1,2,6,	2,2,0,2,	2,3,6,7,	1,7,1,7,	1,5,7,7,	2,2,2,0,	4,3,4,2,
0,6,7,6,	0,5,0,7,	1,7,4,1,	2,3,4,6,	7,2,2,0,	6,4,4,6,	6,4,2,2,	6,5,3,4,
2,3,5,7,	7,1,0,0,	0,3,1,2,	0,1,6,2,	7,4,4,3,	2,5,4,5,	6,4,2,5,	6,2,2,4,
7,0,6,2,	3,7,2,5,	4,2,4,1,	5,5,3,6,	1,1,3,2,	7,5,7,0,	7,3,5,0,	0,1,2,0

The TLC/AGC guard sequence symbols are modulated directly as described in C.5.1.3.7, without undergoing the PN spreading described in C.5.1.3.6.

C.5.1.3.2 BW0 Acquisition preamble.

The BW0 acquisition preamble provides an opportunity for the receiver to detect the presence of the waveform and to estimate various parameters for use in data demodulation. The preamble component of BW0 is the sequence of 384 tribit symbols shown in C-VI. The preamble symbols are modulated directly as described in C.5.1.3.7, without undergoing the PN spreading described in C.5.1.3.6. The preamble symbols are transmitted in the order shown in table C-VI, starting at the top left and moving from left to right across each row, and from top to bottom through successive rows.

When it detects a BW0 acquisition preamble, the BW0 receiver issues a BW0_Pre_Detect service primitive, as described in table C-IV.

TABLE C-VI. BW0 acquisition preamble symbol values.

7,7,7,7,	5,4,3,1,	1,2,0,2,	7,2,2,0,	1,3,4,7,	5,3,7,7,	4,3,1,0,	1,1,5,2,
1,6,0,0,	4,7,6,2,	2,3,6,0,	5,1,7,6,	1,6,1,7,	6,6,6,1,	7,3,0,4,	7,1,2,2,
3,3,6,7,	7,1,7,3,	1,5,0,3,	3,4,5,2,	5,2,5,3,	1,7,2,1,	5,7,6,1,	2,5,3,5,
3,6,2,0,	7,5,6,6,	0,1,4,2,	5,4,1,1,	7,0,0,6,	6,7,5,6,	3,7,4,0,	2,6,3,6,
4,5,1,0,	0,4,5,5,	4,7,1,5,	1,5,6,7,	3,3,5,2,	2,2,7,2,	3,3,0,4,	1,4,1,3,
6,0,7,2,	6,1,5,0,	1,4,1,1,	7,0,7,4,	0,2,4,5,	3,0,0,3,	1,2,6,4,	6,5,2,6,
0,0,7,3,	5,3,4,0,	6,2,7,2,	3,3,7,6,	7,1,0,0,	6,7,3,1,	5,5,0,2,	3,4,2,7,
7,4,5,2,	1,6,1,0,	4,7,1,6,	1,2,4,0,	3,6,5,4,	5,4,4,6,	1,2,5,1,	3,6,2,7,
2,6,7,4,	7,3,0,1,	5,0,5,3,	4,5,0,7,	3,2,7,0,	3,2,7,0,	6,1,6,7,	7,1,4,2,
6,7,7,4,	2,7,2,7,	3,7,6,3,	2,6,5,6,	6,3,6,6,	4,1,0,6,	2,6,4,1,	5,5,4,3,
3,4,6,3,	5,2,4,1,	1,7,5,3,	7,1,6,5,	4,6,6,2,	3,4,2,3,	3,7,4,1,	4,4,5,4,
6,1,3,4,	6,1,7,4,	1,3,5,2,	6,5,5,4,	2,1,5,1,	6,1,2,7,	1,4,4,2,	3,4,7,3

C.5.1.3.3 BW0 Forward error correction.

BW0 carries a payload of 26 protocol bits. The 26 protocol bits are encoded using the r = 1/2, k = 7 convolutional encoder shown in Figure C-6, creating 52 coded bits. A ‘tail-biting’ convolutional encoding approach is used as follows:

1. Initialize the six memory cells x^1 .. x^6 of the encoder with the last six bits of the payload sequence, p_{20} .. p_{25}, so that cell x^1 contains p_{20} and cell x^6 contains p_{25}.

2. Shift the first bit of the payload sequence, p_0, into cell x^6.

3. Extract the two coded output bits b_0 and b_1, in that order, as shown in figure C-6.

4. Shift the next payload bit into cell x^6, then extract the two coded output bits b_0 and b_1.

5. Repeat step 4 until a total of 52 coded bits have been produced.

No flush bits are necessary for the encoding process.

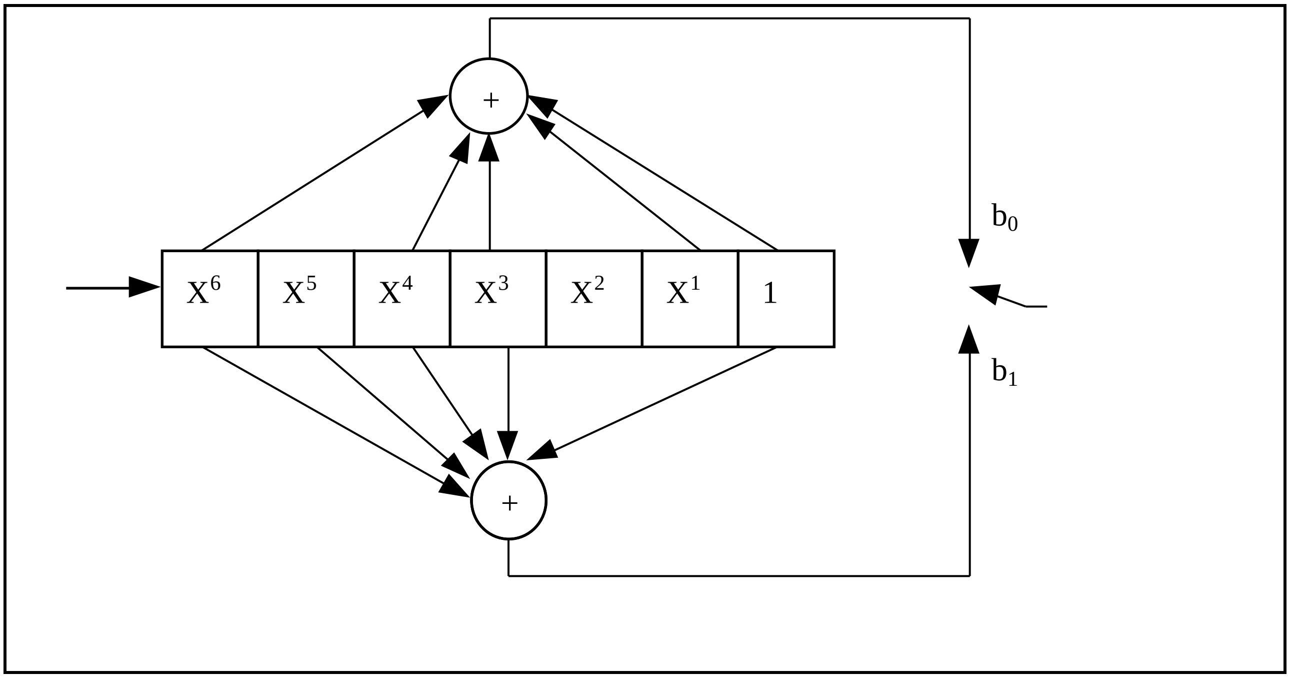


FIGURE C-6. Rate 1/2, constraint length 7 convolutional encoder.

The polynomials used are:

- $b_0 = x^6+x^4+x^3+x^1+1$
- $b_1 = x^6+x^5+x^4+x^3+1$

where x^6 corresponds to the most recent encoder input bit.

C.5.1.3.4 BW0 Interleaving.

BW0 utilizes a simple block interleaver structure which can be viewed as a 4 by 13 element rectangular array. See C.5.1.2 for a description of the interleaving process. The interleaver parameters for BW0 are as follows:

TABLE C-VII. BW0 interleaver paramenters.

R	4
C	13
i_{rs}	0
i_{cs}	1
Δi_{rs}	1
Δi_{cs}	0
i_{rf}	1
i_{cf}	0
Δi_{rf}	0
Δi_{cf}	1

C.5.1.3.5 BW0 Orthogonal symbol formation.

The interleaver fetch process removes 4 coded bits at a time from the interleaver matrix. These four coded bits are mapped into a 16-tribit sequence using the mapping given in table C-VIII. Note that each of the four-bit sequences in the Coded Bits column of the table is of the form $b_3b_2b_1b_0$, where b_3 is the first bit fetched from the interleaver matrix. The 16-tribit sequence thus obtained is repeated 4 times to obtain a 64-tribit sequence. The tribit values are placed in the output tribit sequence in the order in which they appear in the corresponding row of table C-VIII, moving from left to right across the row.

TABLE C-VIII. Walsh modulation of coded bits to tribit sequences.

Coded Bits (shown as $\mathbf{b_3b_2b_1b_0}$)	Tribit Sequence
0000	0000 0000 0000 0000
0001	0404 0404 0404 0404
0010	0044 0044 0044 0044
0011	0440 0440 0440 0440
0100	0000 4444 0000 4444
0101	0404 4040 0404 4040
0110	0044 4400 0044 4400
0111	0440 4004 0440 4004
1000	0000 0000 4444 4444
1001	0404 0404 4040 4040
1010	0044 0044 4400 4400
1011	0440 0440 4004 4004
1100	0000 4444 4444 0000
1101	0404 4040 4040 0404
1110	0044 4400 4400 0044
1111	0440 4004 4004 0440

This process repeats for a total of 13 iterations (one for each group of four coded bits) to produce 832 raw tribit values.

C.5.1.3.6 BW0 Psuedo noise (PN) spreading sequence generation and application.

A sequence of 832 pseudo-random tribit values s_i is generated by extracting 64-tribit sequences from table C-IX, 832/64 = 13 times. The tribit values are extracted in the order shown in table C-IX, starting at the top left and moving from left to right across each row, and from top to bottom through successive rows. The table contains 256 values; therefore, the PN spreading sequence is repeated every 4 blocks of 64 tribit sequences.

TABLE C-IX. BW0 PN spreading sequence.

0,2,4,3,	3,6,4,5,	7,6,7,0,	5,5,4,3,	5,4,3,7,	0,7,6,2,	6,2,4,6,	7,2,4,7,
5,5,7,0,	7,3,3,3,	7,3,3,1,	4,2,3,7,	0,2,7,7,	3,5,1,0,	1,4,0,5,	0,0,0,0,
6,5,0,1,	2,7,6,5,	5,2,7,3,	3,3,2,1,	2,5,6,1,	3,4,2,1,	0,1,2,3,	6,4,7,5,
2,2,6,2,	7,6,5,2,	4,6,5,4,	7,2,5,1,	0,0,7,7,	3,5,4,2,	1,4,2,7,	0,3,4,0,
0,0,7,7,	3,5,4,2,	1,4,2,7,	0,3,4,0,	1,0,5,2,	6,0,3,5,	1,0,5,1,	5,2,5,6,
3,2,3,7,	1,2,2,0,	7,1,3,6,	4,2,6,2,	7,4,3,7,	6,7,2,3,	1,7,4,1,	5,1,5,4,
7,1,1,2,	3,6,7,7,	6,6,1,2,	2,4,1,7,	7,5,5,4,	7,7,5,0,	7,3,7,5,	7,7,5,0,
6,6,6,1,	3,4,4,4,	0,3,3,2,	1,4,5,4,	5,3,1,1,	1,2,5,1,	7,1,5,7,	2,0,0,6

The 832 tribit values s_i of the PN sequence are then combined with the 832 raw tribit values r_i produced by the orthogonal symbol formation process described in the previous section. Each symbol of the PN sequence s_i is combined with the corresponding symbol r_i of the raw tribit sequence to form a channel symbol c_i, by adding s_i to r_i modulo 8. For instance, if $s_i = 7$, $r_i = 4$, then $c_i = 7 \quad 4 = 3$, where the symbol represents modulo-8 addition.

The process can be summarized:

$$\begin{array}{c} c_0 \\ \vdots \\ c_{2303} \end{array} = \begin{array}{c} r_0 \\ \vdots \\ r_{2303} \end{array} \quad \begin{array}{c} s_0 \\ \vdots \\ s_{2303} \end{array}$$

where r is the vector of data raw tribit values, s is the vector of PN sequence tribit values, c is the resulting vector of combined tribit values, and the symbol represents component-wise modulo-8 addition.

C.5.1.3.7 BW0 Modulation.

The sequence of channel symbols consisting of

- the TLC/AGC guard sequence of 256 tribit symbols described by C.5.1.3.1 (on which no PN-spreading has been performed), followed by

- C.5.1.4.3, C.5.1.4.4, and C.5.1.4.5 discuss the mapping of input bits to the raw tribit values for the data waveform component via FEC, interleaving, and orthogonal Walsh symbol formation.
- C.5.1.4.6 discusses generation of tribit values for the PN spreading sequence and the combining of raw tribit values and PN spreading sequence tribit values.
- C.5.1.4.7 discusses carrier modulation using combined tribit values.

C.5.1.4.1 BW1 TLC/AGC guard sequence.

The TLC/AGC guard sequence portion of the BW1 waveform provides an opportunity for both the transmitting radio's Transmit Level Control process (TLC) and the receiving radio's Automatic Gain Control process (AGC) to reach steady states before the BW1 preamble appears at their respective inputs, minimizing the distortion to which the preamble can be subjected by these processes. The TLC/AGC guard sequence is a sequence of 256 pseudo-random tribit symbols having the values shown in table C-X. The tribit symbols are transmitted in the order shown in table C-X, starting at the top left and moving from left to right across each row, and from top to bottom through successive rows. For convenience of implementation, the length of the TLC/AGC guard sequence (256 PSK symbols) has been chosen so as to be an integral multiple of the length (64 PSK symbols) of the Walsh-function modulated orthogonal symbols described in C.5.1.4.5.

TABLE C-X. TLC/AGC guard sequence symbol values.

```
2,6,1,6, 1,6,3,0, 6,0,1,1, 5,0,0,6, 2,6,2,1, 6,2,3,2, 7,6,4,3, 0,2,3,5,
2,7,5,1, 5,1,7,6, 1,7,1,5, 4,4,0,7, 2,2,6,2, 2,2,6,3, 3,3,7,7, 3,2,4,5,

0,7,4,7, 7,7,2,3, 1,6,7,6, 5,7,0,5, 1,0,7,6, 2,4,0,2, 7,5,5,4, 1,5,1,5,
6,7,3,0, 2,7,6,6, 4,0,7,4, 3,2,2,6, 6,7,4,7, 2,0,2,7, 2,1,5,4, 6,2,3,2,

1,6,0,7, 1,1,2,6, 2,2,0,2, 2,3,6,7, 1,7,1,7, 1,5,7,7, 2,2,2,0, 4,3,4,2,
0,6,7,6, 0,5,0,7, 1,7,4,1, 2,3,4,6, 7,2,2,0, 6,4,4,6, 6,4,2,2, 6,5,3,4,

2,3,5,7, 7,1,0,0, 0,3,1,2, 0,1,6,2, 7,4,4,3, 2,5,4,5, 6,4,2,5, 6,2,2,4,
7,0,6,2, 3,7,2,5, 4,2,4,1, 5,5,3,6, 1,1,3,2, 7,5,7,0, 7,3,5,0, 0,1,2,0
```

The TLC/AGC guard sequence symbols are modulated directly as described in C.5.1.4.7, without undergoing the PN spreading described in C.5.1.4.6.

C.5.1.4.2 BW1 Acquisition preamble.

The BW1 acquisition preamble provides an opportunity for the receiver to detect the presence of the waveform and to estimate various parameters for use in data demodulation. The preamble component of BW1 is a sequence of 576 tribit symbols having the values shown in table C-XI. The preamble symbols are transmitted in the order shown in table C-XI, starting at the top left and moving from left to right across each row, and from top to bottom through successive rows.

The preamble symbols are modulated directly as described in C.5.1.4.7, without undergoing the PN spreading described in C.5.1.4.6.

When it detects a BW1 acquisition preamble, the BW1 receiver issues a BW1_Pre_Detect service primitive, as described in table C-IV.

TABLE C-XI. BW1 acquisition preamble symbol values.

```
4,4,7,3, 7,7,1,0, 4,7,1,6, 6,5,5,0, 6,4,0,6, 0,1,6,2, 2,7,3,5, 1,5,4,5,
5,6,6,0, 2,0,4,0, 7,0,2,6, 3,7,1,5, 2,3,2,3, 7,1,1,7, 0,0,0,7, 4,5,2,3,

2,3,7,3, 1,0,3,4, 2,5,6,6, 6,5,2,3, 2,7,6,7, 6,6,1,0, 1,2,6,5, 6,5,1,4,
3,5,2,6, 5,6,5,2, 5,2,0,0, 2,6,7,0, 4,2,2,5, 0,2,5,1, 5,2,1,2, 3,4,1,7,

4,1,0,5, 1,1,4,1, 6,2,6,5, 2,2,3,1, 7,1,0,6, 0,4,6,2, 3,3,6,1, 5,0,4,2,
5,1,4,6, 7,2,1,5, 4,1,4,7, 7,1,5,4, 1,7,0,2, 6,2,4,7, 1,1,3,0, 6,4,2,1,

7,3,2,5, 4,0,4,4, 5,4,6,7, 7,2,6,7, 1,1,4,6, 0,5,1,6, 6,1,2,3, 2,5,3,4,
5,2,0,4, 1,4,6,5, 2,6,3,2, 2,3,0,7, 7,0,2,2, 1,6,6,6, 0,5,1,3, 4,5,1,6,

7,2,2,2, 1,3,7,5, 7,0,6,6, 5,7,2,4, 0,3,0,6, 1,4,3,4, 0,1,5,4, 5,1,5,7,
6,5,6,4, 7,7,0,1, 4,3,5,6, 1,5,7,1, 5,3,1,0, 5,5,0,4, 2,2,2,5, 2,4,5,3,

6,2,6,3, 5,0,4,0, 0,7,3,5, 1,4,5,5, 2,5,2,6, 6,3,7,6, 0,2,7,1, 4,3,5,2,
6,1,2,0, 6,5,1,7, 1,0,6,3, 0,4,7,6, 0,5,0,4, 1,5,7,0, 4,6,6,1, 7,0,5,1,

6,0,6,4, 6,6,1,4, 6,3,3,2, 1,4,4,1, 4,6,7,2, 6,2,4,6, 1,0,5,0, 4,0,5,4,
4,2,5,2, 7,2,4,4, 7,3,6,4, 7,5,6,5, 6,5,5,3, 2,3,4,7, 5,7,2,7, 1,5,5,3,

0,3,5,4, 2,1,3,7, 1,5,4,4, 3,7,5,5, 5,4,7,0, 7,7,1,0, 5,4,7,0, 4,6,7,1,
0,0,3,4, 5,4,7,0, 3,3,2,2, 2,0,3,2, 7,0,0,3, 0,5,3,7, 1,4,2,3, 5,3,5,7,

1,3,3,1, 0,1,1,6, 5,1,5,1, 5,0,7,0, 2,5,7,6, 7,7,3,1, 0,3,1,4, 2,3,5,1,
4,0,2,1, 7,1,1,7, 4,5,0,1, 0,0,3,6, 6,6,6,3, 7,3,2,6, 0,3,7,5, 1,0,1,6
```

C.5.1.4.3 BW1 Forward error correction.

BW1 carries a payload of 48 protocol bits. The 48 protocol bits are coded using the r = 1/3, k = 9 convolutional encoder shown in figure C-8, creating 144 coded bits. A ‘tail-biting’ convolutional encoding approach is used as follows:

1. Initialize the eight memory cells x^1 .. x^8 of the encoder with the last eight bits of the payload sequence, p_{40} .. p_{47}, so that cell x^1 contains p_{40} and cell x^8 contains p_{47}.
2. Shift the first bit of the payload sequence, p_0, into cell x^8.
3. Extract the first three coded output bits $bitout_0$, $bitout_1$, and $bitout_2$, in that order, as shown in figure C-8.
4. Shift the next payload bit into cell x^8, then extract the three coded output bits $bitout_0$, $bitout_1$, and $bitout_2$.
5. Repeat step 4 until a total of 144 coded bits have been produced.

No flush bits are necessary for the encoding process. The polynomials used are:

- $\text{Bitout}_0 = x^8+x^7+x^6+x^3+1$
- $\text{Bitout}_1 = x^8+x^7+x^5+x^4+x^1+1$
- $\text{Bitout}_2 = x^8+x^6+x^5+x^3+x^2+x^1+1$

where x^8 corresponds to the most recent encoder input bit.

The order of output to the interleaving process is Bitout_0 then Bitout_1 then Bitout_2.

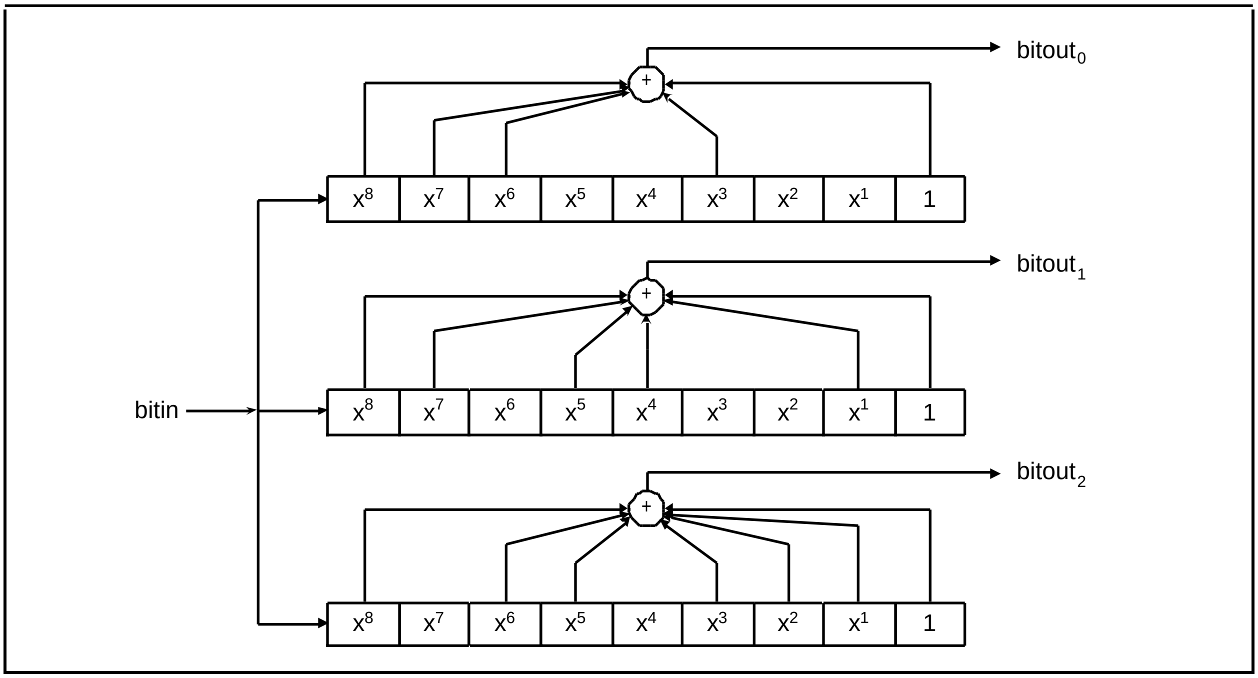


FIGURE C-8. Rate 1/3, constraint length 9 convolutional encoder.

C.5.1.4.4 BW1 Interleaving.

See C.5.1.2 for a description of the interleaving process. The interleaver parameters for BW1 are as follows:

TABLE C-XII. Interleaver parameters for BW1.

R	16
C	9
i_{rs}	0
i_{cs}	1
Δi_{rs}	1
Δi_{cs}	0
i_{rf}	1
i_{cf}	0
Δi_{rf}	0
Δi_{cf}	1

See C.5.1.2 for a complete description of the block interleaving process used by the various burst waveforms.

C.5.1.4.5 BW1 Orthogonal symbol formation.

The interleaver fetch process removes 4 coded bits at a time from the interleaver matrix. These 4 coded bits are mapped into a 16-tribit sequence using the mapping given in table C-VIII. Note that each of the four-bit sequences in the Coded Bits column of the table is of the form $b_3b_2b_1b_0$, where b_3 is the first bit fetched from the interleaver matrix. The tribit values are placed in the output tribit sequence in the order in which they appear in the corresponding row of table C-VIII, moving from left to right across the row. The 16-tribit sequence thus obtained is repeated 4 times to obtain a 64-tribit sequence. This process repeats for a total of 36 iterations to produce 2304 raw tribit values.

C.5.1.4.6 BW1 PN spreading sequence generation and application.

A sequence of 2304 pseudo-random tribit values s_i is generated by repeating the 256-tribit sequence presented in table C-XIII, 2304 / 256 = 9 times. The tribit values are used in the order shown in table C-XIII, starting at the top left and moving from left to right across each row, and from top to bottom through successive rows.

TABLE C-XIII. BW1 PN spreading sequence.

```
0,2,4,3, 3,6,4,5, 7,6,7,0, 5,5,4,3, 5,4,3,7, 0,7,6,2, 6,2,4,6, 7,2,4,7,
5,5,7,0, 7,3,3,3, 7,3,3,1, 4,2,3,7, 0,2,7,7, 3,5,1,0, 1,4,0,5, 0,0,0,0,

6,5,0,1, 2,7,6,5, 5,2,7,3, 3,3,2,1, 2,5,6,1, 3,4,2,1, 0,1,2,3, 6,4,7,5,
2,2,6,2, 7,6,5,2, 4,6,5,4, 7,2,5,1, 0,0,7,7, 3,5,4,2, 1,4,2,7, 0,3,4,0,

0,0,7,7, 3,5,4,2, 1,4,2,7, 0,3,4,0, 1,0,5,2, 6,0,3,5, 1,0,5,1, 5,2,5,6,
3,2,3,7, 1,2,2,0, 7,1,3,6, 4,2,6,2, 7,4,3,7, 6,7,2,3, 1,7,4,1, 5,1,5,4,

7,1,1,2, 3,6,7,7, 6,6,1,2, 2,4,1,7, 7,5,5,4, 7,7,5,0, 7,3,7,5, 7,7,5,0,
6,6,6,1, 3,4,4,4, 0,3,3,2, 1,4,5,4, 5,3,1,1, 1,2,5,1, 7,1,5,7, 2,0,0,6
```

The 2304 tribit values s_i of the PN sequence are then combined with the 2304 raw tribit values r_i produced by the orthogonal symbol formation process described in the previous section. Each symbol of the PN sequence s_i is combined with the corresponding symbol r_i of the raw tribit sequence to form a channel symbol c_i, by adding s_i to r_i modulo 8. For instance, if $s_i = 7$, $r_i = 4$, then $c_i = 7 \quad 4 = 3$, where the symbol   represents modulo-8 addition.

The process can be summarized:

$$\begin{matrix} c_0 \\ \vdots \\ c_{2303} \end{matrix} = \begin{matrix} r_0 \\ \vdots \\ r_{2303} \end{matrix} \qquad \begin{matrix} s_0 \\ \vdots \\ s_{2303} \end{matrix}$$

where r is the vector of data raw tribit values, s is the vector of PN sequence tribit values, c is the resulting vector of combined tribit values, and the symbol   represents component-wise modulo-8 addition.

C.5.1.4.7 BW1 Modulation.

The sequence of channel symbols consisting of

- the TLC/AGC guard sequence of 256 tribit symbols described by C.5.1.4.1 (on which no PN-spreading has been performed), followed by
- the acquisition preamble sequence of 576 tribit symbols described by C.5.1.4.2 (on which no PN-spreading has been performed), followed by
- the 2304-length sequence of BW1 channel symbols (data symbols), PN-spread as described in C.5.1.4.6,

is used to PSK modulate an 1800 Hz carrier signal at 2400 channel symbols/sec.

See C.5.1.8 for a description of how the channel symbol values are mapped to carrier phases and the subsequent carrier modulation process.

C.5.1.5 Burst Waveform 2 (BW2).

Burst Waveform 2 (BW2) is used for transfers of traffic data by the HDL protocol. Figure C-9 summarizes the structure and timing characteristics of the BW2 waveform.

BW2 is used to transmit a sequence of data packets from a transmitting station to a receiving station, where a data packet is defined as a fixed-length sequence of precisely 1913 data bits. The HDL protocol process (described in C.5.4) causes BW2 to modulate a Forward Transmission containing a sequence of data packets by invoking the BW2_Send primitive. The BW2_Send primitive has two parameters:

- payload: a sequence of NumPKTs data packets, where NumPKTs = 3, 6, 12, or 24; and
- reset: a boolean parameter which is set to TRUE by the HDL protocol for the first Forward Transmission performed in delivering a datagram, and set to FALSE at all other times. reset = TRUE causes counters used in BW2's FEC encoding and PN spreading processes to be reinitialized.

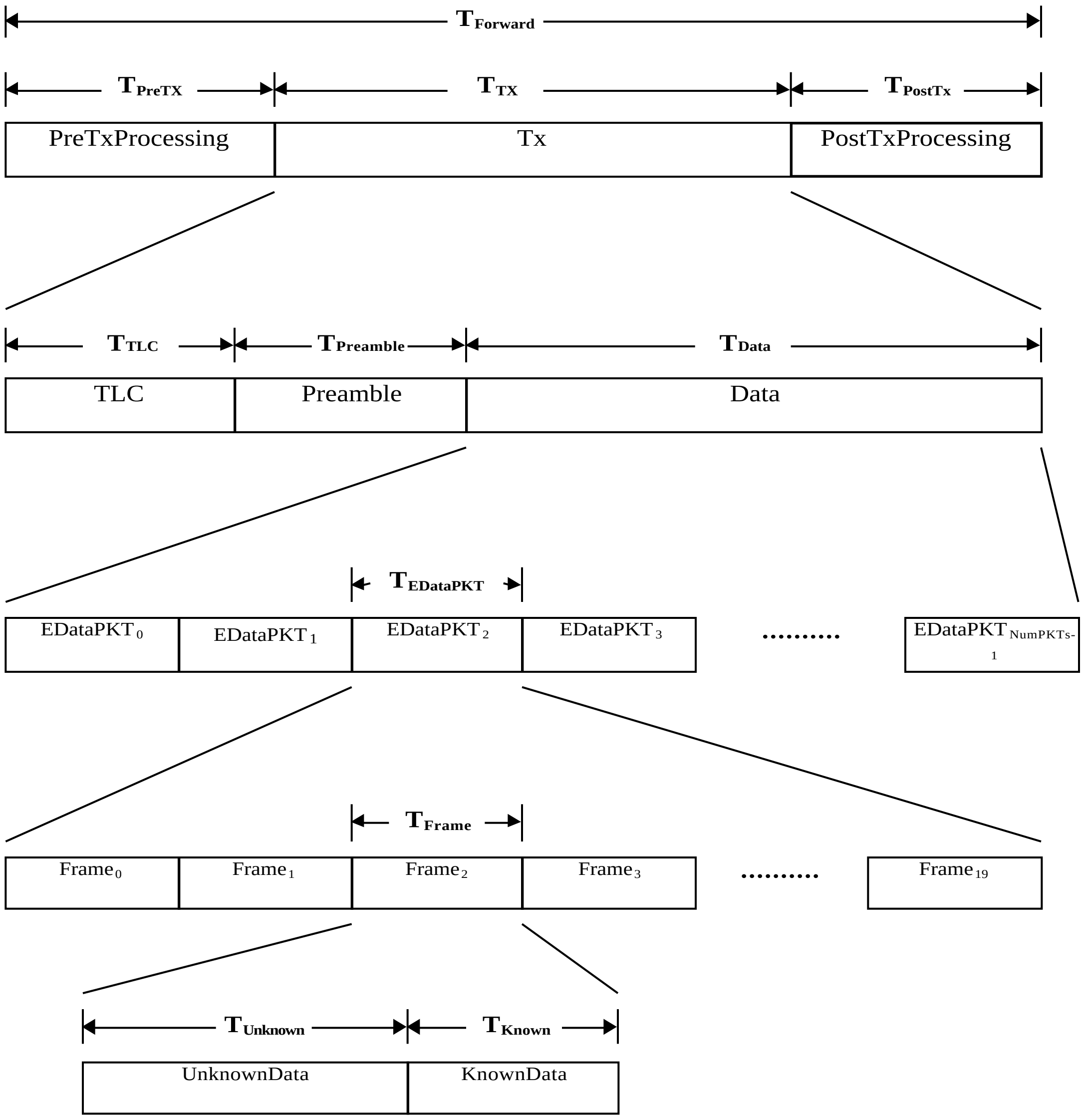
C.5.4 describes the manner in which the HDL protocol determines the values assigned to these parameters.

The total duration of the Forward Transmission phase of the HDL protocol is 0.64 + (0.40 * NumPKTs) seconds, and includes a constant-length portion and a variable-length portion. The constant-length portion has a fixed duration of 0.64 seconds ($T_{FORWARD}$ - T_{DATA}), which includes:

- a PreTxProcessing interval of 293.33 ms (704 PSK symbol times, at 2400 symbols per second), in which no waveform is transmitted or received
- a PostTxProcessing interval of 220 ms (528 PSK symbol times, at 2400 symbols per second), in which no waveform is transmitted or received
- a TLC/AGC guard sequence of 240 PSK symbols, with a duration of 100 ms (T_{TLC})
- a BW2 acquisition preamble sequence of 64 PSK symbols, with a duration of 26.67 ms (T_{PRE}).

The variable-length portion has a duration (T_{DATA}) of 400 * NumPKTs milliseconds (equal to 960 * NumPKTs PSK symbol times).

The BW2 modulation process uses a count variable, FTcount, to keep track of how many Forward Transmissions have occurred in transmitting the current datagram. At the start of each Forward Transmission, FTcount is initialized to zero if and only if the reset parameter of the current invocation of BW2_Send is TRUE. At the end of each Forward Transmission, FTcount is incremented. The value of FTcount is used in FEC encoding (as described in C.5.1.5.5), in rotating the modulation symbols containing FEC-coded data (as described in C.5.1.5.8), and in generating the spreading symbol sequence used to PN-spread the BW2 gray-coded modulation symbols (as described in C.5.1.5.8).



T_{Known} = 16 8-PSK symbols at 2400 symbols/sec = 6.67 ms
$T_{Unknown}$ = 32 8-PSK symbols at 2400 symbols/sec = 13.33 ms
T_{Frame} = 20.0 ms
$T_{EDataPKT}$ = 20 * 20.0 ms = 400 ms
T_{Data} = NumPKTs * 400 ms
$T_{Preamble}$ = 64 8-PSK symbols at 2400 symbols/sec = 26.67 ms
T_{TLC} = 240 8-PSK symbols at 2400 symbols/sec = 100.00 ms
T_{TX} = 126.67 + (NumPKTs * 400) ms
T_{PostTX} = 220 ms
T_{PreTX} = 293.33 ms
$T_{Forward}$ = 0.64 + (NumPKTs * 0.40) seconds

FIGURE C-9. BW2 waveform structure and timing characteristics.

The subsections of this section describe the manner in which the values of the symbols in the TLC/AGC guard sequence, the preamble sequence, and the variable-length data portion of each

Forward Transmission are determined, and then describe the manner in which the resulting symbol sequence is PN-spread and modulated.

C.5.1.5.1 BW2 TLC/AGC guard sequence.

The TLC/AGC guard sequence portion of the BW2 waveform provides an opportunity for both the transmitting radio's Transmit Level Control process (TLC) and the receiving radio's Automatic Gain Control process (AGC) to reach steady states before the BW2 preamble appears at their respective inputs, minimizing the distortion to which the preamble can be subjected by these processes. The BW2 TLC/AGC guard sequence is composed of the first 240 of the pseudo-random tribit symbol values shown in table C-X. The tribit symbols are transmitted in the order shown in table C-X, starting at the top left and moving from left to right across each row, and from top to bottom through successive rows.

For convenience of implementation, the length of the TLC/AGC guard sequence (240 PSK symbols) has been chosen so as to be an integral multiple of the length of an unknown/known symbol frame as described in C.5.1.5.7.

The TLC/AGC guard sequence symbols are modulated directly as described in C.5.1.5.9, without undergoing the PN spreading described in C.5.1.5.8.

C.5.1.5.2 BW2 Acquisition preamble.

The BW2 acquisition preamble is a sequence of 64 tribit symbols all having values of zero (000). The BW2 acquisition preamble symbols undergo PN spreading as described in C.5.1.5.8; the PN-spread preamble symbols are then modulated as described in C.5.1.5.9.

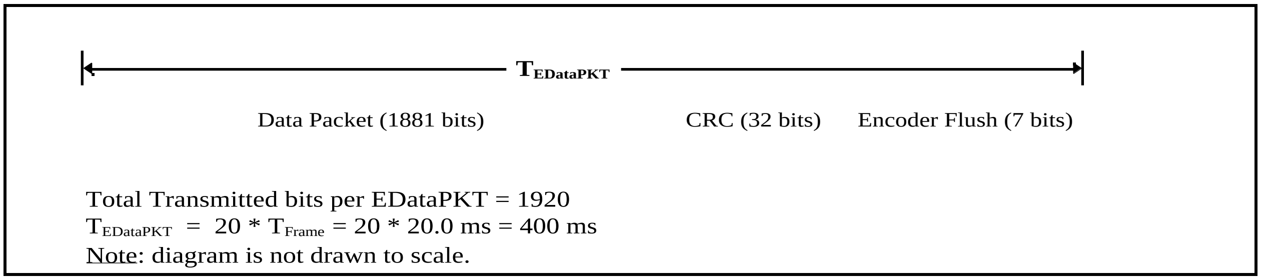
C.5.1.5.3 BW2 CRC computation.

A 32-bit Cyclic Redundancy Check (CRC) value is computed across the 1881 payload data bits in each data packet, and is then appended to the data packet. The generator polynomial used in computing the CRC is:

$$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^{8} + X^{7} + X^{5} + X^{4} + X^{2} + X^{1} + 1.$$

Other details of the CRC computation procedure are as defined in C.4.1.

C.5.1.5.4 BW2 Data packet extension.



Total Transmitted bits per EDataPKT = 1920
$T_{EDataPKT}$ = 20 * T_{Frame} = 20 * 20.0 ms = 400 ms
Note: diagram is not drawn to scale.

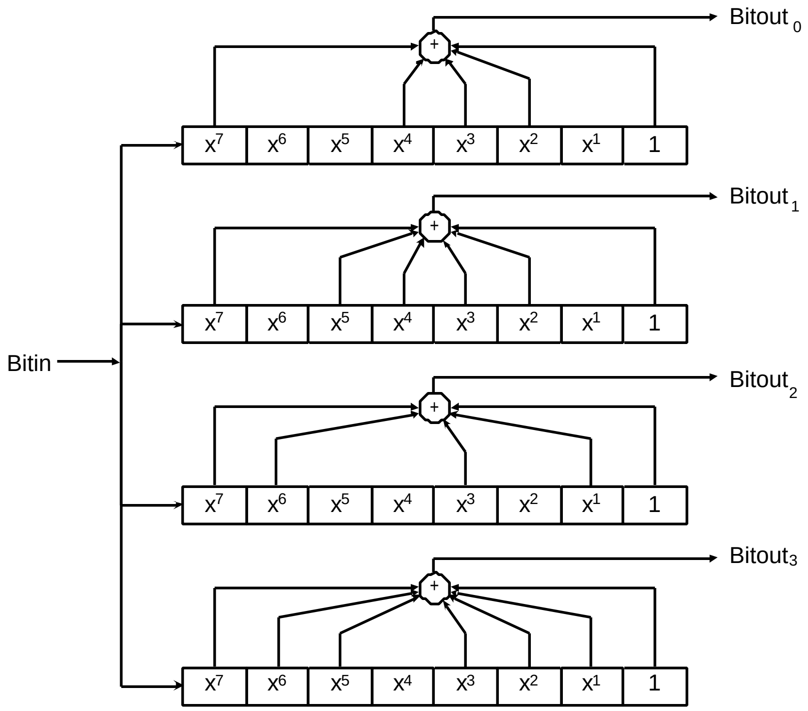
FIGURE C-10. Data packet extension with encoder flush bits.

As is shown in figure C-10, seven encoder flush bits with values of zero are appended to the 1913 payload and CRC bits of each data packet to produce an extended data packet, known henceforth as an ‘EDataPkt’ (i.e., an “Extended Data Packet”). Note that the further processing (FEC, symbol formation, and frame formation) of each EDataPKT is not affected by the presence of the CRC and flush bits in the EDataPKT; in these processes, each EDataPKT is treated as an arbitrary sequence of 1920 bits. As described below, each 1920-bit EDataPKT is transformed into 20 frames of 48 PSK symbols each. Each of the 32 known 8-PSK symbols in a frame carries three data bits, so that each frame carries 96 of the 1920 bits in an EDataPKT.

C.5.1.5.5 BW2 Forward error correction.

The 1920 bits in each EDataPkt are convolutionally encoded using the rate 1/4, constraint length 8, non-systematic convolutional encoder shown in figure C-11. The encoder produces 4 encoded bits: Bitout_0, Bitout_1, Bitout_2, and Bitout_3, for each single input bit. As each EDataPkt is encoded, the coded bits from each of the four coded bit streams are accumulated into a block of 1920 coded bits. This produces a total of four Encoded Blocks of 1920 bits each (EBlk_0 through $\text{EBlk}_{3,}$ where each EBlk_k is composed solely of output bits from Bitout_{k}.). Only one of the four Encoded Blocks resulting from the encoding of each EDataPkt is transmitted in each Forward Transmission. Which of the four Encoded Blocks is transmitted is determined by the value of the BW2 modulation process’s FTcount variable: the Encoded Block transmitted is EBlk_n (containing coded data bits from Bitout_n), where n = FTcount mod 4. For instance, the sixth Forward Transmission of a datagram contains EBlk_1 (since FTcount = 5 and 5 mod 4 = 1) for every EDataPkt in the Forward Transmission. Each successive retransmission of a EDataPkt contains a different Encoded Block derived from the EDataPkt contents, providing the decoder at the remote station with additional information as to the contents of the EDataPkt.

The seven zeroes in the Encoder Flush field at the end of each EDataPkt return the convolutional encoder to its initial (all-zero) state before it starts to encode the contents of the next EDataPkt.



The generator polynomials corresponding to the four shift registers shown here are respectively (from top to bottom):

1. Bitout_0: $X^7 + X^4 + X^3 + X^2 + 1$
2. Bitout_1: $X^7 + X^5 + X^4 + X^3 + X^2 +1$
3. Bitout_2: $X^7 + X^6 + X^3 + X^1 + 1$
4. Bitout_3: $X^7 + X^6 + X^5 + X^3 + X^2 + X^1 +1$,

where X^7 corresponds to the bit most recently input to the encoder.

FIGURE C-11. Rate 1/4, constraint length 8 convolutional encoder.

C.5.1.5.6 BW2 Modulation Symbol formation.

Once the NumPKTs encoded blocks for each forward transmission have been produced, the contents of the encoded blocks are formed into three-bit modulation symbols. Each modulation symbol is formed by taking three bits one at a time from the current Encoded Block, starting with the first bit of the first Encoded Block, and shifting them successively into the modulation symbol's least significant bit-position (so that the first bit of the three is eventually placed in the most significant bit-position). This continues until 1920/3 = 640 modulation symbols have been formed for each Encoded Block.

The modulation symbols for all Encoded Blocks are then rotated toward the least significant bit-position (so that $M_2M_1M_0$ becomes $M_0M_2M_1$), FTcount mod 3 times. This causes each successive transmission of an Encoded Block to have its data contents mapped onto different modulation symbol values. After this rotation has been performed, the rotated modulation symbols are gray-coded as shown in table C-XIV, yielding a sequence of gray-coded modulation symbols.

TABLE C-XIV. Gray coding table.

Input Data **(Modulation Symbol)**	**Output Data** **(Gray-Coded Modulation Symbol)**
000	000
001	001
010	011
011	010
100	111
101	110
110	100
111	101

C.5.1.5.7 BW2 Frame formation.

Once the NumPKTs Encoded Blocks have been formed into modulation symbols, rotated, and gray-coded, the resulting gray-coded modulation symbols are formed into Frames. Each Frame (as shown on figure C-9) is formed by taking the next 32 consecutive gray-coded modulation symbols (known as “unknown symbols” because they contain coded payload data not known a priori) from the sequence produced as described in the previous section, and appending to them 16 known symbols having symbol values equal to zero (000). The 640 gray-coded modulation symbols for each Encoded Block are incorporated into the unknown sections of 20 Frames (since 640/32 = 20). For a Forward Transmission containing NumPKTs EDataPkts, there will therefore be (20 * NumPKTs) Frames, each containing 32 gray-coded modulation symbols (unknown symbols) derived from encoded payload data, followed by 16 known symbols having values of zero.

C.5.1.5.8 BW2 PN spreading sequence generation and application.

The length 2^{16}-1 Maximum-Length Sequence Generator shown in figure C-12 is used to PN-spread the sequence of modulation symbols (tribits) consisting of the 64 symbols of the BW2 acquisition preamble described by C.5.1.5.2, followed by the (960 * NumPKTs) gray-coded modulation symbols generated as described in sections C.5.1.5.3 through C.5.1.5.7.

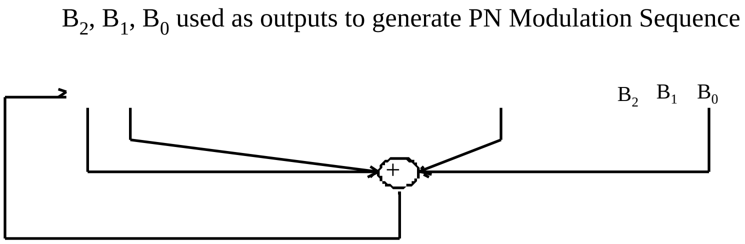

B2, B1, B0 used as outputs to generate PN Modulation Sequence
B2
B1
B0
+

FIGURE C-12. 2^{16} - 1 maximum-length sequence generator.

The Forward Transmission count variable FTcount (described in C.5.1.5) is used in initializing the state of the sequence generator: at the start of each Forward Transmission, the state of the generator is initialized to (0xAB91 + FTcount) mod 0x 10000.

The outputs of the sequence generator are used to PN-spread the modulation symbols as follows:

1. For each input symbol (preamble symbol or gray-coded modulation symbol), a three-bit spreading symbol is formed by cycling the PN generator three times, and then taking the three least significant bits B_2, B_1, and B_0 (as shown in figure C-12) from the shift register, with B_2 becoming the most significant bit of the spreading symbol.
2. The spreading symbol is then summed modulo 8 with the input symbol to form a three-bit channel symbol.

This is performed for each of the 64 preamble symbols and each of the (960 * NumPKTs) gray-coded modulation symbols. Note that since all of the preamble symbols and the known modulation symbols were filled with zero values (000), and the Gray-coding of zero yields zero, the preamble channel symbols and the known channel symbols actually contain the spreading symbols.

C.5.1.5.9 BW2 Modulation.

The sequence of channel symbols consisting of:

- the TLC/AGC guard sequence described by C.5.1.5.1 (on which no gray-coding or PN-spreading has been performed), followed by
- the 64-length sequence of BW2 acquisition preamble symbols described by C.5.1.5.2, PN-spread as described in C.5.1.5.8; followed by
- the (960 * NumPKTs) gray-coded modulation symbols generated as described in C.5.1.5.3 through C.5.1.5.7, and PN-spread as described in C.5.1.5.8,

is used to PSK modulate an 1800 Hz carrier signal at 2400 channel symbols/sec.

See C.5.1.8 for a description of how the channel symbol values are mapped to carrier phases and the subsequent carrier modulation process.

C.5.1.6 Burst Waveform 3 (BW3).

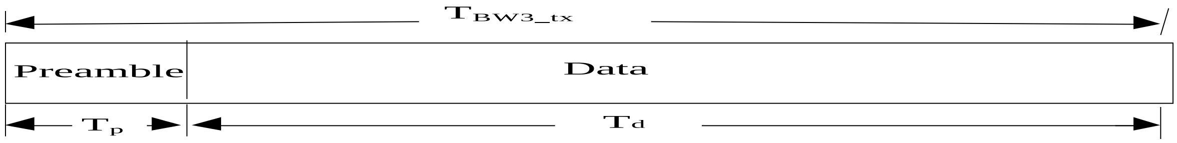
Burst Waveform 3 (BW3) is used for transfers of traffic data by the LDL protocol. Figure C-13 summarizes the structure and timing characteristics of the BW3 waveform.

BW3 is used to transmit a single LDL data packet from a transmitting station to a receiving station. An LDL data packet is defined as a fixed-length sequence of bits. The number of bits in a BW3 data packet is of the form 8n+25, where n = 32*m bytes. The value ‘n’ used throughout this section refers to the number of payload data bytes (or octets) carried by each LDL protocol forward transmission; the additional 25 bits of payload in each BW3 transmission are LDL overhead. The value ‘m’ takes on integer values in the range 1 through 16, so n ranges from 32 through 512 bytes. BW3 is used only to deliver forward transmissions of the LDL protocol described in C.5.5.

The LDL protocol process causes the generation of a BW3 burst by invoking the BW3_Send primitive. The BW3_Send primitive has two parameters:

- payload: the (8n+25)-bit data packet to be transmitted; and
- reset: a boolean parameter which is set to TRUE by the LDL protocol for the first Forward Transmission performed in delivering a datagram, and set to FALSE at all other times. Reset = TRUE causes the Forward Transmission counter FTcount used in BW3’s FEC encoding process to be reinitialized.

C.5.5 describes the manner in which the LDL protocol determines the values assigned to these parameters.



T_p = 266.67 milliseconds: 640 PSK symbols at 2400 symbols/second

T_d = 106.67 + (n*13.33) milliseconds: 32n + 256 PSK symbols at 2400 symbols/second,

n = 32*m bytes,

m = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16.

T_{BW3_tx} = 373.33 + (n*13.33) milliseconds, which ranges from 0.800 through 7.200 seconds

FIGURE C-13. BW3 timing.

The BW3 modulation process maintains a count variable, FTcount, to keep track of how many forward transmissions have occurred in transmitting the current datagram. FTcount is initialized to zero upon reception of a BW3_Send PDU having its Reset parameter set to TRUE. At the end of each BW3 forward transmission, FTcount is incremented by one. The value of FTcount is used in FEC encoding as described in C.5.1.6.3.

The description of BW3 waveform generation will proceed as follows:

- Section C.5.1.6.1 discusses generation of tribit values for the Preamble waveform component.
- Sections C.5.1.6.2, C.5.1.6.3, C.5.1.6.4, and C.5.1.6.5 discuss the mapping of input bits to raw tribit values for the data waveform component via CRC computation, FEC, interleaving, and orthogonal Walsh symbol formation.
- Section C.5.1.6.6 discusses the generation of tribit values for the PN spreading sequence and the combining of these PN spreading sequence tribit values with the raw tribit values for the data waveform component.
- Section C.5.1.6.8 discusses carrier modulation using the preamble and PN-spread data tribit values.

C.5.1.6.1 BW3 Preamble.

This portion of the burst waveform provides an opportunity for both the transmitting radio’s Transmit Level Control process (TLC) and the receiving radio’s Automatic Gain Control process (AGC) to reach steady states, and provides an opportunity for the receiver to detect the presence of the waveform and to estimate various channel parameters for use in data demodulation. The preamble component of BW3 is a sequence of 640 tribit values having the values shown in table C-XV. The preamble symbols are transmitted in the order shown in table C-XV, starting at the top left and moving from left to right across each row, and from top to bottom through successive rows. The preamble symbols are modulated directly as described in section C.5.1.4.7, without undergoing PN spreading as described in section C.5.1.6.6.

Unlike the other burst waveforms specified in this appendix, a TLC/AGC guard sequence is not provided as an explicit part of the BW3 waveform, since the correlation-based receive processing of the BW3 waveform is relatively insensitive to such signal perturbations as are likely to be introduced by the TLC and AGC processes. The duration of the BW3 preamble includes sufficient time for preamble acquisition to be performed after the TLC and AGC processes have settled.

TABLE C-XV. BW3 preamble symbol values.

7,2,3,5,	7,5,0,6,	7,5,3,5,	6,3,5,4,	7,1,4,5,	7,7,3,5,	0,5,1,6,	0,3,1,0,
2,7,6,2,	4,4,2,0,	0,7,2,0,	2,3,7,4,	1,1,3,0,	1,3,6,3,	0,1,3,1,	5,4,5,6,
3,2,5,2,	6,0,6,6,	0,4,6,3,	7,1,7,0,	6,2,1,5,	5,2,2,5,	3,3,3,2,	1,4,7,0,
0,2,0,2,	5,7,5,7,	7,5,3,6,	2,2,1,6,	4,4,7,1,	5,4,7,2,	7,5,6,1,	1,5,0,0,
1,4,0,5,	3,4,7,3,	6,2,2,5,	4,0,2,7,	6,2,7,1,	6,5,5,3,	2,3,2,5,	7,7,3,7,
3,2,2,2,	4,0,0,7,	5,4,5,3,	5,0,3,3,	3,0,6,4,	6,5,6,4,	2,7,6,2,	6,6,1,0,
5,1,0,7,	1,4,2,7,	6,0,1,6,	7,5,6,1,	1,7,5,1,	0,0,1,0,	3,1,7,4,	5,4,4,5,
4,3,2,0,	4,1,6,6,	2,7,6,4,	4,6,2,2,	3,0,3,5,	2,1,1,6,	2,7,6,2,	2,5,7,1,
2,5,5,6,	4,0,7,1,	7,2,3,2,	5,2,0,2,	2,2,0,3,	6,6,6,2,	5,5,5,6,	0,0,2,3,
6,7,6,5,	7,2,2,4,	5,5,2,5,	7,3,2,7,	0,3,0,1,	4,1,6,2,	5,7,0,1,	6,0,1,6,
5,1,3,6,	5,4,2,0,	4,4,2,1,	2,6,1,1,	0,1,1,3,	5,7,5,0,	4,3,1,5,	3,0,0,4,
5,6,7,5,	7,6,1,5,	5,1,2,7,	5,0,3,6,	3,5,2,7,	0,6,6,0,	6,5,4,2,	7,5,6,0,
4,1,7,0,	4,7,4,7,	3,1,2,3,	7,2,2,6,	7,5,1,6,	6,7,2,5,	6,4,0,3,	0,4,7,1,
6,2,5,4,	3,6,0,6,	6,5,3,3,	4,4,5,1,	2,6,7,3,	1,3,0,7,	7,4,6,2,	5,2,0,7,
3,6,7,6,	3,6,3,1,	4,4,6,3,	7,7,6,4,	4,5,2,2,	5,4,7,4,	5,6,2,6,	0,2,4,6,
3,3,4,3,	5,5,0,7,	6,3,1,6,	0,2,2,0,	4,2,6,7,	7,2,0,5,	1,3,7,6,	3,7,2,0,
1,6,3,5,	1,0,3,7,	5,4,6,7,	2,4,0,0,	2,2,7,1,	2,6,3,3,	7,1,7,7,	4,1,2,2,
5,4,0,3,	3,5,6,1,	0,4,5,6,	7,1,2,0,	3,1,6,2,	4,6,1,5,	6,7,7,2,	6,3,7,6,
7,2,3,4,	4,4,6,0,	4,3,7,7,	1,5,7,1,	3,4,5,6,	6,3,2,3,	4,4,0,1,	4,0,3,6,
7,3,5,0,	3,0,7,1,	0,5,4,5,	4,4,3,7,	6,1,1,5,	0,1,1,1,	4,6,0,7,	2,5,4,3

C.5.1.6.2 BW3 CRC computation.

A 32-bit Cyclic Redundancy Check (CRC) value is computed across the payload data bits in the data packet, and is then appended to the data packet. The generator polynomial used in computing the CRC is:

$$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^{8} + X^{7} + X^{5} + X^{4} + X^{2} + X^{1} + 1.$$

Other details of the CRC computation procedure are as defined in C.4.1

C.5.1.6.3 BW3 Forward error correction.

7 flush bits having the value 0 are appended to the (8n+57) bits of the data packet with CRC to ensure that the encoder is in the all-zero state upon encoding the last flush bit. The data and CRC bits and the 7 flush bits are coded using the r = 1/2, k = 7 convolutional encoder shown in C-14.

NOTE 1. Since BW3 uses a k=7 convolutional code, only 6 bits are literally needed to flush the encoder. The seventh ‘flush bit’ is added purely for convenience -- to make the number of coded bits per BW3 transmission a multiple of four, so that each group of four bits can then be mapped to an orthogonal symbol as described below.

NOTE 2. The generator polynomials corresponding to Bitout0 and Bitout1 are:

- Bitout_0: $X^6+X^4+X^3+X+1$

- Bitout$_1$: $X^6+X^5+X^4+X^3+1$
- where X^6 corresponds to the most recent input bit.

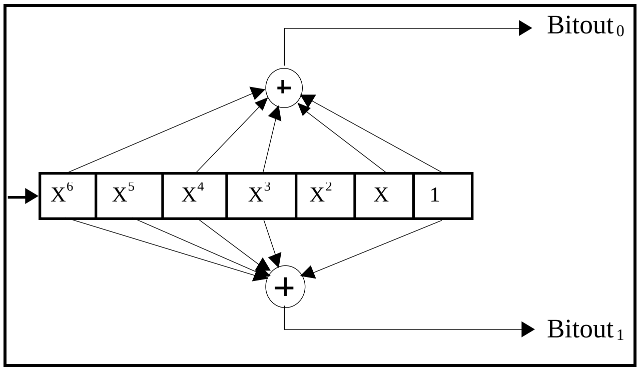


FIGURE C-14. BW3 rate 1/2, k=7 convolutional encoder.

This encoder produces two encoded bits, Bitout$_0$ and Bitout$_1$, for each single input bit. Encoding an entire sequence of (8n+57) data and CRC bits followed by 7 flush bits results in two encoded blocks of (8n+64) coded bits each, EBlk$_0$ and EBlk$_1$, where each EBlk$_k$ is made up solely of output bits from Bitout$_k$. In each forward transmission, only coded bits from EBlk$_{(FTcount\ mod\ 2)}$ are passed forward to the interleaving process, where FTcount is the forward transmission count variable described in C.5.1.6; the encoded bits from the other encoded block are retained to possibly be transmitted in one or more subsequent forward transmissions. For instance, the fourth forward transmission of a data packet contains the coded bits from EBlk$_1$ (since FTcount = 3 and 3 mod 2 = 1).

C.5.1.6.4 BW3 Interleaving.

See C.5.1.2 for a description of the interleaving process. The interleaver parameters for BW3 depend on the value n (which determines the BW3 payload size), as shown in table C-XVI.

TABLE C-XVI. Interleaver parameters for BW3.

n	64	128	256	512
R	24	32	44	64
C	24	34	48	65
i_{rs}	7	7	7	7
i_{cs}	0	0	0	0
Δi_{rs}	0	0	0	0
Δi_{cs}	1	1	1	1
i_{rf}	1	1	1	1
i_{cf}	-7	-7	-7	-7
Δi_{rf}	0	0	0	0
Δi_{cf}	-7	-7	-7	-7

C.5.1.6.5 BW3 Orthogonal symbol formation.

The interleaver fetch process removes 4 coded bits at a time from the interleaver matrix. These 4 coded bits are mapped into a 16-tribit sequence using the mapping given in tabel C-VIII. Note that each of the four-bit sequences in the Coded Bits column of the table is of the form $b_3b_2b_1b_0$, where b_3 is the first bit fetched from the interleaver matrix. The tribit values are placed in the output tribit sequence in the order in which they appear in the corresponding row of table C-VIII, moving from left to right across the row. This process repeats for a total of 2n+16 iterations to produce the 32n+256 raw tribit values of the data portion of BW3.

C.5.1.6.6 BW3 PN spreading sequence generation and application.

A sequence of 32n+896 pseudo-random tribit values s_i is generated by repeating the 32-tribit sequence presented in table C-XVII, (32n+256) / 32 = n+8 times. The tribit values are used in the order shown in table C-XVII, starting at the top left and moving from left to right across each row, and from top to bottom through successive rows.

TABLE C-XVII. BW3 PN spreading sequence.

0,0,0,0,	0,2,4,6,	0,4,0,4,	0,6,4,2,
0,0,0,0,	1,3,5,7,	2,6,2,6,	3,1,7,5

The 32n+256 tribit values s_i of the PN sequence are then combined with the 32n+256 raw tribit values r_i produced by the orthogonal symbol formation process described in the preceding section. Each symbol of the PN sequence s_i is combined with the corresponding symbol r_i of the raw tribit (data) sequence to form a channel symbol c_i, by adding s_i to r_i modulo 8. For instance, if s_i = 7, r_i = 4, then c_i = 7   4 = 3, where the symbol   represents modulo-8 addition.

The process can be summarized:

$$\begin{matrix} c_0 \\ \vdots \\ c_{32n+255} \end{matrix} = \begin{matrix} r_{d0} \\ \vdots \\ r_{d32n+255} \end{matrix} \quad \begin{matrix} s_0 \\ \vdots \\ s_{32n+255} \end{matrix}$$

where r_d is the vector of data raw tribit values, s is the vector of PN sequence tribit values, c is the resulting vector of combined tribit values, and the symbol represents component-wise modulo-8 addition.

C.5.1.6.7 BW3 Modulation.

The sequence of channel symbols consisting of

- the preamble sequence of 640 tribit symbols described by section C.5.1.6.1 (on which no PN-spreading has been performed), followed by
- the sequence of (32n+256) BW3 channel symbols (data symbols), PN-spread as described in section C.5.1.6.6,

is used to PSK modulate an 1800 Hz carrier signal at 2400 Channel Symbols/sec. See section C.5.1.8 for a description of how combined tribit values are mapped to carrier phases and the subsequent carrier modulation process.

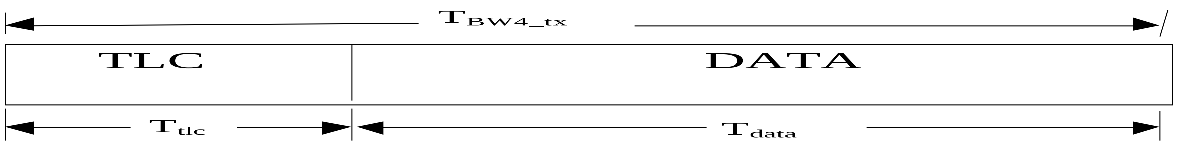
C.5.1.7 Burst Waveform 4 (BW4).

Burst Waveform 4 (BW4) is used to convey the LDL protocol’s LDL_ACK PDU. Figure C-15 summarizes the structure and timing characteristics of the BW4 waveform.

A user process (the LDL protocol) causes the generation of a BW4 waveform by issuing a BW4_Send primitive. The BW4_Send primitive has one parameter:

- payload: the two bits of payload data to be transmitted.

C.5.5 describes the manner in which values are assigned to the payload parameter.



T_{tlc} = 106.666 milliseconds: 256 PSK symbols at 2400 symbols/second

T_{data} = 533.333 milliseconds: 1280 PSK symbols at 2400 symbols/second

T_{BW4_tx} = 640.0 milliseconds

FIGURE C-15. BW4 timing.

The description of the BW4 waveform generation will proceed as follows:

- C.5.1.7.1 discusses generation of raw tribit values for the TLC/AGC guard sequence
- C.5.1.7.2 discusses the mapping of input bits to the raw tribit values for the Data waveform component.
- C.5.1.7.3 discusses the combining of raw tribit values with the PN spreading sequence tribit values.
- C.5.1.7.4 discusses carrier modulation using combined tribit values.

C.5.1.7.1 BW4 TLC/AGC guard sequence.

The TLC/AGC guard sequence portion of the BW4 waveform provides an opportunity for both the transmitting radio's Transmit Level Control process (TLC) and the receiving radio's Automatic Gain Control process (AGC) to reach steady states before the BW4 preamble appears at their respective inputs, minimizing the distortion to which the preamble can be subjected by these processes. The TLC/AGC guard sequence is a sequence of 256 pseudo-random tribit symbols having the values shown in table C-X. The tribit symbols are transmitted in the order shown in table C-X, starting at the top left and moving from left to right across each row, and from top to bottom through successive rows.

The TLC/AGC guard sequence symbols are modulated directly as described in C.5.1.7.4, without undergoing PN spreading as described in C.5.1.7.3.

C.5.1.7.2 BW4 Orthogonal symbol formation.

BW4 carries a payload of two protocol bits. The two protocol bits are mapped into a 16-tribit sequence using the mapping given in table C-XVIII. Note that each of the two-bit sequences in the Payload Bits column of the table is of the form b_1b_0, where b_1 is the first payload bit. The tribit values are placed in the output tribit sequence in the order in which they appear in the corresponding row of table C-XVIII, moving from left to right across the row. The 16-tribit sequence thus obtained is repeated 80 times to produce 1280 tribit values.

TABLE C-XVIII. Walsh modulation of BW4 payload bits to tribit sequences.

Payload Bits (shown as b_1b_0)	Tribit Sequence
00	0000 0000 0000 0000
01	0404 0404 0404 0404
10	0044 0044 0044 0044
11	0440 0440 0440 0440

C.5.1.7.3 BW4 PN spreading sequence generation and application.

The BW4 PN spreading sequence is the sequence of 1280 pseudo-random tribit values s_i shown in table C-XIX. The tribit values are used in the order shown in table C-XIX, starting at the top left and moving from left to right across each row, and from top to bottom through successive rows.

symbol of the PN sequence s_i is combined with the corresponding symbol r_i of the raw tribit sequence to form a channel symbol c_i, by adding s_i to r_i modulo 8. For instance, if s_i = 7, r_i = 4, then c_i = 7 4 = 3, where the symbol represents modulo-8 addition.

The process can be summarized:

$$\begin{array}{ccccc} c_0 & & r_0 & & s_0 \\ \vdots & = & \vdots & & \vdots \\ c_{1279} & & r_{1279} & & s_{1279} \end{array}$$

where r is the vector of data raw tribit values, s is the vector of PN sequence tribit values, c is the resulting vector of combined tribit values, and the symbol represents component-wise modulo-8 addition.

C.5.1.7.4 BW4 Modulation.

The sequence of channel symbols consisting of:

- the TLC/AGC guard sequence of 256 tribit symbols described by C.5.1.7.1 (on which no PN-spreading has been performed), followed by
- the 1280-length sequence of BW4 channel symbols (data symbols), PN-spread as described in C.5.1.7.3,

is used to PSK modulate an 1800 Hz carrier signal at 2400 channel symbols/sec.

See C.5.1.8 for a description of how combined tribit values are mapped to carrier phases and the subsequent carrier modulation process.

C.5.1.8 Burst waveform modulation.

The burst waveform descriptions have thus far only discussed the mapping of protocol bits to tribit values. This section will describe the process by which the tribit values are used to create the transmitted signal.

The transmitted signal consists of a 8-ary phase-shift-keyed 1800Hz single-tone carrier modulated at a constant 2400 symbols per second. The phase shift of the signal relative to that of the unmodulated carrier is a function of the tribit values as given in the tribit-value-to-carrier-phase mapping of table C-XX:

TABLE C-XX. 8-ary PSK signal space.

Tribit Value	Phase Shift	In-Phase (I)	Quadrature (Q)
0	0	1.0	0.0
1	$\pi/4$	$1/\sqrt{2}$	$1/\sqrt{2}$
2	$\pi/2$	0.0	1.0
3	$3\pi/4$	$-1/\sqrt{2}$	$1/\sqrt{2}$
4	π	-1.0	0.0
5	$5\pi/4$	$-1/\sqrt{2}$	$-1/\sqrt{2}$
6	$3\pi/2$	0.0	-1.0
7	$7\pi/4$	$1/\sqrt{2}$	$-1/\sqrt{2}$

The transmitted waveform is generated as illustrated in C-16. The tribit values are converted to the complex 8-PSK resulting in separate In-Phase [I] and Quadrature [Q] waveforms as given in C-XX. These waveforms are interpolated and independently filtered by equivalent low pass filters to provide spectral containment and image rejection. Finally, the interpolated and filtered In-phase and Quadrature waveforms are used to modulate the f = 1800 Hz sub-carrier. The accuracy of the clock linked with the generation of the sub-carrier frequency is 1 part in 10^5.

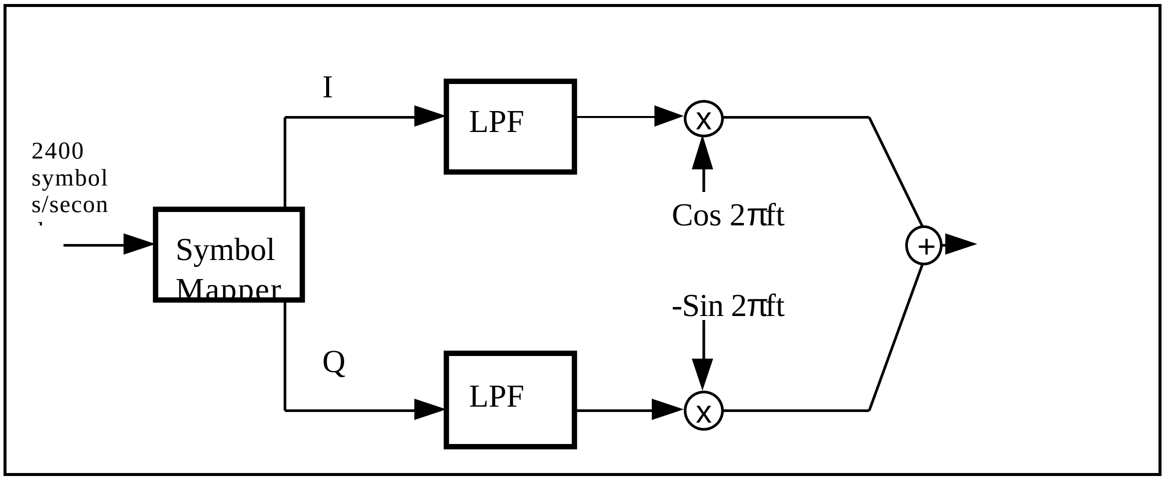


FIGURE C-16. Carrier modulation.

C.5.2 3G-ALE protocol definition.

3G ALE shall be implemented as defined in the following paragraphs.

C.5.2.1 3G-ALE service primitives.

Table C-XXI defines the service primitives exchanged between the CM process and the ALE process. Note that there is no requirement that implementations of 3G ALE contain precisely these service primitives; nor are the service primitives defined below necessarily all of the service primitives that would be required in an implementation of this protocol.

TABLE C-XXI. 3G-ALE service primitives.

Name	Attribute	Values	Description
LE_Link_Req	Overview		LE_Link_Req: issued by ALE client process (usually connection manager) to request establishment of a link
	Parameters	destAddr	11-bit 3G address of the station to be called
		callType	one of INDIVIDUAL, UNICAST, MULTICAST, BROADCAST, OTHER_NET
		trafType	Identifies the type of traffic for which the link is requested; one of: Packet Data, Modem Circuit (for HF data modems only), Voice Circuit (for analog voice or non-HF modems), High-Quality Circuit
		pri	Priority of the traffic for which the link is requested; one of Highest, High, Routine, Low
		callChan	Channel number on which the call is to be placed.
		trafChan	Optional traffic channel number for one-way and broadcast calls.
	Originator	user process	
	Preconditions		none
LE_Link_Ind	Overview		LE_Link_Ind: issued by ALE process to indicate the establishment of link as responder
	Parameters	addr	11-bit 3G address of the station or multicast to which link has been established
		callType	Identifies the type of link that has been established; same values as above
		trafType	Identifies the type of traffic that the link will carry; same values as above
	Originator	ALE entity	
	Preconditions		none

TABLE C-XXI. 3G-ALE service primitives (continued).

Name	Attribute	Values	Description
LE_Link_Confirm	Overview		LE_Link_Confirm: issued by ALE process to indicate establishment of link as caller
	Parameters	addr	11-bit 3G address of the station or multicast to which link has been established
	Originator	ALE entity	
	Preconditions		link has been requested or established
LE_Status_Ind	Overview		LE_Status_Ind: issued by ALE process to indicate its current status
	Parameters	status	Current ALE status; one of: SCANNING, CALLING, LINKED
	Originator	ALE entity	
	Preconditions		none
LE_Link_Det_Ind	Overview		LE_Link_Det_Ind: issued by ALE process to report detection of the establishment or termination of a link between remote stations
	Parameters	status	One of LINKED, AVAILABLE
		trafChan	traffic channel to be used in link
		caller	11-bit 3G address of the calling station
		responder	11-bit 3G address of the responding station
	Originator	ALE entity	
	Preconditions		none
LE_Link_Fail_Ind	Overview		LE_Link_Fail_Ind: issued by ALE process to indicate the failure of a link
	Parameters	reason	Reason for link failure; one of: NO_RESPONSE, REJECTED, NO_TRAF_CHAN, LOW_QUALITY, CALL_CHAN_OCCUPIED
	Originator	ALE entity	
	Preconditions		link has been requested or established
LE_Disconnect_Req	Overview		LE_Disconnect_Req: issued by user process to request termination of link and return to scanning operation; also used to reject an incoming link
	Parameters	reason	One of ABORT, RELINK, UNLINK, STATION_NOT_AVAILABLE
	Originator	user process	
	Preconditions		link has been requested or established

TABLE C-XXI. 3G-ALE service primitives (continued).

Name	Attribute	Values	Description
LE_Disconnect_Ind	Overview		LE_Disconnect_ Indication: issued to user process to indicate that the local station has ended its participation in a link for a reason other than the user process having issued a LE_Disconnect_Req primitive
	Parameters	reason	reason for disconnection. Value is one of: · SIGN_OFF: indicates that the local station has left a link due to another station's having signed-off the link — the other station in a PTP link, or the last remaining station in a multicast or broadcast circuit of which the local station was master. · ABORT: the local station has left a link due to the link master station's having aborted the link. · EOM: the local station has left a packet link due to successful completion of the packet data transfer and the absence of any packet traffic pending in the reverse direction. · RELINK: the remote station has initiated a re-link operation in which the participating PUs will attempt to re-establish the link on a different channel, by sending a LE_TERM or ALM PDU to the local station with Reason = RELINK. Used only on point-to-point links. · TRAF_TIMEOUT: the local station has left a circuit link, due to the CLC's having detected no traffic on the circuit link over a time interval equal to its traffic timeout period. · UNLINK: the remote master station of the currently-established multicast circuit has requested that the circuit link be dropped after a final roll call ("unlink") is performed. An LE_Disconnect_Ind service primitive with reason = UNLINK requests that the user process respond with a LE_Disconnect_Resp service primitive indicating whether the local station has succeeded or failed in receiving the traffic delivered on the circuit link. · STATION_NOT_AVAILABLE: The local station has received a Terminate PDU with the reason code set to "station not available". Thus, the called station responded to the call, but dismissed the possibility of establishing a link at this time.
	Originator	ALE process	
	Preconditions		A link is currently established

TABLE C-XXI. 3G-ALE service primitives (continued).

Name	Attribute	Values	Description
LE_Disconnect_ Conf	Overview		LE_ Disconnect Confirm: issued to CM by ALE to acknowledge that a currently-active link is being dropped as a result of a LE_Disconnect_Req service primitive
	Parameters	reason	The reason for which the link is being dropped. Possible values and their meanings are the same as for the reason parameter of the LE_Disconnect_Req service primitive, as described above.
		Responses	list of responses to an optional unlink roll-call performed at the conclusion of a multicast circuit connection, in which each response is in the form of an ordered pair (indAddr, ackNak), where indAddr is the address of a PU whose roll call response was heard, and ackNak is the Reason field-value of the LE_TERM PDU sent as the PU's response: UNL_ACK or UNL_NAK. The responses parameter has a value only when a multicast circuit link has been concluded with an unlink roll call. The list of responses can be incomplete for either of two reasons · at a slave PU, the user process has requested that the PU remain in the circuit link only long enough to transmit its own roll call response, by setting the watch parameter of its LE_Disconnect_Resp primitive to FALSE. In this case, the reason parameter has the value UNLINK; and responses are not included in the list from those PUs whose roll call time slots fall after the local PU's time slot. · at either the master PU or a slave PU, the user process may have cut short the PU's participation in the roll call, by issuing a LE_Disconnect_Req service primitive while the roll call is in progress. In this case, the reason parameter-value will be ABORT at the master PU, or SIGN_OFF at a slave PU; the response list will include only responses received before the LE_Disconnect_Req was accepted. The responses parameter is shown in the state diagrams only where it is used: where a multicast circuit link is being dropped with a concluding unlink roll call operation
	Originator	ALE process	
	Preconditions		user process issued an LE_Disconnect_Req
LE_TOD_Req	Overview	LE TOD Request: issued to the ALE by the user process to request and receive Time Of Day (TOD), by means of an asynchronous call and TOD request PDU on a specified frequency. In response to this service request, ALE will issue several asynchronous LE_Request PDUs, followed by the LE_TOD_Req PDU, on the specified frequency. It will then wait for response, and either indicate successful TOD reception, or failure.	
	Parameters	destAddr	address of the PU to be called. Usually all 1's, indicating the reigning Net Controller
		pri	Priority of the request; one of Highest, High, Routine, Low
		Channel	Calling channel number as in LE_Connect_Req.
	Originator	CM	
	Preconditions	Station is not already linked, and is available for linking.	
LE_TOD_Ind	Overview	ALE TOD Indication: issued by the ALE to the user process to indicate that it has received the TOD from another PU.	
	Parameters	TOD	Time quality set to all 1's indicates failure
	Originator	ALE	
	Preconditions	CM issued an LE_TOD_Req	
LE_TODReq_Ind	Overview	Indicates that the PU received a TOD request PDU.	
	Parameters	none	

TABLE C-XXI. 3G-ALE service primitives (continued).

Name	Attribute	Values	Description
LE_McastUpdate	Overview		LE_McastUpdate: issued by user process to add or delete a dwell group from a multicast
	Parameters	multicast	affected multicast address (this 6-bit address is used as member number in calls to all dwell groups)
		status	one of: INCLUDED, EXCLUDED (for this station)
	Originator	user process	
	Preconditions		none

Table C-XXII defines the default meanings of the trafType parameter in the 3G-ALE primitives.

TABLE C-XXII. 3G Traffic Type Codes.

Traffic Type Code	Default meaning
0	NO TRAFFIC TO SEND
1	ANLG_VOICE
2	DGTL_VOICE
3	ANDVT [STANAG 4197, STANAG 4198] (parameters not specified due to auto-detect capability)
4	STANAG 4285 [2400, long intlv]
5	STANAG 4285 [2400, short intlv]
6	STANAG 4285 [1200, long intlv]
7	STANAG 4285 [1200, short intlv]
8	STANAG 4285 [600, long intlv]
9	STANAG 4285 [600, short intlv]
10	STANAG 4285 [300, long intlv]
11	STANAG 4285 [300, short intlv]
12	STANAG 4285 [150, long intlv]
13	STANAG 4285 [150, short intlv]
14	STANAG 4285 [75, long intlv]
15	STANAG 4285 [75, short intlv]
16	STANAG 4415 (autodetect of long/short intlvr)
17	STANAG 4539_HDR (parameters not specified due to auto-detect capability)
18	SER_110B (parameters not specified due to auto-detect capability)
19	HDL_24
20	HDL_12
21	HDL_6
22	HDL_3
23	LDL_32
24	LDL_64
25	LDL_96
26	LDL_128
27	LDL_160
28	LDL_192
29	LDL_224
30	LDL_256
31	LDL_288
32	LDL_320
33	LDL_352
34	LDL_384
35	LDL_416
36	LDL_448
37	LDL_480
38	LDL_512
39 … 50	Reserved for future STANAG 4538 use
51 … 62	Reserved for vendor use (non-interoperable)
63	TOD

C.5.2.2 3G-ALE PDUs.

The link establishment protocol data units (LE-PDUs) are shown in figure C-17. Unused encodings are reserved, and shall not be used until standardized. Order of transmission shall be as specified in C.4.10 Order of transmission. For example, the LE_Broadcast PDU shall begin 0, 1, 1, 1 , 0.

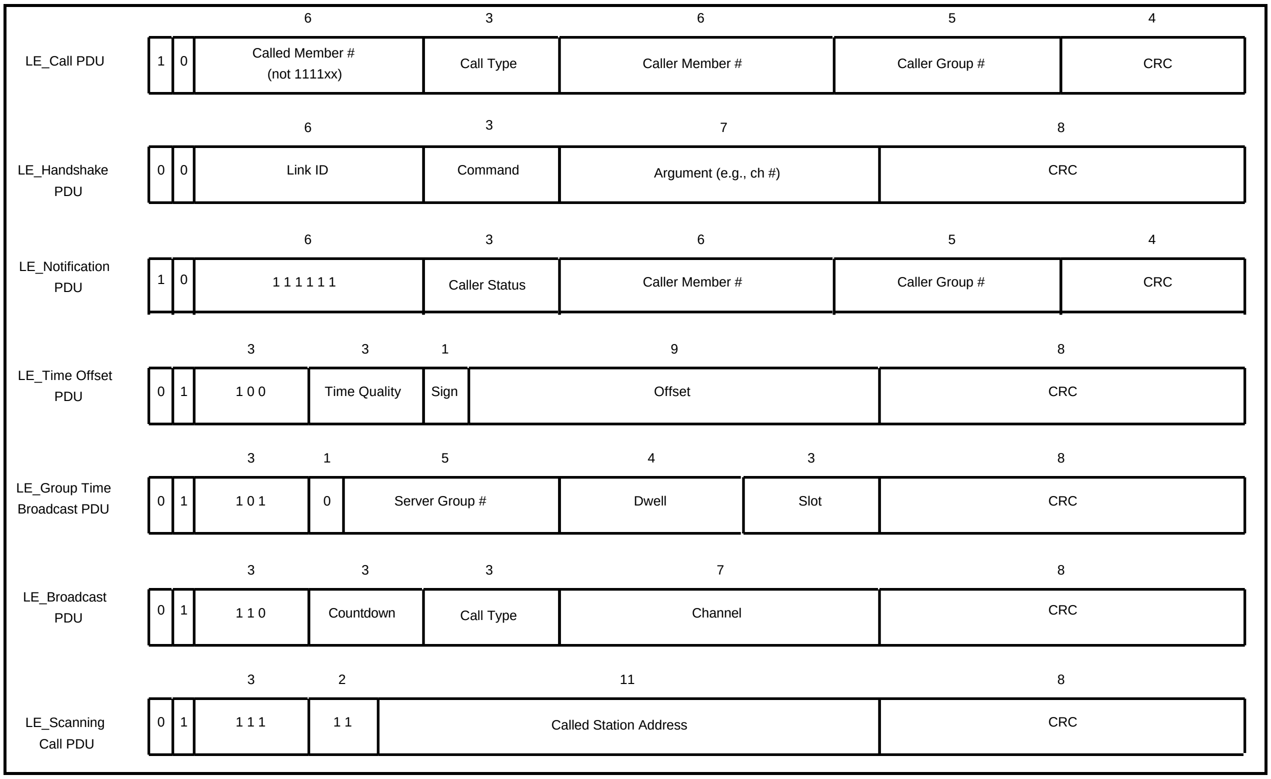


FIGURE C-17. 3G-ALE PDUs.

C.5.2.2.1 LE_Call PDU.

The LE_Call PDU shall be formatted as shown in figure C-17. It conveys necessary information to the responder so that the responder will know whether to respond, and what quality of traffic channel will be needed.

The Call Type field in the LE_Call PDU shall be encoded as specified in table C-XXIII.

- A call type of Packet Data shall be used only when the HDL or LDL protocol will be used to deliver a message after link establishment.
- The call type shall be Modem Circuit when an HF data modem using waveforms other than BW0-BW5 will be used to convey traffic after link establishment.
- The Voice Circuit call type requests a minimum link SNR suitable for orderwire voice operation (for example, 10 dB or better).

- The High-Quality Circuit call type requests a substantially better SNR than an orderwire Voice Circuit (for example, 20 dB or better).
- The unicast and multicast call types are used when the calling station will specify the traffic channel used for a link.
- The link release call type shall be used only when releasing, rather than establishing, a link.

TABLE C-XXIII. Call type field encodings.

Code	Call Type	Second PDU From
0	3G ARQ Packet Data	Responder
1	HF Modem Circuit	Responder
2	Analog Voice Circuit	Responder
3	High-Quality Circuit	Responder
4	Unicast ARQ Packet	Caller
5	Unicast Circuit	Caller
6	Multicast Circuit	Caller
7	Control	Caller

C.5.2.2.2 LE_Handshake PDU.

The LE_Handshake PDU shall be formatted as shown in figure C-17. The link ID shall be computed as follows from the 11-bit addresses of the caller (node sending LE_Call PDU) and responder (node or multicast addressed in LE_Call PDU):

1. temp1 = <caller address> * 0x13C6EF
2. temp2 = <responder address> * 0x13C6EF
3. LinkID = ((temp1 >> 4) + (temp2 >> 15)) & 0x3f

where '*' indicates 32-bit unsigned multiplication, '>> n' indicates right shift by n bits, and '&' indicates bitwise AND. Example LinkID computations are shown below.

Caller	Responder	temp1	temp2	result	result
1	2	0013c6ef	00278dde	3D	61
1	3	0013c6ef	003b54cd	24	36
2	1	00278dde	0013c6ef	4	4
3	1	003b54cd	0013c6ef	33	51
1951	1	96b91771	0013c6ef	1E	30
(decimal)	(decimal)	(hexadecimal)	(hexadecimal)	(hexadecimal)	(decimal)

The Command field shall be encoded as shown in Table C-XXIV. Unused encodings are reserved, and shall not be used until standardized.

TABLE C-XXIV. Command field encodings.

Code	Command	Description	Argument
0	Continue Handshake	The handshake will continue for at least another two-way handshake (on the next assigned called station dwell frequency when operating in synchronous mode).	Reason
1	Commence Traffic Setup	This is the final command sent on a calling channel. The argument is the channel number on which the responding station will (or should) listen for traffic setup. Following this command, all stations proceed to that traffic channel.	Channel
2	Voice Traffic	This command directs called station(s) to tune to a traffic channel and commence voice traffic. The argument is the channel number. Following this command, the calling station will be first to speak. (Uni- and multicast only)	Channel
3	Link Release	This command informs all listening stations that the specified traffic channel is no longer in use by the sending station.	Channel
4	Sync Check	This command directs the called station to measure and report synchronization offset back to the calling station. Used in synchronization management protocol (C.5.2.7).	Quality | Slot
5	(reserved)		
6	(reserved)		
7	Abort Handshake	This command immediately terminates the handshake and needs no response. It is analogous to the TWAS preamble in 2G ALE.	Reason

The Argument field shall contain a channel number, a reason code, or 7 bits of data, as indicated in table C-XXIV. Reasons shall be encoded as 7-bit integers with values selected from table C-XXV. Unused encodings are reserved, and shall not be used until standardized.

TABLE C-XXV. Reason field encodings.

Code	Reason	Remarks
0	NO_RESPONSE	Required resources not responding
1	REJECTED	Station is unwilling to link
2	NO_TRAF_CHAN	All traffic channels are in use
3	LOW_QUALITY	Available traffic channels are of insufficient quality to support requested traffic
others	(reserved)	(reserved)

C.5.2.2.3 LE_Notification PDU.

The LE_Notification PDU shall be formatted as shown in figure C-17, and shall be used as specified in C.5.2.5 Notification Protocol Behavior. The Caller Member Number and Caller Group Number fields shall contain the address of the station sending the PDU. The Caller Status field shall be encoded as shown in Table C-XXVI. Unused encodings are reserved, and shall not be used until standardized.

TABLE C-XXVI. Caller status field encodings.

Code	Station Status
0	Nominal
1	Time server
6	Commencing EMCON
7	Leaving network
others	(reserved)

C.5.2.2.4 LE_Broadcast PDU.

The LE_Broadcast PDU shall be formatted as shown in figure C-17, and shall be used as specified in C.5.2.4.4.5 3G-ALE synchronous mode broadcast calling.

The Call Type field shall describe the traffic to be sent:

- Analog Voice Circuit if the receiving stations are to deliver the received audio directly.
- HF Modem Circuit if an HF modem is to be engaged to process received traffic.
- High-Quality Circuit if a non-HF modem is to be engaged to process received traffic.
- 3G ARQ Packet Data if the link will be used in bidirectional mode using the CLC (see C.5.6) for channel access control.

The Channel field shall specify the channel that will be used for traffic.

The Countdown field shall be used as specified in C.5.2.4.4.5 3G-ALE synchronous mode broadcast calling and in C.5.2.4.5.6 3G-ALE asynchronous mode broadcast call.

C.5.2.2.5 Scanning call PDU.

The LE_Scanning_Call PDU shall be formatted as shown in figure C-17, and shall be used for asynchronous mode scanning calls as specified in C.5.2.4.2.2. The complete address of the called station appears in this PDU.

C.5.2.2.6 CRC computation for 3G-ALE PDUs.

Each LE_PDU contains either a 4-bit or an 8-bit CRC. The 4-bit CRC shall be computed in accordance with C.4.12 using the polynomial $x^4 + x^3 + x + 1$. The 8-bit CRC shall be computed in accordance with C.4.12 using the polynomial $x^8 + x^7 + x^4 + x^3 + x + 1$.

C.5.2.3 Synchronous dwell structure.

When scanning in synchronous mode, 3G systems shall dwell on each assigned channel for 5.4 seconds. Each synchronous dwell time is divided into six slots of 900 ms each, which shall be used as follows (see figure C-18).

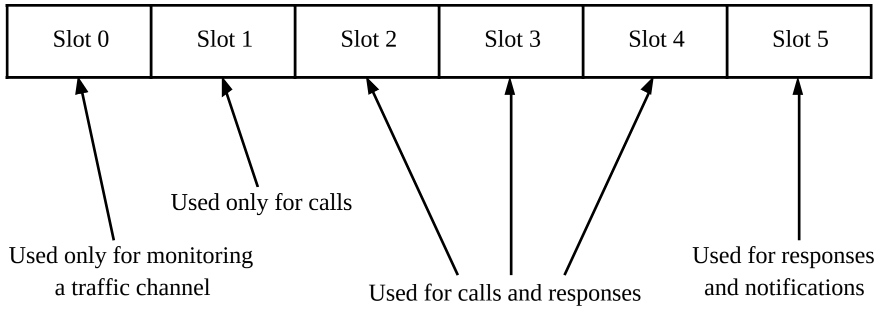

Slot 0
Slot 1
Slot 2
Slot 3
Slot 4
Slot 5
Used only for calls
Used only for monitoring a traffic channel
Used for calls and responses
Used for responses and notifications

FIGURE C-18. Synchronous dwell structure.

Slot 0: Tune and Listen Time. During Slot 0, radio frequency (RF) components shall be retuned to the frequency on which the node may transmit during that dwell.

- A scanning station shall tune to the assigned calling channel for that dwell, computed in accordance with C.4.4.1. Couplers are normally not retuned while scanning.
- A station that will place a call during that dwell shall instead tune to the channel on which it will call during that dwell. The coupler may be retuned either in slot 0 or immediately before transmitting.

For the remainder of slot 0, every receiver shall sample a traffic frequency in the vicinity of the new calling channel, attempting to detect traffic. (This provides recent traffic channel status before stations get involved in a handshake.)

Calling Slots. The remainder of the dwell time is divided into five 900 ms calling slots. These slots shall be used for the synchronous exchange of PDUs on calling channels. A two-way handshake shall not begin in the last slot of a dwell. The last slot of every dwell is reserved for LE_Handshake, LE_Notification, and LE_Broadcast PDUs.

C.5.2.4 3G-ALE protocol behavior.

The behavior of the 3G-ALE protocols is specified first in narrative form, then in the following paragraphs in terms of data structures, states, events, actions, and state transitions.

C.5.2.4.1 3G-ALE synchronous mode protocol.

The synchronous-mode link establishment protocol shall comply with the following requirements as observed over the air. Precise definitions of the protocols are presented following overviews of the individual, multicast, and broadcast calling protocols.

C.5.2.4.1.1 3G-ALE synchronous mode slot selection.

Slot selection for sending an LE_Call PDU shall be randomized over all usable slots, but the probabilities for higher-priority calls shall be skewed toward the early slots while lower-priority calls are skewed toward the later slots. (Such a scheme will operate reasonably well in all situations, while hard partitioning of early slots for high and late slots for low priorities would exhibit inordinate congestion in crisis and/or routine times.) Suggested sets of probabilities are shown in Table C-XXVII for LE_Call PDUs.

TABLE C-XXVII. Probability of slot selection for LE_Call PDUs.

Probability of Slot Selection for LE Broadcast and LE Notification PDUs.				
Call Priority	**Slot 1**	**Slot 2**	**Slot 3**	**Slot 4**
Highest	50%	30%	15%	5%
High	30%	50%	15%	5%
Routine	5%	15%	50%	30%
Low	5%	15%	30%	50%

A new random slot shall be selected for each dwell in which a call will be placed. Random number generation for slot selection shall be essentially independent from one dwell to the next, and among different stations, so that systems that select the same slot in one dwell will not have a higher probability of continuing to select identical slots in subsequent dwells than is indicated in Table C-XXVII.

C.5.2.4.1.2 3G-ALE synchronous mode individual calling overview.

The one-to-one linking protocol identifies a frequency for traffic use relatively quickly (i.e., within a few seconds), and minimizes channel occupancy during this link establishment process. When the ALE process receives an LE_Link_Req primitive from its client process (normally the CM process) during Slot 0 at the beginning of a dwell (or during the final slot of the preceding dwell), it shall do the following:

- select a slot in accordance with C.5.2.4.1.1 3G-ALE synchronous mode slot selection;
- listen on an associated traffic channel (if any) during Slot 0 of the calling dwell;
- if not calling in Slot 1, listen for occupancy on the calling channel during the slot immediately preceding its calling slot;

- defer its call if it believes the channel will be occupied by a response PDU during the selected calling slot; otherwise
- send its LE_Call PDU.

If an LE_Link_Req is received during Slot 1 through the next-to-last slot of a dwell, the ALE process shall do the following:

- select a slot in accordance with C.5.2.4.1.1 3G-ALE synchronous mode slot selection, but add the selected slot number to the current slot number to obtain the prospective calling slot; if the result exceeds the range of valid calling slots, defer the call until the next dwell; otherwise,
- listen for occupancy on the calling channel during the slot immediately preceding its calling slot;
- defer its call if it believes the channel will be occupied by a response PDU during the selected calling slot; otherwise
- send its LE_Call PDU.

A station that receives an LE_Call PDU addressed to its node address shall respond in the immediately following slot with an LE_Handshake PDU that either aborts the call, names a traffic channel, or defers naming a channel but continues the handshake. When a suitable frequency for traffic to the responding station has been found, the stations shall enter the Traffic state. If additional negotiation is required (e.g., to set up a full duplex circuit using a second frequency), the ALM protocol shall be employed on the traffic channel.

C.5.2.4.1.3 <u>3G-ALE synchronous mode unicast calling.</u>

A unicast call is used to contact an individual station and direct it to a traffic channel selected by the calling station.

1. An LE_Call PDU shall be sent as usual, containing the individual responding-station address. The Call Type field shall be Unicast. No station shall respond to a Unicast-type LE_Call PDU.

2. The caller shall send an LE_Handshake PDU in the immediately following response slot that directs the called station to a traffic channel.

3. The called station shall tune to that channel and listen for modem traffic if the command in the LE_Handshake PDU is Commence Traffic Setup. If the command is Voice Traffic, the called station shall tune to the channel and prepare for voice traffic (e.g., unmute the speaker). If the announced traffic does not begin to arrive within the traffic wait timeout, the called station shall return to scan.

4. After sending the LE_Handshake PDU, the caller shall tune to the specified channel and initiate the type of traffic indicated in the LE_Handshake PDU.

Note that a unicast call may be used to set up a link for bidirectional traffic; it is only the link establishment “handshake” that is unidirectional.

C.5.2.4.1.4 3G-ALE synchronous mode multicast calling.

A multicast call is used to contact selected stations concurrently and direct them to a traffic channel selected by the calling station.

1. An LE_Call PDU shall be sent as usual, but it shall contain a multicast responding-station address. The Call Type field shall be Multicast. No station shall respond to a Multicast-type LE_Call PDU.

2. The caller shall send an LE_Handshake PDU in the immediately following response slot that directs the called stations to a traffic channel. The link ID field shall be computed in accordance with C.4.5.3Multicast addresses.

3. The called stations shall tune to that channel and listen for modem traffic if the command in the LE_Handshake PDU is Commence Traffic Setup. If the command is Voice Traffic, the called stations shall tune to the channel and prepare for voice traffic (e.g., unmute the speaker). If the announced traffic does not begin to arrive within the multicast traffic wait timeout, the called stations shall return to scan. (This timeout should be set to accommodate calls to the maximum number of dwell groups whose members may subscribe to the multicast.)

4. When the stations subscribing to a multicast are assigned to more than one dwell group, the multicast call (both PDUs) shall be sent repeatedly by the caller. The caller should select the timing (and possible redundancy) of its transmissions to minimize calling channel occupancy while maximizing the probability of reaching called stations.

5. After sending the (final) LE_Handshake PDU, the caller shall tune to the specified channel and initiate the type of traffic indicated in the LE_Handshake PDU.

C.5.2.4.1.5 3G-ALE synchronous mode broadcast calling - not tested (NT)

An LE_Broadcast PDU directs every station that receives it to a particular traffic channel, where another protocol (possibly voice) will be used. A means shall be provided for operators to disable execution of the broadcast protocol.

- The Call Type field in the LE_Broadcast PDU shall be encoded as in the LE_Call PDU, except that only the circuit call types may be used.

- The Countdown field shall indicate of the number of dwells that will occur between the end of the current dwell and the start of the broadcast. A Countdown value of 0 shall indicate that the broadcast will begin in Slot 1 of the following dwell. Other Countdown field values n _ 0 indicate that the broadcast will begin no later than 4n+3 dwell times in the future.

- The Channel field shall indicate the channel that will carry the broadcast.

A station may send an LE_Broadcast PDU in every slot in a dwell (except Slot 0). It may also change frequency every slot to reach a new dwell group. The calling station should check occupancy on a calling channel before transmitting on that channel. However, a mechanism shall be provided to override this occupancy check for Highest priority broadcasts so that an LE_Broadcast PDU may be sent on a new channel in every slot. (A split-site station with a fast tuner may be able to send any priority LE_Broadcast PDU on a new channel in every slot by listening on the next channel each time and tuning at the start of the slot.)

Stations that receive an LE_Broadcast PDU and tune to the indicated traffic channel shall return to scan if the announced traffic does not begin within the traffic wait timeout period after the announced starting time of the broadcast.

At the beginning of slot 1 in the dwell after the caller sent LE_Broadcast PDU(s) with the countdown field set to 0, the caller shall commence TM on the indicated traffic channel.

C.5.2.4.1.6 3G-ALE synchronous mode link release.

At the conclusion of an individual or unicast link, the caller shall send a link release. The link release shall be an LE_Call PDU containing the original called station address, with a type of Control, followed by an LE_Handshake PDU that identifies the traffic channel and contains a link release command.

The link release shall be sent on the calling channel on which the handshake that set up the link occurred. The calling station shall attempt to send the link release during the first dwell after the link is terminated during which the called dwell group is listening on that calling channel. If calling channel occupancy during that dwell prevents transmission of the link release, the calling station need not attempt to send a link release later.

C.5.2.4.1.7 3G-ALE synchronous mode protocol examples.

An example of synchronous mode 3G-ALE protocol behavior is shown in figure C-19. The first call occurs in Slot 3. The responder receives the call, but has not identified a suitable traffic channel for the requested traffic, and therefore sends an LE_Handshake PDU containing a Continue Handshake command.

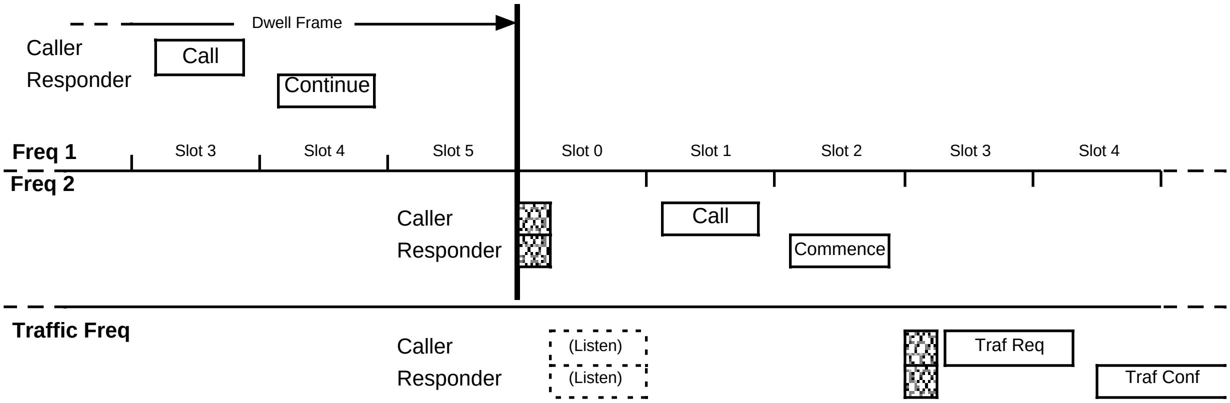


FIGURE C-19. Example 3G-ALE synchronous link establishment.

In the next dwell, both stations tune during Slot 0, then listen for occupancy on a nearby traffic frequency. The caller selects Slot 1 this time, and the responder determined that the traffic channel was available. When the LE_Call PDU is received by the responder, the measured channel quality is sufficient for the offered traffic, and the responder sends an LE_Handshake PDU containing a Commence Traffic Setup command that indicates the traffic channel to be used. Both stations tune to that channel in the following slot, and the caller initiates the traffic setup protocol.

C.5.2.4.1.8 3G-ALE synchronous mode timing characteristics.

Synchronous-mode 3G-ALE timing is specified in terms of T_{sym}, where 2400 T_{sym} = 1 s. The time at which each type of 3G-ALE PDU shall be sent is specified in the following paragraphs. In each case of sending a PDU, the transmitter shall have reached 90 percent of steady-state power at the time that the PDU begins. Unless otherwise stated, deviation from specified timing shall not exceed ±10 percent.

C.5.2.4.1.8.1 3G-ALE synchronous mode tuning time.

All emanations for tuning the RF components in a synchronous-mode 3G-ALE system shall occur only during the first 256 T_{sym} (approximately 106.7 ms) of Slot 0, or between the start of a calling slot and the beginning of a PDU sent by that station in that slot. (Emanations required for the initial or learning tuning of a coupler with presets may occur at any time.)

C.5.2.4.1.8.2 3G-ALE synchronous mode traffic channel evaluation timing.

Synchronous-mode 3G-ALE systems shall listen for occupancy of a traffic channel during most of the remainder of Slot 0 during each scanning dwell , and shall meet the requirements of C.4.6.1.2 Occupancy detection. The receiver shall be re-tuned to the calling channel in time to receive a PDU that begins coincident with the start of Slot 1. (NOTE: a PDU may arrive this early due to differences in local time bases.)

C.5.2.4.1.8.3 3G-ALE synchronous mode call, broadcast, and notification timing.

LE_Call, LE_Broadcast, and LE_Notification PDUs shall be sent during slots selected as described previously. The PDU shall begin at the later of the following two instants:

1. 256 T_{sym} (approximately 106.7 ms) has elapsed since the start of the selected slot.
2. If and only if a PDU was received in the preceding slot, 512 T_{sym} (approximately 213.3 ms) has elapsed since the end of that PDU.

C.5.2.4.1.8.4 3G-ALE synchronous mode response timing

A responding station shall commence transmission of an LE_Handshake PDU at the later of the two following instants:

1. 256 Tsym (approximately 106.7 ms) has elapsed since the start of the response slot at the responding station.
2. 512 Tsym (approximately 213.3 ms) has elapsed since the end of the received LE_Call PDU.

C.5.2.4.1.8.5 3G-ALE synchronous mode unicast, multicast, and link release command timing

When a 3G-ALE system is sending a unicast or multicast call, or a link release, the LE_Handshake PDU that designates the traffic channel shall be sent in the slot that immediately follows the LE_Call PDU. The transmitter shall be keyed when 256 T_{sym} (approximately 106.7 ms) has elapsed since the start of that slot.

C.5.2.4.2 3G-ALE asynchronous mode protocol.

When a 3G-ALE network is operating in asynchronous mode, calls shall be extended to capture scanning receivers (similar to 2G-ALE) as described in C.5.2.4.2.2 3G-ALE asynchronous mode scanning call. The remainder of the handshake shall be self-timed as described in C.5.2.4.2.3 3G-ALE asynchronous mode handshake.

C.5.2.4.2.1 3G-ALE asynchronous mode listen before transmit.

Systems operating in 3G-ALE asynchronous mode shall listen on the calling channel before sending a scanning call or a sound. The duration of this listen before transmit period shall be programmable, with a default value of 2 s.

C.5.2.4.2.2 3G-ALE asynchronous mode scanning call.

The LE_Scanning_Call PDU shall be sent repeatedly to capture scanning receivers, followed by an LE_Call PDU. The number of times the LE_Scanning_Call PDU is sent shall be a programmable multiple M of the number of channels scanned (denoted C). By default, the multiplier M shall be 1.3. The number of LE_Scanning_Call PDUs sent shall be the smallest integer that is greater than or equal to the product of C and M.

During a scanning call, only the first LE_Scanning_Call PDU shall include T_{tlc} (used for transmitter level control and receiver AGC settling). All succeeding LE_Scanning_Call PDUs and the LE_Call PDU shall omit T_{tlc}, and include only the BW0 preamble (T_{pre}) and data (T_{data}) portions.

C.5.2.4.2.3 3G-ALE asynchronous mode handshake.

The asynchronous mode 3G-ALE handshake is self-timed. The responding station shall

1. Decode an LE_Call PDU when it is received,

2. Tune its RF components (if necessary),

3. Listen on a traffic channel for approximately 800 ms to determine occupancy, and

4. Key its transmitter for its response, in accordance with C.5.2.4.2.10.2 3G-ALE asynchronous mode handshake timing.

The use of LE_Handshake PDU commands shall be the same as in the synchronous mode.

If the calling station receives a Commence Traffic Setup command in the responding LE_Handshake PDU, it shall commence the data link protocol (or ALM protocol, if required) starting 4 T_{slot} = 3.6 s after the beginning of its LE_Call PDU.

C.5.2.4.2.4 3G-ALE asynchronous mode unicast call.

A unicast call is used to contact an individual station and direct it to a traffic channel selected by the calling station. An asynchronous-mode unicast call shall consist of a scanning call in accordance with C.5.2.4.2.2 3G-ALE asynchronous mode scanning call, with a Call Type of Unicast ARQ Packet, Unicast Circuit, or Control in the LE_Call PDU, followed immediately by an LE_Handshake PDU that contains the following:

- A link ID (C.5.2.2.2) computed from the called station address and the calling station address.
- A Voice Traffic command if requesting a link for analog voice traffic, or a Commence Traffic Setup command if requesting a link for other traffic types.
- The channel number for the traffic channel that will be used for traffic.

This LE_Handshake PDU shall not include T_{tlc}, but only the BW0 preamble (T_{pre}) and data (T_{data}) portions.

C.5.2.4.2.5 3G-ALE asynchronous mode multicast call.

A multicast call is used to contact selected stations concurrently and direct them to a traffic channel selected by the calling station. An asynchronous-mode multicast call shall consist of a scanning call in accordance with C.5.2.4.2.2 3G-ALE asynchronous mode scanning call, with a Call Type of Multicast in the LE_Call PDU, followed immediately by an LE_Handshake PDU that contains the following:

- A link ID (C.5.2.2.2) computed from the multicast address and the calling station address, in accordance with C.4.5.3 Multicast addresses.
- A Voice Traffic command if the link is for analog voice, otherwise a Commence Traffic Setup command.
- The channel number for the traffic channel that will be used for the multicast.

This LE_Handshake PDU shall not include T_{tlc}, but only the BW0 preamble (T_{pre}) and data (T_{data}) portions.

C.5.2.4.2.6 3G-ALE asynchronous mode broadcast call.

The asynchronous-mode broadcast call shall consist of at least M C + 1 repetitions of an LE_Broadcast PDU, where C is the number of calling channels scanned by the stations being called, and M is the multiplier defined in C.5.2.4.2.2 3G-ALE asynchronous mode scanning call. The Call Type and Channel fields shall be used as specified in C.5.2.4.1.5 3G-ALE synchronous mode broadcast calling. The Countdown field shall be used as follows:

1. A repetition factor R shall be computed as the smallest integer that is greater than or equal to M C / 7. For example, if M = 1.2 and C = 10, M C / 7 = 1.7, so R shall be 2.

2. The initial value of the Countdown field shall be the smallest integer that is greater than or equal to M C / R. For example if M = 1.2 and C = 10, the initial Countdown value shall be 6.

3. R identical repetitions of the LE_Broadcast PDU shall be sent, after which the Countdown field shall be decremented.

4. Step 3 shall be repeated until the decremented value of the Countdown field is 0. A single instance of the LE_Broadcast PDU with Countdown = 0 shall be sent.

5. The broadcast shall commence on the indicated channel 2 T_{slot} after the end of the final LE_Broadcast PDU.

During an asynchronous-mode broadcast call, only the first LE_Broadcast PDU shall include T_{tlc} (used for transmitter level control and receiver AGC settling). All succeeding LE_Broadcast PDUs shall omit T_{tlc}, and include only the BW0 preamble (T_{pre}) and data (T_{data}) portions.

A means shall be provided for operators to disable execution of the asynchronous-mode broadcast protocol.

C.5.2.4.2.7 3G-ALE asynchronous mode link release.

Transmission of link releases is optional when operating in asynchronous mode. When used, an asynchronous-mode link release shall be sent after termination of a link on the calling channel that was used in establishing the link. Asynchronous-mode link releases shall begin with a scanning call addressed to the called station in accordance with C.5.2.4.2.2 3G-ALE asynchronous mode scanning call, with a Call Type of Link Release in the LE_Call PDU, followed immediately by an LE_Handshake PDU that contains the following:

- A link ID computed from the called address and the calling station address.
- A Link Release command.
- The channel number for the traffic channel that is being released.

This LE_Handshake PDU shall not include T_{tlc}, but only the BW0 preamble (T_{pre}) and data (T_{data}) portions.

C.5.2.4.2.8 3G-ALE asynchronous mode entry to synchronous networks.

Stations not synchronized to network time may link with synchronous mode stations by sending either normal (C.5.2.4.2.2) or extended scanning calls addressed to those stations. The duration of an extended scanning call is 4 C seconds, which ensures that the destination station will dwell on the calling channel during the call.

C.5.2.4.2.9 3G-ALE asynchronous mode protocol example.

The asynchronous mode 3G-ALE protocol is illustrated in figure C-20. The called station ("Responder") receives a scanning call and computes a snapshot of the quality of the calling channel by evaluating the received LE_Scanning_Call and LE_Call PDUs. Having determined that the channel quality is sufficient for the type of traffic announced in the LE_Call PDU, the Responder tunes on the calling channel for sending a response, listens on a nearby traffic channel, finds the traffic channel unoccupied, and therefore sends a Commence Traffic Setup command in an LE_Handshake PDU.

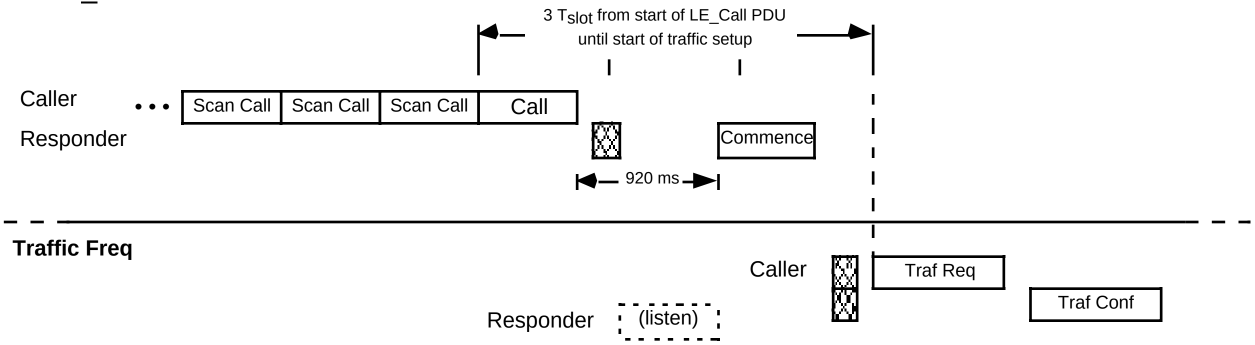


FIGURE C-20. 3G-ALE asynchronous mode link establishment.

C.5.2.4.2.10 3G-ALE asynchronous mode timing characteristics.

Asynchronous mode timings are referenced only to the start of the scanning call, not to any global timing system. Unless otherwise stated, deviation from specified timing shall not exceed ±10 percent.

C.5.2.4.2.10.1 3G-ALE asynchronous mode scanning call timing.

The duration of a 3G-ALE asynchronous-mode scanning call (including LE_Scanning_Call PDUs and the LE_Call PDU) shall be as follows, where C is the number of scanned channels, M is the multiplier of C.5.2.4.2.2 3G-ALE asynchronous mode scanning call, and the ceiling function returns the smallest integer greater than or equal to its argument.

$$\begin{aligned} T_{sc} &= T_{tlc} + \text{ceiling}(M\,C + 1)\,(T_{BW0\ pre} + T_{BW0\ data}) \\ &= 2.640\text{ s} \quad \text{when } C = 3 \text{ calling channels, and } M = 1.3 \end{aligned}$$

When an LE_Handshake PDU is appended to the LE_Call PDU, T_{sc} is increased by $T_{BW0\ pre} + T_{BW0\ data}$ or approximately 506.7 ms.

C.5.2.4.2.10.2 3G-ALE asynchronous mode handshake timing.

When an LE_Handshake PDU is sent by a responding station, it shall begin 32 T_{sym} (approximately 13.3 ms) after the transmitter is keyed. Total elapsed time from the end of the LE_Call PDU until the start of the LE_Handshake PDU shall be 2208 T_{sym} (920 ms), measured at the responding station.

The duration of a 3G-ALE asynchronous-mode handshake, from the start of the scanning call through the start of traffic setup on the traffic channel is as follows:

$$T_{handshake} = T_{tlc} + \text{ceiling}(M\ C)\ (T_{BW0\ pre} + T_{BW0\ data}) + 3\ T_{slot}$$
$$= 4.83\text{ s when } C = 3 \text{ calling channels, and } M = 1.3$$

C.5.2.4.3 3-G ALE Formal Protocol Description.

This section provides a more formal definition of the 3G-ALE protocols specified in narrative form in the preceding sections. Note that the specific data structures, states, events, actions, and state transitions mentioned here are not requirements of a compliant implementation, but only serve to illustrate the required over-the-air behavior of compliant systems. The data structures, events, and actions are listed in a single set of tables, which are used by both the synchronous-mode and asynchronous-mode protocol definitions. Separate behavior tables are specified for the two modes.

C.5.2.4.3.1 3G-ALE protocol data.

The internal variables used in the description of the 3G-ALE protocol are defined in table C-XXVI. These are for illustrative use only, and are not mandatory in implementations of 3G ALE except as required elsewhere.

TABLE C-XXVIII. 3G-ALE protocol data.

Data item	Description
myIndivAddr	11-bit address of this station
myMulticastAddresses	list of 11-bit addresses for multicasts to which this station subscribes
networkTime	coordinated network time (usually synchronized to GPS time)
myCurrentDwellChannel	calling channel on which this station listens for calls
myCurrentTrafficChannel	traffic channel on which this station monitors occupancy
channelOccupancy	array of channel occupancy records, including result, time measured
callingChannel	current dwell channel of destination station
destStation	ID of destination station (indiv, mcast, or bcast)
linkID	Link ID value computed for current handshake
linkQualityTable	array of link quality records for all stations and channels
prefTrafChan	preferred traffic channel for current handshake partner
myCallingSlot	slot in which call will be sent
bcastCount	LE_Broadcast PDU countdown (use varies for sync and async modes)
announcedBroadcastChannel	channel specified in LE_Broadcast PDU
numScanChan	number of calling channels in scan list
scanCallCountdown	number of times LE_Scanning_Call PDU is sent
scanSoundCountdown	number of times LE_Notification PDU is sent when sounding
trafWaitTime	time called station will wait for traffic to start after link is established
trafWaitTimeMcast	time called station will wait for traffic to start after link is established (longer time allowed for multicasts)

C.5.2.4.3.2 3G-ALE protocol events.

Table C-XXIX defines the events to which the 3G-ALE entity responds. The event names are used in the state transition tables in C.5.2.4.3.4 and C.5.2.4.3.5 which define the behavior of the 3G-ALE protocol.

TABLE C-XXIX. 3G-ALE protocol events.

Event name	Description
End of dwell	A boundary between dwells has occurred
TuningComplete	Tuning has been completed in all RF components
FinishedListening	The occupancy check of a channel has been completed
D: LE_Link_Req(destAddr, callType, pri, [chan])	An LE_Link_Req primitive was received from the user process (Connection Manager); chan is optional
D: LE_ReturnToScan	An LE_ReturnToScan primitive was received from the user process (Connection Manager)
R: LE_Call(destAddr, srcAddr, callType)	received an LE_Call PDU
R: LE_ScanCall(destAddr)	received an LE_Scanning_Call PDU for indicated destination address
R: LE_Hshake(ID, CMD, ARG)	received an LE_Handshake PDU
R: LE_Bcast(countdown, callType, chan)	received an LE_Broadcast PDU
FinishedSendingPDU	occurs at end of slot (synchronous mode) or end of PDU (asynchronous mode)
SlotAvailable	Occupancy check of preceding slot(s) and analysis of any received PDUs indicates that no handshake in progress will extend into the slot now beginning
SlotOccupied	Occupancy check of preceding slot(s) and analysis of any received PDUs indicates that a handshake in progress will extend into the slot now beginning
ResponseTimeout	No response arrived within the timeout previously set
ScanCallTimeout	End of scanning call did not occur within allowed timeout
ScanCallDropout	Unable to identify BW0 preamble for three consecutive PDUs during scanning call timeout period
TrafWaitTimout	Traffic did not begin within the timeout previously set
TimeToSound(channel)	Time to sound on indicated channel

C.5.2.4.3.3 3G-ALE protocol actions.

Table C-XXX defines the actions which the 3G-ALE entity can perform. The action name is used in the state transition tables used below to define the behavior of the 3G-ALE protocol.

TABLE C-XXX. 3G-ALE protocol actions.

Action name	Description
InitBroadcastCount	Initializes broadcastCount to number of times LE_Broadcast PDU will be sent
InitBcastCountdown(number)	Sets broadcastCount to number
TuneToNewChannel(chan)	Retune transceiver, PA, coupler, etc to specified channel; TuningComplete event when done
SelectTrafficChannel(chan)	Selects a traffic channel "near" specified channel, considering age of channel measurements
ListenOnChannel(chan)	Listen for occupancy on specified channel; FinishedListening event after preset interval
RecordOccupancy(chan)	store results of listening on chan in channelOccupancy array
ListenForCalls(chan)	Listen for 2G and 3G calls on specified channel; EndOfDwell event at end of current dwell
SelectSlot(pri)	Compute myCallingSlot using pri
WaitForSlot(slot)	Listens on myCurrentTrafficChannel until end of slot-1; SlotAvailable event if channel believed vacant, otherwise SlotOccupied (or R: xxx) event
U: LE_Link_Ind(callerAddr, callType)	Inform user process (Connection Manager) that a link has been established by a calling station
U: LE_Link_Confirm(destAddr)	Inform user process that link has been established to destAddr
U: LE_Status_Ind(status)	Inform user process (Connection Manager) of ALE status
U: LE_Link_Det_Ind(status, trafChan, caller, dest)	Inform user process that a change in link status between caller and dest has been detected (link established or terminated)
U: LE_Link_Fail_Ind(reason)	Inform user process (Connection Manager) that link has failed
S: LE_Call(destAddr, srcAddr, trafType, pri)	Send an LE_Call PDU
S: LE_Bcast(countdown, trafType, pri, chan)	Send an LE_Broadcast PDU
S: LE_Hshake(ID, CMD, ARG)	Send an LE_Handshake PDU
InitResponseTimeout	Set timeout for end of next slot
InitScanCallTimeout	Set timeout for maximum allowed scanning call duration
InitAsyncCount	Initialize asynchronous-mode broadcast countdown
InitTrafWaitTimeout(time)	Set timeout (trafWaitTime or trafWaitTimeMcast) to bound time waiting for traffic to start

C.5.2.4.3.4 3G-ALE synchronous mode protocol behavior.

Implementations of 3G ALE operating in synchronous mode shall exhibit the same over-the-air behavior as that described in Table C-XXXI.

TABLE C-XXXI. 3G-ALE synchronous mode protocol behavior.

State	Event	Condition	Action	Next State
Scanning	D: LE_Link_Req(sync, dest, PTPA, PTP1 or PTM, trafType, pri, chan)		TuneToNewChannel(chan) + ListenOnChannel(chan)	LBT
	D: LE_Link_Req(sync, dest, BCAST, count, trafType, pri, chan)		TuneToNewChannel(chan) + ListenOnChannel(chan)	LBT
	D: LE_Link_Req(sync, dest, RELEASE, trafType, pri, chan)		TuneToNewChannel(chan) + ListenOnChannel(chan)	ReleaseWait
	D: LE_Link_Req(sync, dest, NOTIFICATION, trafType, chan)		TuneToNewChannel(chan) + ListenOnChannel(chan)	NotificationWait
	R: LE_Call(myIndivAddr, srcAddr, callType is one of the PTPA types 0..3)	willing to link w/srcAddr & good traffic channel known	SelectChannelForTraffic(srcAddr, callType) + ComputeLinkID(srcAddr, myIndivAddr) + S: LE_Hshake(linkID, COMMENCE, prefTrafChan) + U: LE_Connect_Ind(srcAddr)	Linked
		not willing to link with srcAddr	ComputeLinkID(srcAddr, myIndivAddr) + S: LE_Hshake(linkID, ABORT, REJECTED)	Scanning
		willing to link w/srcAddr but no suitable traffic channel known	ComputeLinkID(srcAddr, myIndivAddr) + S: LE_Hshake(linkID, CONTINUE, NO_TRAF_CHAN)	Scanning
	R: LE_Call(myIndivAddr, srcAddr, PTP1)		InitResponseTimeout	R_Unicast
	R: LE_Call(dest, srcAddr, PTM)	dest addr is in myMulticastAddresses	InitResponseTimeout	R_Multicast
	R: LE_Call(dest, srcAddr, Control)		InitResponseTimeout	R_Control
	R: LE_Bcast(countdown, trafType, pri, chan)	broadcasts accepted	InitBcastTimeout(countdown) + broadcastPriority=pri + ListenOnChannel(chan)	R_Bcast
	R: 2G_Call	2G calls accepted	U: 2G_Call_Ind	Linked_2G
	others		queue or ignore	Scanning

TABLE C-XXXI. 3G-ALE synchronous mode protocol behavior (continued).

State	Event	Condition	Action	Next State
LBT	SlotAvailable	PTPA call	S: LE_Call(destAddr, myIndivAddr, callType) + ComputeLinkID(myIndivAddr, destAddr) + InitResponseTimeout	ListenForResponse
		PTP1 or PTM call	S: LE_Call(destAddr, myIndivAddr, callType) + ComputeLinkID(myIndivAddr, destAddr)	N_Commence
		broadcast call	S: LE_Bcast(myCurrentTrafficChannel, callType, count)	Send_Bcast
	SlotOccupied		U: LE_Link_Fail_Ind(reason = CALL_CHAN_OCCUPIED)	Scanning
	others		queue or ignore	LBT
ListenFor Response	R: LE_Hshake(id, cmd, arg)	id is correct, cmd=Commence	TuneToNewChannel(arg) + U: LE_Connect_Confirm(dest)	Linked
		id is correct, cmd=Continue	U: LE_Link_Fail_Ind(reason = NO_TRAF_CHAN)	Scanning
		id is correct, cmd = Abort	U: LE_Link_Fail_Ind(reason = REJECTED)	Scanning
		wrong id or other command	U: LE_Link_Fail_Ind(reason = NO_RESPONSE)	Scanning
	ResponseTimeout		U: LE_Link_Fail_Ind(reason = NO_RESPONSE)	Scanning
	others		queue or ignore	ListenFor Response
N_Commence (sending first PDU of a 1-way call)	FinishedSendingPDU	PTP1 or PTM call	S: LE_Hshake(linkID, COMMENCE or VOICE, myCurrentTrafficChannel) + U: LE_Connect_Confirm(dest)	Linked
	others		queue or ignore	N_Commence
Send_Bcast	FinishedSendingPDU	broadcastCount = 0	TuneToNewChannel(trafChan) + U: LE_Connect_Confirm(Broadcast)	Linked
		broadcastCount > 0		Send_Bcast
	others		queue or ignore	Send_Bcast
R_One_Way (listening for 2ndPDU of unicast or multicast call)	R: LE_Hshake(id, cmd, arg)	id is correct, cmd = Commence or Voice	TuneToNewChannel(arg) + U: LE_Connect_Ind(srcAddr)	Linked
		wrong id or other command		Scanning
	ResponseTimeout			Scanning
	others		queue or ignore	R_One_Way

TABLE C-XXXI. 3G-ALE synchronous mode protocol behavior (continued).

State	Event	Condition	Action	Next State
R_Bcast (waiting on bcast channel for TM)	U: BW1_Rcv_Ind		U: LE_Connect_Ind(trafType, BCAST)	Linked
	BcastTimeout		U: LE_Disconnect_Ind(TrafTimeout)	Scanning
	others		queue or ignore	R_Bcast
R_Control (listening for econd PDU of nk release)	R: LE_Hshake(id, cmd, arg)	id is correct, cmd=LinkRelease	U: LE_Link_Det_Ind(Available, trafChan, srcAddr, dest)	Scanning
		id is correct, cmd=SyncCheck	Compute offset in received time + S: LE_SyncOffset (time quality, offset)	Scanning
		wrong id or other command		Scanning
	ResponseTimeout			Scanning
	others		queue or ignore	R_Control
Linked	D: LE_Disconnect_Req (all other events routed via CM process)		U: LE_Disconnect_Conf( )	Scanning
	others		queue or ignore	Linked
ReleaseWait	SlotAvailable		S: LE_Call(destAddr, myIndivAddr, Control) + ComputeLinkID(myIndivAddr, destAddr)	LinkRelease
	SlotOccupied		U: LE_Link_Fail_Ind(reason = CALL_CHAN_OCCUPIED)	Scanning
	others		queue or ignore	ReleaseWait
LinkRelease	FinishedSendingPDU		S: LE_Hshake(linkID, LinkRelease, myCurrentTrafficChannel)	Scanning
	others		queue or ignore	LinkRelease
NotificationWait	SlotAvailable		S: LE_Notification(myIndivAddr, status)	Scanning
	SlotOccupied		U: LE_Link_Fail_Ind(reason = CALL_CHAN_OCCUPIED)	Scanning
	others		none	ReleaseWait
Linked_2G	D: LE_Disconnect_Req (all other events routed via CM process)		U: LE_Disconnect_Conf( )	Scanning
	others		queue or ignore	Linked_2G

C.5.2.4.3.5 3G-ALE asynchronous mode protocol behavior.

Implementations of 3G ALE operating in asynchronous mode shall exhibit the same over-the-air behavior as that described in Table C-XXXII. Note that Table C-XXXII adds state behavior to that specified for synchronous mode 3G ALE in Table C-XXXI.

TABLE C-XXXII. 3G-ALE asynchronous mode protocol behavior
(in addition to synchronous mode behavior in Table C-XXXI).

State	Event	Condition	Action	Next State
Scanning	D: LE_Link_Req(async, dest, callType, trafType, pri, channel)		TuneToNewChannel(channel) + ListenOnChannel(channel)	AsyncLBT
	R: LE_Scan_Call(addr)	addr is myIndivAddr or is subscribed multicast or PU is listening for LinkReleases	InitScanCallTimeout	ListenToCall
	others		queue or ignore	Scanning
AsyncLBT	SlotAvailable	not broadcast call	S: LE_Scan_Call(destAddr, count)	SendScanCall
		broadcast	InitBcastCount(count) + S: LE_Bcast(channel, callType, count)	AsyncBcast
	SlotOccupied		U: LE_Link_Fail_Ind(reason = CALL_CHAN_OCCUPIED)	Scanning
	others		queue or ignore	AsyncLBT
SendScanCall	EndOfScanCall	point-to-point call	S: LE_Call(destAddr, myIndivAddr, callType) + ComputeLinkID(myIndivAddr, destAddr) + InitResponseTimeout	ListenForResponse
		unicast or multicast call	S: LE_Call(destAddr, myIndivAddr, callType) + ComputeLinkID(myIndivAddr, destAddr)	N_Commence
	others		queue or ignore	Scanning
AsyncBcast	FinishedSendingPDU	broadcastCount = 0	TuneToNewChannel(trafChan) + U: LE_Connect_Confirm(Broadcast)	Linked
		broadcastCount > 0	DecrementBcastCount + S: LE_Bcast(channel, callType, count)	AsyncBcast
	others		queue or ignore	AsyncBcast

TABLE C-XXXII. 3G-ALE asynchronous mode protocol behavior (continued).

State	Event	Condition	Action	Next State
ListenToCall	R: LE_Call(myIndivAddr, srcAddr, callType is packet or circuit)	willing to link w/srcAddr & good traffic channel known	SelectChannelForTraffic(srcAddr, callType) + ComputeLinkID(srcAddr, myIndivAddr) + ListenOnChannel(trafChan) + U:LE_Link_Ind(srcAddr)	AsyncTrafCheck
		not willing to link with srcAddr	ComputeLinkID(srcAddr, myIndivAddr) + S: LE_Hshake(linkID, ABORT, UNAVAILABLE)	Scanning
		willing to link w/srcAddr but no suitable traffic channel known	ComputeLinkID(srcAddr, myIndivAddr) + S: LE_Hshake(linkID, CONTINUE, NO_TRAF_CHAN)	Scanning
	R: LE_Call(myIndivAddr, srcAddr, unicast)		InitResponseTimeout	R_One_Way
	R: LE_Call(dest, srcAddr, multicast)	dest addr is in myMulticastAddresses	InitResponseTimeout	R_One_Way
	ScanCallTimeout or R:Other			Scanning
	others		queue or ignore	ListenToCall
AsyncTraf Check	FinishedListening		SelectChannelForTraffic(srcAddr, callType) + S: LE_Hshake(linkID, COMMENCE, prefTrafChan) +TuneToNewChannel(trafChan)	Linked
	others		queue or ignore	AsyncTrafCheck

C.5.2.5 Notification protocol behavior.

Sending LE_Notification PDUs is optional. Network managers may wish to enable the notification protocol when the use of channel time for this overhead function provides a worthwhile return in tracking station and channel status.

C.5.2.5.1 Station status notification.

When station status notification is enabled, stations shall broadcast an LE_Notification PDU when one of the following occurs:

- Station status changes (or is about to change to a non-communicative state).
- A periodic timer prompts a notification.

A notification shall be sent on one or more channels that are selected by the ACS function to efficiently inform other network members of station status.

C.5.2.5.2 Sounding.

Sounding will normally be unnecessary in 3G-ALE systems. Knowledge of propagating channels can be used in synchronous networks to delay the start of calling and thereby reduce calling channel occupancy. However, with synchronous scanning, knowledge of propagating channels will have only slight effect on linking latency unless non-propagating channels are removed from the scan list (see Calling Channel Management, above).

When a synchronous network contains multiple "server" stations to provide geographic diversity for "client" stations calling into the server pool, the servers should sound to provide a database of propagation measurements at the client stations for use in selecting the best server to call. A synchronous sound consists of a single LE_Notification PDU sent as described in C.5.2.5.3.

In asynchronous 3G-ALE networks, sounding may be desired if propagation data is unobtainable by other means. In this case, periodic transmissions of a repeated LE_Notification PDU indicating Nominal station status may be employed (see C.5.2.5.4).

C.5.2.5.3 Synchronous mode notifications.

In networks operating in synchronous mode, LE_Notification PDUs shall be sent singly in the final slot of a dwell using the same listen before transmit procedure as for LE_Call PDUs: the station intending to send a notification shall listen in the slot preceding the final slot, and shall defer the transmission if it detects traffic or the first PDU of a handshake in that preceding slot.

C.5.2.5.4 Asynchronous mode notifications.

In networks operating in asynchronous mode, LE_Notification PDUs shall be sent M C + 1 times, after listening before transmitting, where C is the number of scanned channels, and M is the multiplier of C.5.2.4.2.2.

C.5.2.6 Calling into a 3G network.

Stations that have not been assigned an address in a network may call into that network by randomly selecting a Net Entry address (C.5.2.6.1) and placing a call using that address in accordance with an appropriate calling protocol. The call may be placed on any frequency known to be monitored by the network to be entered. Networks that support net entry by non-member stations should assign a well-known address (e.g., all 0's) to field such calls. Linking protection should be employed when spoofing is a concern.

Net entry calling and acceptance of net entry calls shall be supported by all 3G systems. A means shall be provided for operators to disable acceptance of net entry calls.

C.5.2.6.1 Net entry addresses.

Net Entry addresses are of the form 1111xxxxxxx. A station placing a net entry call shall randomly select one of these 128 addresses for use in a 3G-ALE calling protocol and subsequent protocols until it is assigned a member address.

C.5.2.6.2 Link establishment for net entry.

A station calling into a network operating in synchronous or asynchronous mode shall place an asynchronous-mode unicast call to a well-known address in accordance with C.5.2.4.2.4 3G-ALE asynchronous mode unicast call. When the calling station does not know the channel assignments in the foreign network, it should specify channel 127 which results in linking for traffic on the calling channel. When more than one of the frequencies scanned by the destination network are known, calling attempts should be placed on each channel in rotation until a link is established.

If the calling station seeks only a one-time analog voice link, the Call Type should be "Unicast Circuit" and the LE_Handshake PDU should carry a Commence Voice command. Otherwise, the TM protocol will normally be engaged after linking, so the Call Type should be "Unicast ARQ Packet" and the LE_Handshake PDU should carry a Commence Traffic Setup command.

C.5.2.6.3 Acquisition of operating data.

When a station calling into a network is to begin regular operation as a network member, the 3G packet protocol and the network management protocol of Appendix D should be used to transfer the following network operating data to the entering station:

- A station self address for use in the newly entered network (not a net entry address). Normally the calling station will provide its call sign and receive a 3G-ALE (11 bit) address in return.
- Channel table (frequencies, usage flags, and power limits); see C.4.11.3.

This transfer may be accomplished during the traffic phase of the initial net entry call, using the net entry address.

Synchronization of the entering station with network time shall comply with C.4.3 Network synchronization. If over-the-air synchronization will be required, it is recommended that operating data be set up in the new network member before the net entry synchronization protocol of C.5.2.7.6 is executed.

C.5.2.7 Synchronization management protocol (not tested).

3G networks operating in synchronous mode are intended to maintain synchronization using external means such as GPS receivers. This section describes a synchronization management protocol that is intended to serve as a fallback mechanism for use when external time references are unavailable or their use is otherwise impractical. Network managers should avoid use of this protocol when other alternatives are available because it requires use of the HF channels for this overhead function.

The synchronization management protocol supports the following tasks:

- Synchronization maintenance to compensate for time base drift
- Time service for late net entry
- Initial distribution of time to network member stations

This is an optional protocol. However, all 3G networks that must operate without external synchronization available to every station should implement these functions.

C.5.2.7.1 Synchronization data.

For successful operation in synchronous mode, third generation systems must maintain time base accuracy in accordance with C.4.3. The formula used by synchronous-mode stations to compute their current dwell channels in C.4.4.1 includes the time since midnight (network time). Network time is conceptually stored as a GPS week counter (week 0 was the week beginning 00:00 6 January 1980), a day of the week, elapsed seconds in the current day, and elapsed T_{sym} within the current second. A dwell counter is extracted from the seconds counter by dropping the two least-significant bits. Note that GPS time runs at the same rate as coordinated universal time (UTC), but GPS time does not add leap seconds and therefore differs by a small number of seconds from UTC.

In addition to a local estimate of network time, each station shall maintain a bound on the uncertainty (loss of accuracy) of this time base. This uncertainty value shall be set when the time base is adjusted, as described later, and shall increase steadily until the next time base update at a rate determined by the accuracy of the time base oscillator. When the oscillator has an accuracy of 1 part per million, the time base may drift by +/– 3.6 ms per hour, so the total width of time base uncertainty shall be increased by 7.2 ms per hour.

C.5.2.7.2 Time quality.

When one station sends a time update to another station, the uncertainty at the sending station shall be encoded as a Time Quality code in accordance with Table C-XXXIII. Note that only UTC sites may claim Time Quality 0. Stations that receive regular updates from a local GPS receiver or other stable time base that maintains their uncertainty below 1 ms shall report Time Quality 1. Other stations shall use the smallest code whose corresponding uncertainty value is greater than or equal to the local total uncertainty width.

TABLE C-XXXIII. 3G-ALE synchronization management time quality codes.

SM Time Quality Code	Total Time Uncertainty
0 (000)	none: UTC station
1 (001)	1 ms: local GPS receiver or equiv.
2 (010)	2 ms or stand-alone master station
3 (011)	5 ms
4 (100)	10 ms
5 (101)	20 ms
6 (110)	50 ms
7 (111)	unbounded or unknown

NOTE: When a network is operating in stand-alone mode (i.e., no network member has access to UTC, GPS, or equivalent time), one network member station should be designated as the master time reference, and that station should always use Time Quality 2. Of all station that have suitably stable oscillators, the designated station may be selected as the one with the lowest 3G-ALE address (e.g., a designated net entry server, with address zero).

C.5.2.7.3 Synchronization management PDUs.

The LE_PDUs (see figure C-17) used in synchronization management are described in the following paragraphs.

C.5.2.7.3.1 Group time broadcast PDU.

The group time broadcast PDU (LE_GTB PDU) conveys limited-precision time to any station that receives it. It is sent singly as described later in this section. The Server Group Number shall contain the dwell group number portion of the sending station address. The Dwell Number field shall contain the four least-significant bits of the counter of dwell periods since midnight network time. The Slot Number field shall indicate the slot in which the PDU is sent. (The Slot Number field is set to 7 during initial time distribution, as described in C.5.2.7.5.)

An LE_GTB PDU shall always be sent starting 256 T_{sym} after the beginning of the indicated slot at the sender. However, receiving stations may not know the propagation delay from sender to receiver, so this PDU by itself is insufficient to synchronize stations to meet the requirements of C.4.3.

> NOTE: each day contains an even multiple of 16 dwells so the four Dwell Number bits sent in this PDU increment naturally from 1111 to 0000 at midnight. The time indicated by this PDU should never be ambiguous unless (and only when) network time adds a leap second. For this reason, GPS time may be preferred over UTC.

C.5.2.7.3.2 Sync check PDU.

The Sync Check PDU is an LE_Handshake PDU containing a Sync Check command. It is sent following a Control-type LE_Call PDU during a sync check handshake, and shall always be sent 256 T_{sym} after the beginning of its slot at the sending station.

The most-significant bit of the Argument field shall be 0. The next three bits shall contain the time quality code from Table C-XXXIII corresponding to the total time uncertainty at the station sending the PDU. The three least-significant bits shall contain the number of the slot in which the PDU is sent: 001, 010, 011, or 100.

The Argument field may be set to all 1's when used in Late Net Entry Sync Acquisition (see C.5.2.7.6).

C.5.2.7.3.3 Sync offset PDU.

The LE_Sync Offset PDU is used to compensate for time base drift and propagation delay among stations during a Sync Check Handshake. It shall be sent by the responding station 512 Tsym (+/- 1 ms) after the end of a Sync Check PDU. The Quality field shall contain the time quality of the responding station in accordance with C.5.2.7.2. The Offset field shall indicate the magnitude of the difference between the time when the end of the Sync Check PDU arrives at the responding station and the "ideal time" when a PDU sent by the responding station in that same slot would have ended (i.e., beginning of the slot plus 256 Tsym plus 613 ms, which is 1728 Tsym or 720 ms into the slot). The Offset field shall be encoded in accordance with table C-XXXIII. The Sign bit shall be 1 if the PDU arrived early (before the ideal time), 0 if it arrived after the ideal time.

TABLE C-XXXIV. 3G-ALE synchronization management sync offset codes.

Sync Offset Code	Magnitude of Offset (ms)	Range of Offsets (ms)
0 - 50	2 x Code	0 – 100
51 – 175	100 + 10 x (Code – 50)	110 – 1350
176 - 511	1350 + 50 x (Code – 175)	1400 – 18,150

C.5.2.7.4 Sync check handshake.

The sync check handshake is used to update the time base at the station that initiates the handshake. It consists of a Control-type LE_Call PDU, followed by an LE_ Sync Check PDU from the initiator. The called station then responds with an LE_Sync Offset PDU. The initiating station shall compute its new local time and total time uncertainty as follows after receiving the LE_Sync Offset PDU:

1. The initiator shall measure the elapsed time between the end of its Sync Check PDU and the time of arrival of the end of the LE_Sync Offset PDU. The propagation delay T_{pd} shall be computed as T_{pd} = (Elapsed time – 826.7 ms) / 2.

2. If the LE_Sync Offset PDU Sign field is 0 (initiator is behind), the initiator shall subtract Tpd from the Offset field and add the result to its local time. Otherwise (Sign field = 1), the initiator shall add T_{pd} to the Offset field and subtract the result from its local time.

3. The initiator shall set its time base uncertainty to the value corresponding to the Quality code in the LE_Sync Offset PDU plus 5 ms to allow for unmeasured fluctuations in time of PDU release and in propagation delay.

A sync check handshake shall begin with equal probability in slot 1 or slot 2, and shall not begin in later slots.

C.5.2.7.5 Synchronization maintenance.

Stations operating in synchronous mode should request a time base update whenever their total time uncertainty will increase past the maximum allowed tolerance (C.4.3) within 60 minutes. A sync check handshake should then be initiated with the time server in its group. If no response (or a garbled response) is received, the initiating station should try again C + 1 dwells later on the next channel, and so on.

If a station’s time uncertainty grows past the limit of C.4.3, it must cease synchronous operation and attempt to reacquire network synchronization using the Late Net Entry procedure specified in C.5.2.7.6.

C.5.2.7.6 Synchronization for late net entry.

A station that is not synchronized to a network but whose time base is within +/- 30 s of network time may request and synchronize to network time using the following protocol. The protocol is more robust if the unsynchronized station knows the channel assignments of the network (see step 3), but may be used by a station that knows only one frequency that is monitored by the network stations.

1. The unsynchronized station (caller) initiates the protocol by sending an asynchronous Control call to a time server or other known address. The caller may use either a Net Entry address (C.5.2.6.1) or a network member address assigned as described in C.5.2.6.3. The call shall consist of LE_Scanning Call PDUs addressed to the called station (responder), a Control type LE_Call PDU addressed to the responder, and an LE_Sync Check PDU with the Argument field set to all 1s. This special value in the Argument field indicates that the call is a time request rather than a sync maintenance request.

2. In response to an LE_Sync Check PDU with the Argument field set to all 1s, the responder will return an LE_GTB PDU rather than a Sync Offset PDU. The LE_GTB PDU shall be sent 256 Tsym after the slot boundary that follows the end of the received LE_Sync Check PDU, and shall report the slot number of the slot it occupies (which may be slot 0). The LE_GTB PDU shall be sent on the frequency that carried the call. After sending the LE_GTB PDU, the responder shall remain on the same frequency, ignore the next slot and check the slot after that for an LE_Call PDU.

3. The caller should correct its local time using the time contained in the LE_GTB PDU, with the assumption that propagation time from the responder was zero, and set its time uncertainty to 70 ms. It should then initiate the synchronization maintenance protocol described above (C.5.2.7.4) in the second slot after the LE_GTB PDU. If no response (or a garbled response) is received, the station may continue to attempt Sync Check handshakes with the responder on the responder’s assigned dwell channels using the formula in C.4.4.1 if it knows the calling channels in use in the network. Otherwise it must restart this protocol at step 1.

4. If the responder receives error-free LE_Call and LE_Handshake PDUs containing the expected fields for a Sync Check handshake from the entering station, it shall complete the handshake by sending an LE_Sync Offset PDU as described in C.5.2.7.3.3. After sending this PDU, or after failing to receive the appropriate PDUs, the responder shall return to the Scanning state on its then-current assigned dwell channel.

An entering station shall not place any other synchronous call to the network until this synchronization is completed.

C.5.2.7.7 Initial time distribution.

Before a network is synchronized, stations should be scanning in asynchronous mode. Initial time distribution to a network begins with an asynchronous mode notification sequence, followed by an LE_GTB PDU from the master time station for the network (e.g., station 0). The repeated LE_Notification PDU shall contain the master time station address and a status of Time Server. The LE_GTB PDU shall contain a Slot Number value of seven, which indicates that the initial time distribution protocol is commencing. The PDU sequence that ends in this LE_GTB PDU shall be timed such that the LE_GTB PDU occurs in the final slot of a dwell (according to what will be network time), and the call shall be sent on the channel that the master time station would be monitoring during that dwell in synchronous mode.

Following receipt of this transmission, each station in the network shall compute the limited-precision time implied by the LE_GTB PDU, temporarily set its time base to this time, set its total time uncertainty to 70 ms and commence scanning its assigned channels in synchronous mode. In the following dwells, each dwell group will in turn monitor the channel that carried the time distribution transmission. During that dwell, the time server in each dwell group shall execute a Sync Check handshake with the network master time station to refine its time and set its time uncertainty.

After 32 dwells have elapsed since the end of the initial LE_GTB transmission, all stations shall stop scanning and remain on their current calling channel. Each dwell group will be on a distinct channel, and the time server in each group should have completed a Sync Check handshake with the master time station. Each such dwell group time server shall then transmit identical LE_Notification PDUs in slots 1, 2, 3, and 4 of the dwell, followed by an LE_GTB PDU in slot 5. The Slot Number field in this LE_GTB PDU shall be 5. Dwell group members shall perform Sync Check handshakes with their respective group time servers in the following 60 dwells, starting with member number 0 in the first dwell after this normal LE_GTB PDU, member number 1 in the next dwell and so on. After 60 dwells, all stations shall resume scanning on their assigned channels.

Breakdowns in this protocol are handled as follows:

1. Any dwell group time server that has not received the expected LE_Notification/LE_GTB sequence from the master time station within 8 minutes after the expected startup of the time distribution protocol should initiate the late net entry synchronization protocol from C.5.2.7.6, calling the master time station and, if it receives no response after calling on all channels, calling designated alternate master time station(s) and other group time servers.

2. Any dwell group member that received the initial LE_Notification/LE_GTB sequence from the master time station, but does not receive the expected LE_Notification/LE_GTB sequence from the group time server after listening for 60 dwells should attempt late net entry synchronization calling its dwell group time server first, followed by calls to the master and alternate master time stations.

3. Any dwell group member that does not receive the initial LE_Notification/LE_GTB sequence from the master time station within 10 minutes after the expected startup of the time distribution protocol should initiate the late net entry synchronization protocol from C.5.2.7.6, calling its group time server first, followed by calls to the master and alternate master time stations.

As each station achieves synchronization, it commences synchronous operation.

C.5.2.8 NATO-mode network addressing.

When 3G ALE is operating in NATO mode, the network number portion of the ARCS address of *called* station(s) shall be used to scramble 3G-ALE PDUs as shown in figure C-21.

The network number portion of the address of called station(s) shall be replicated to match the width of the key variable used for linking protection (see Appendix B). Denoting the most-significant bit of the network number as N_0 and the most-significant bit of the key as K_0, the next-most-significant bits as N_1 and K_1, and so on, each bit of key K_i is paired with bit $N_{i \bmod 13}$. The key variable V actually used to encrypt and decrypt 3G-ALE PDUs is the result of adding the paired bits modulo-2: $V_i = K_i +_2 N_{i \bmod 13}$

The network number of the calling station is neither sent over the air nor used in applying linking protection, but the XN flag indicates whether it is the same as the called network number.

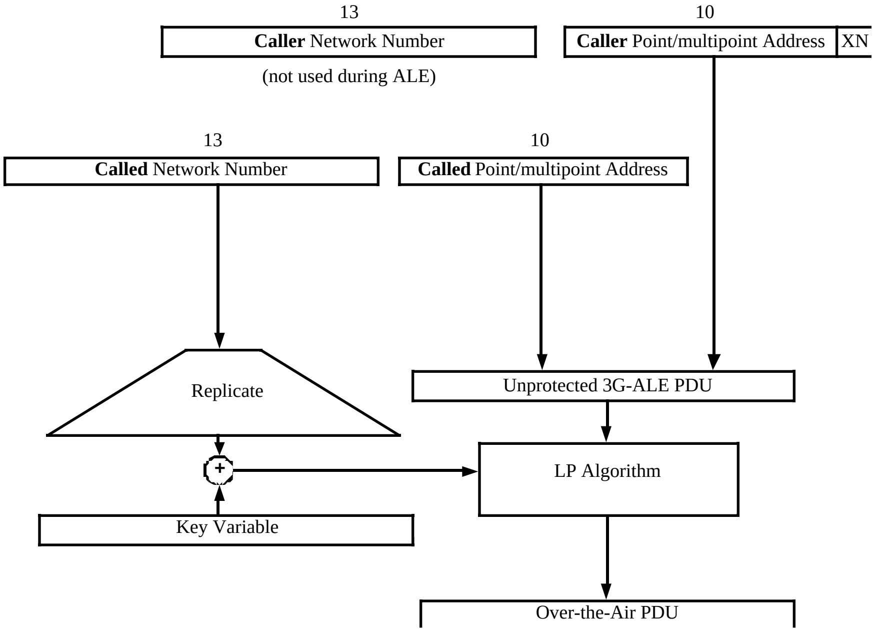

13
10
Caller Network Number
Caller Point/multipoint Address
XN
(not used during ALE)
13
10
Called Network Number
Called Point/multipoint Address
Replicate
Unprotected 3G-ALE PDU
+
LP Algorithm
Key Variable
Over-the-Air PDU

FIGURE C-21. NATO-Mode Address Processing using Linking Protection.

The Linking Protection (LP) procedures, data structures, and algorithms are specified in Appendix B. This section specifies additional LP rules for use in NATO mode 3G ALE. When operating in encrypt mode, the LP algorithm takes as inputs a PDU to be scrambled, a key variable, and a "seed" that contains Time of Day (TOD) and the frequency that carries a protected transmission. It produces as output a bit string the same length as the input PDU. Each bit of the output is a function of all of the input bits, so that a change in any input bit has a roughly 50% probability of inverting each output bit. In decrypt mode, the inputs are the key, the seed, and a received bit string; the output is a candidate PDU.

C.5.2.8.1 NATO mode normal LP operation

Unless LP is disabled or operating in Network Isolation Mode, calling stations shall use either the NATO asynchronous LP procedure or the NATO synchronous LP procedure (see below), depending on the type of call (async or sync, respectively).

The default LP Protection Interval (PI) shall be 24 hours for the NATO async procedure, and 108 seconds for the NATO sync procedure, respectively. All non-default LP PI settings shall be a multiple of 5.4 seconds.

Response PDUs shall be scrambled using the same LP procedure as used for the call.

Once a link is established, the LP procedure (Sync or Async) shall remain the same as used by the original caller for the duration of the link.

Once a linking process is started using a specific PI, the calling station shall not change the PI even if a PI boundary is crossed during the linking process.
Responding stations shall respond using the same PI as the caller used, even if a PI boundary is crossed during the linking process. Once linked, the PI shall change according to the sync or async rules used by the caller.

Stations that receive a 3G-ALE PDU while scanning shall test four possible PIs in the order shown, until one of the following is successful:

- The receive station’s current PI for the synchronous method (if the station is synchronized),
- The PI period nearest to the receive station’s current synchronous PI, if applicable. Clarification: If the receiving station’s TOD is nearer to its previous PI than to its next PI, then it should test the previous PI and the current PI. Likewise, if the receiver station’s TOD is nearer to the next PI, then it should test the next PI and the current PI.
- The receive station’s current async PI, and
- The PI period nearest to the receive station’s current async PI.

C.5.2.8.2 NATO network isolation mode

When the protection against spoofing offered by LP is not required, LP may be used without a key variable or seed to provide scrambling based on the network number alone. In this mode, the key and seed are forced to all 0 bits; the destination network number is replicated as described in C.5.2.8, exclusive-Ored with an equal-length string of 0 bits (i.e., is unchanged), and the result is used in the LP algorithm in place of the key variable.

C.5.2.8.3 NATO operation with LP disabled

LP may also be entirely disabled. In this case, PDUs are sent over the air unscrambled. Because network numbers are not used, this mode shall not be used when distinct networks share channels.

C.5.3 TM protocol.

C.5.3.1 Overview.

The TM protocol is used to coordinate traffic exchanges on connections established using the Third Generation ALE (3G-ALE) protocol. Following the end of the ALE phase in which a connection is established, the stations participating in the connection enter the Traffic Set-Up (TSU) phase in which the TM protocol is used to establish a traffic link on which traffic can be delivered. Once a connection has been established, the participating stations have determined:

1. the identities of the stations intended to participate in the connection;
2. the connection topology: point-to-point, multicast, or broadcast;
3. the link mode: packet or circuit ("hard link");
4. the HF frequency (or "traffic channel") that will be used for signalling within the connection.

In addition, each participating station knows whether or not it initiated the connection (even though stations other than the initiator do not always know which station originated the connection, as in broadcast connections), so that the initiating station knows it can transmit a TM PDU in the first transmit time-slot of the TSU phase.

During the TSU phase, the participating stations exchange TM PDUs in order to determine:

1. whether the connection will be used to deliver data or voice traffic, if it is a circuit connection;
2. which data link protocol(s), waveform(s), and/or baseband modulation formats will be used to deliver traffic on the connection;
3. the priority of the traffic to be delivered;
4. the fine time synchronization required for the HDL and LDL protocols, on traffic links established for packet traffic.

If the traffic link is a multicast circuit link (has a multicast topology), the participating stations initially conduct a roll-call procedure to determine which of the stations in the multicast group received the 3G-ALE signalling and are now present on the traffic frequency. A second roll-call can be conducted on the traffic link just before the traffic link is terminated and the participating stations resume scanning. This allows a station sending information on a Multicast circuit link to know whether the intended recipients of the information were on the traffic frequency to receive it, and allows the station initiating the traffic link to drop the current link and attempt to re-establish it if desired stations are absent from the link. When traffic exchanges have been completed on a traffic link, the TM protocol is used to coordinate the participating stations' departure from the traffic link.

SUPERSEDES PAGE 357 OF MIL-STD-188-141B.

C.5.3.2 Data object types.

The terms defined in Table C-XXXV are used to refer to specific types of data objects in defining the TM protocol.

TABLE C-XXXV. TM data object types.

Data object type	**Definition**	
traffic type	identifies a kind of traffic that can be delivered on a traffic link established using the TM protocol. The defined traffic types and their meanings are as follows:	
	value	**description**
	HDL_n	HDL (HDL), n data packets per forward transmission (n = 24, 12, 6, or 3)
	LDL_n	LDL , n payload data bytes per LDL forward transmission (n = 32, 64, ..., 512)
	ANDVT	Digitized voice using the ANDVT digitized voice coding method and modem waveforms defined by STANAG 4197 and STANAG 4198.
	DGTL_VOICE	Digitized voice using a non-ANDVT digitized voice coding method and/or modem waveform. The receiving station is assumed to be able to detect the voice coding and modem waveform and apply the appropriate receive processing.
	ANLG_VOICE	Analog voice traffic.
	SER_110	Bit-pipe data traffic delivered using the MIL-STD-188-110 serial tone modem signalling format.
	HQ_n SER_4285	Bit-pipe data traffic delivered using a high-rate data modem signalling format at n bits per second (n = 9600, 6400, 4800, or 3200). Bit-pipe data traffic delivered using the STANAG 4285 serial tone modem signalling format.
	PKT	packet traffic: refers to any traffic type that can be delivered on a packet traffic link: i.e., any of the HDL_n or LDL_n traffic types. Is used in contexts in which a station knows that a packet link is required, but not the specific type of packet traffic to be delivered on the link.
	CKT	circuit traffic: refers to any traffic type that can be delivered on a circuit traffic link, including all of the defined traffic types except HDL_n and LDL_n (which are delivered only on packet traffic links). Is used in contexts in which a station knows that a circuit link is required, but not the specific type of traffic to be delivered on the circuit link.
	NO_TR	no traffic: the sender has no traffic to deliver, and will await traffic from the other participant(s) in the traffic link.
	Note: In the TM behavior definitions, 'pktTraf' is used as an abbreviation for a traffic type of either HDL_n or LDL_n, which can be delivered only on a packet traffic link, or for 'NO_TR' (no traffic). 'cktTraf' is used as an abbreviation for any traffic type other than HDL_n or LDL_n -- i.e., any traffic type that can be delivered on a circuit traffic link – or for 'NO_TR' (no traffic).	

C.5.3.3 Service primitives.

Table C-XXXVI describes the service primitives exchanged between the TM entity and a user process at TM's upper interface. Note that there is no requirement that implementations of the waveforms and protocols defined in this Appendix contain precisely these service primitives; nor are the services primitives defined below necessarily all of the service primitives that would be required in an implementation of these waveforms and protocols.

TABLE C-XXXVI. TM service primitives.

Name	Attribute	Value	Description
TM_Connect_Req	Overview	TM Connect Request: issued to TM by the user process to request establishment of a traffic link.	
	Parameters	topology	Identifies the topology of the traffic link being established; one of: · P2P (point-to-point): the traffic link will contain the initiating station and a single called station. · MC (multicast): the traffic link will contain the initiating station together with all members (or as many as possible) of a defined multicast group. · BC (broadcast): the traffic link will contain the initiating station together with all other stations in the net that receive the ALE Broadcast PDUs used to place the broadcast call. The topology value must be 'P2P' if trafficType is any of the 3G data link traffic types HDL_n or LDL_n.
		trafficType	Identifies the type of traffic for which the traffic link is being established; value can be any of the traffic type values defined in table C-XXII.
		role	role of the local station in the established traffic link: MASTER (initiator of the link) or SLAVE.
		priority	priority level of the traffic (at the initiating station) for which the traffic link is being requested: HIGHEST, HIGH, ROUTINE, or LOW.
		addr	address of the station or group of stations to be included with the local station in the requested traffic link: an individual or multicast address, or a null (ignored) address for broadcast traffic links.
		reqIntvl	the duration of the time-interval through which TM should wait to receive a TM_REQ PDU from the initiating station before timing-out. The reqIntvl parameter-value is used only by slave stations in broadcast circuit links; it is ignored in all other cases.
	Originator	user process	
	Preconditions	1. A connection has just been established by 3G ALE, with this station as a participant. 2. No traffic link is established. I.e., the most recent service primitive (if any) passed between TM and the user process was a TM_Disconnect_Req, a TM_Disconnect_Ind, or a TM_Disconnect_Conf.	

TABLE C-XXXVI. TM service primitives (continued).

Name	Attribute	Value	Description
TM_Connect_Ind	Overview	TM Connect Indication: issued to the user process to indicate that a traffic link has been established at the request of a remote station, with the local station as a participant.	
	Parameters	trafficType	Identifies the type of traffic for which the traffic link is being established; value can be any of the traffic type values defined in table C-XXII. Obtained from the Argument (Traffic type) field of the TM_REQ PDU sent by the initiating station.
		priority	priority level of the traffic for which the traffic link has been established: HIGHEST, HIGH, ROUTINE, or LOW. This value is obtained from the Priority field-value of the TM_REQ PDU sent by the link master station to initiate the establishment of the link.
		srcAddr	station address of the station that initiated establishment of the link (the link master). Obtained from the Source Addr field-value of the TM_REQ PDU sent by the initiating station.
		responses	list of addresses of the members of the multicast group (for multicast links) from which a valid roll-call response was received; null (ignored) for point-to-point and broadcast links (on which no roll-call is performed).
	Originator	TM	
	Preconditions	No traffic link is established. I.e., the most recent service primitive (if any) passed between TM and the user process was a TM_Disconnect_Req, a TM_Disconnect_Ind, or a TM_Disconnect_Conf.	
TM_Connect_Conf	Overview	TM Connect Confirm: issued to the user process by TM, to confirm that a traffic link has been established as requested by a preceding TM_Connect_Req service primitive.	
	Parameters	responses	list of addresses of the members of the multicast group (for multicast links) from which a valid roll-call response was received; null (ignored) for point-to-point and broadcast links (on which no roll-call is performed).
	Originator	TM	
	Preconditions	Either a traffic link is being established at the request of the local user process; i.e., the most recent service primitive passed between TM and the user process was a TM_Connect_Req; or an established traffic link is being reversed after successful delivery of packet data; i.e., the most recent service primitive passed between TM and the user process was a TM_Connect_Ind.	

TABLE C-XXXVI. TM service primitives (continued).

Name	Attribute	Value	Description
TM_Disconnect_ Req	Overview	TM Disconnect Request: issued to TM by the user process, to request that the local station cease to participate in the current traffic link, and if the local station is the link master, that the traffic link be terminated entirely, with all of the remaining participants leaving the traffic frequency (or frequencies).	
	Parameters	reason	reason for disconnection. Value is one of: · RELINK: requests that the traffic link be re-established on a different channel. Used only on point-to-point traffic links. · SIGN_OFF: requests that the local station sign-off a multicast or broadcast circuit link, without necessarily causing the link to be terminated: other stations may stay linked. Used only at slave stations on multicast and broadcast circuit links. · ABORT: requests that the traffic link be immediately terminated. In broadcast or multicast circuit connections, can be issued only by the user process of the circuit master: the station that initiated the circuit. · UNLINK: requests that a multicast traffic link be terminated with a final roll call occurring just before the link is dropped. Used only on multicast circuit links; can be issued only at the master station: the station that initiated the circuit.
		period	on point-to-point circuit and multicast circuit links, the maximum amount of time that TM can wait for the link to become available so that the TM_TERM PDU sent as the station departs from the link does not collide with other traffic. After TM waits this amount of time, if the CLC still indicates that the link is busy, TM sends a TM_TERM PDU and drops the link regardless of any ongoing link activity. The period parameter-value is ignored (not used) on packet and broadcast circuit links.
	Originator	user process	
	Preconditions	A traffic link is currently established or is being established. I.e., TM has accepted a TM_Connect_Req service primitive since the most recent time at which it issued a TM_Disconnect_Ind or a TM_Disconnect_Conf, or accepted a TM_Disconnect_Req.	

TABLE C-XXXVI. TM service primitives (continued).

Name	Attribute	Value	Description
TM_Disconnect_ Ind	Overview	TM Disconnect Indication: issued to the user process to indicate that the local station has ended its participation in a traffic link for a reason other than the user process' having issued a TM_Disconnect_Req primitive.	
	Parameters	reason	reason for disconnection. Value is one of: · SIGN_OFF: indicates that the local station has left a traffic link due to another station's having signed-off the link — the other station in a unicast link, or the last remaining station in a multicast or broadcast circuit of which the local station was master. · ABORT: the local station has left a traffic link due to the link master station's having aborted the link. · EOM: the local station has left a packet traffic link due to successful completion of the packet data transfer and the absence of any packet traffic pending in the reverse direction. · RELINK: the remote station has initiated a re-link operation in which the participating stations will attempt to re-establish the traffic link on a different channel, by sending a TM_TERM PDU to the local station with Reason = RELINK. Used only on point-to-point traffic links. · REQ_TIMEOUT: the local station has left a traffic link due to failure to receive a TM_REQ PDU in the time period in which one was expected. If the traffic link being established was a unicast link, the two stations will attempt to re-link. · CONF_TIMEOUT: the local station has left a traffic link due to failure to receive a TM_CONF PDU in the time period in which one was expected. If the traffic link being established was a unicast link, the two stations will attempt to re-link. · TRF_TIMEOUT: the local station has left a circuit traffic link, due to the CLC's (CLC's) having detected no traffic on the circuit link over a time interval equal to its traffic timeout period. · UNLINK: the remote master station of the currently-established multicast circuit has requested that the circuit link be dropped after a final roll call ("unlink") is performed. A TM_Disconnect_Ind service primitive with reason = UNLINK requests that the user process respond with a TM_Disconnect_Resp service primitive indicating whether the local station has succeeded or failed in receiving the traffic delivered on the circuit link.
	Originator	TM	
	Preconditions	A traffic link is currently established or is being established. I.e., TM has accepted a TM_Connect_Req service primitive since the most recent time at which it issued a TM_Disconnect_Ind or a TM_Disconnect_Conf, or accepted a TM_Disconnect_Req.	
TM_Disconnect_ Resp	Overview	TM Disconnect Response: issued to TM by the user process to acknowledge that a currently-active multicast circuit is being "unlinked": i.e., dropped after a final roll call is performed.	
	Parameters	ackNak	Positive or negative acknowledgement of having received the traffic delivered on the multicast circuit. Possible values are · ACK: the traffic delivered on the multicast circuit was received successfully (with the user process determining what counts as "success") · NAK: the traffic delivered on the multicast circuit was not detected or received containing (an excessive quantity of) errors.
		watch	Boolean: if TRUE, TM will wait on the traffic channel to hear the unlink roll call responses from the other circuit participants. Otherwise, the station will wait only long enough to transmit its own roll call response in its unlink roll call time slot, and will immediately afterward return to 3G-ALE scanning.
	Originator	TM	
	Preconditions	1. A multicast circuit link is presently active. 2. TM has just issued a TM_Disconnect_Ind service primitive to the user process with reason = UNLINK.	

TABLE C-XXXVI. TM service primitives (continued).

Name	Attribute	Value	Description
TM_Disconnect_ Conf	Overview	TM Disconnect Confirm: issued to the user process by TM to acknowledge that a currently-active traffic link is being dropped as a result of a TM_Disconnect_Req service primitive.	
	Parameters	reason	The reason for which the traffic link is being dropped. Possible values and their meanings are the same as for the reason parameter of the TM_Disconnect_Req service primitive, as described above.
		responses	list of responses to an optional unlink roll-call performed at the conclusion of a multicast circuit connection, in which each response is in the form of an ordered pair (indAddr, ackNak), where indAddr is the address of a station whose roll call response was heard, and ackNak is the Reason field-value of the TM_TERM PDU sent as the station's response: UNL_ACK or UNL_NAK. The responses parameter has a value only when a multicast circuit link has been concluded with an unlink roll call. The list of responses can be incomplete for either of two reasons: 1. at a slave station, the user process has requested that the station remain in the circuit link only long enough to transmit its own roll call response, by setting the watch parameter of its TM_Disconnect_Resp primitive to FALSE. In this case, the reason parameter has the value UNLINK; and responses are not included in the list from those stations whose roll call time slots fall after the local station's time slot. 2. at either the master station or a slave station, the user process may have cut short the station's participation in the roll call, by issuing a TM_Disconnect_Req service primitive while the roll call is in progress. In this case, the reason parameter-value will be ABORT at the master station, or SIGN_OFF at a slave station; the response list will include only responses received before the TM_Disconnect_Req was accepted. The responses parameter is shown in the state diagrams only where it is used: where a multicast circuit link is being dropped with a concluding unlink roll call operation.
	Originator	TM	
	Preconditions		
TM_Suspend_ Req	Overview	TM Suspend Request: issued to TM by the user process, to suspend the current multicast circuit link. This primitive can be issued by the user process at the station which has initiated a multicast circuit link, when the responses to a just-completed roll call indicate that not all members of the multicast group are present in the circuit link. In response to the TM_Suspend_Req service primitive, the TM entity sends a TM_TERM PDU with reason = SUSPEND to hold the stations that answered the roll call on the traffic channel, then repeats the 3G-ALE multicast calling process in order to bring as many as possible of the remaining stations into the multicast circuit link.	
	Parameters	(none)	
	Originator	user process	
	Preconditions	A multicast circuit link initiated by the local station is currently active. I.e., since any other exchange of TM service primitives, TM has issued a TM_Connect_Conf service primitive to the user process whose responses parameter identifies those multicast group member stations which responded to the most recent multicast circuit roll call.	
TM_Resume_ Req	Overview	TM Resume Request: issued to TM by the user process, to cause the current multicast circuit link to be resumed after it has been suspended by means of a TM_Suspend_Req service primitive. In response to the TM_Resume_Req, TM sends a TM_REQUEST PDU on the traffic channel, to initiate an additional roll call and determine which of the multicast group members are now present on the traffic channel.	
	Parameters	(none)	
	Originator	user process	
	Preconditions	1. A multicast circuit link is currently suspended. 2. Since the multicast circuit link was suspended by means of a TM_Suspend_Req PDU, an additional 3G-ALE multicast call has been completed.	

C.5.3.4 PDUs.

The sub-sections of this section describe the PDUs exchanged between a TM entity and its remote peer entities. All TM PDUs have the common format and contents shown in Table C-XXXVII. Behavioral descriptions of the TM protocol refer to three kinds of PDUs: TM_REQ, TM_CONF, and TM_TERM. These PDUs all have the format shown in table XXXVI, and are distinguished from one another by the values of their Type fields:

- a “TM_REQ PDU” is a TM PDU having Type = TM_REQUEST (0)
- a “TM_CONF PDU” is a TM PDU having Type = TM_CONFIRM (1)
- a “TM_TERM PDU” is a TM PDU having Type = TM_TERM (2)

The field-values of each TM PDU are transmitted in order of their occurrence in Table C-XXXVII, starting with the protocol field. The bits of each field-value are transmitted in order of significance, starting with the most significant bit.

All of the TM PDUs are sent and received using the burst waveform BW1 described in section C.5.1.4. Each outgoing PDU is used as the Payload parameter value for a BW1_Send service primitive as described in table C-IV; each incoming PDU is received as the value of the Payload parameter of a BW1_Receive service primitive (see table C-XXXVII).

The TM entity at each station remains active while the station is exchanging voice or data traffic with other stations on an established traffic link, so as to be ready to drop the link on request. On traffic links established for packet traffic delivered using the HDL protocol (C.5.4) or the LDL protocol (C.5.5), the user process can terminate the data link transfer and use the next data link transmission time slot in either direction — i.e., the time slot for the xDL_DATA or the xDL_ACK PDU — to instead send a TM_TERM PDU (by issuing a TM_Disconnect_Req primitive) as many times as will fit within the data link PDU time-slot. This means that while a data link transfer is in progress, each station must be simultaneously attempting to demodulate TM_TERM PDUs conveyed by the BW1 waveform as it is attempting to demodulate and receive data link signalling conveyed by BW2, BW3, or BW4. Similarly, on a circuit link, each station must attempt to detect and demodulate TM_TERM PDUs conveyed by the BW1 waveform at all times when the station is not transmitting.

TABLE C-XXXVII. TM PDU format.

Field name	Size (bits)	Values	Description
Protocol	3	001_2 (fixed)	distinguishes TM PDUs from HDL and ALM PDUs
Priority	2	In all TM_REQ and some TM_CONF PDUs (i.e., TM PDUs having Type = TM_REQUEST or TM_CONFIRM), indicates the priority level of the traffic (if any) that the sender of the PDU intends to send on the traffic link once it is established. In TM_CONF PDUs, this field is used only when the Argument field value refers to one of the High-Rate (‘HDL_n’) or LDL (‘LDL_n’) traffic types as shown in table C-XXII. The field-value is set to 3 (LOW) in all other TM_CONF PDUs and in all TM_TERM PDUs, and must be ignored by the receiver.	
		0	HIGHEST: highest priority
		1	HIGH
		2	ROUTINE
		3	LOW: lowest priority

TABLE C-XXXVII. TM PDU format (continued).

Dest Addr	11	any	address of the station to which this PDU is being sent. Dest Addr can be the individual address of a single intended recipient station (abbreviated 'indAddr' below), a multicast address designating a multicast group ('MCaddr'), or all ones on a broadcast traffic link ('BCaddr') (see note). When the destination address is a multicast address, the lowest-order five bits of the address (corresponding to the dwell group number in an individual address) shall be all ones. In NATO networks, the destination station or multicast address occupies the ten most-significant bits. The least-significant bit is set to 0 in point-to-point links, and to 1 for multicasts.
Source Addr	11	any	address of the station that is sending this PDU. Is always the station address of a single station — never a multicast or broadcast address. In NATO networks, the source address occupies the ten most-significant bits. The least-significant bit is set to 1 when the source and destination network numbers are not the same, otherwise to 0.
Type	3	Type of PDU, indicating its role in the TM protocol. Note that the state diagrams and other materials refer to, for instance, a "TM_REQ PDU"; this is a TM PDU whose Type field value is 0 (TM_REQUEST).	
		0	TM_REQUEST: A PDU with Type = TM_REQUEST is sent in order to request that a traffic link be established between the station sending the TM_REQUEST and the other stations specified by the PDU's destination address.
		1	TM_CONFIRM: A PDU with Type = TM_CONFIRM is sent in response to a received TM_REQUEST PDU, to confirm the sender's readiness to participate in a traffic link.
		2	TM_TERM: a PDU with Type = TM_TERM is sent in order to terminate the station's participation in a traffic link (during or after link establishment), and when sent by the link master, to terminate the link as a whole.
		3..7	reserved
Argument	6	variant field whose usage and meaning depend on the value of the Type field; when Type is TM_REQUEST or TM_CONFIRM, Argument is Traffic Type (see TABLE XXII); when Type is TM_TERM, Argument is Reason for termination; see below:	
		0 ("ABORT")	Immediately terminate the traffic link, with all participating stations leaving the traffic frequency (-ies) assigned to the link. Reason = ABORT indicates nothing about any measures that may be taken to resume any data transfer that may have been in progress.
		1 ("RELINK")	Immediately terminate the traffic link, with all participating stations leaving the traffic frequency (-ies) assigned to the link. Reason = RELINK indicates that the user process will attempt to resume the data transfer, possibly on a different frequency or frequencies.
		2 ("SIGN_OFF")	The station sending the TM_TERM PDU is departing the multicast circuit link, of which it is not the master. If two or more stations remain on the link, they may continue to exchange traffic.
		3 ("UNLINK")	Is sent by the initiator of a multicast circuit link, to cause the link to be torn-down after a final roll call is performed.
		4 ("UNL_ACK")	Is sent by a participant in a multicast circuit link in response to a TM_TERM PDU with Reason = UNLINK, to indicate that the station has successfully received all traffic sent on the multicast circuit (see note).
		5 ("UNL_NAK")	Is sent by a participant in a multicast circuit link in response to a TM_TERM PDU with Reason = UNLINK, to indicate that the station is still present in the multicast circuit, but has not received all traffic sent on the multicast circuit successfully.
		6 ("SUSPEND")	Suspends the current multicast circuit link while the link initiator repeats the 3G-ALE multicast call in order to retrieve as many as possible of the multicast group members that were found to be absent in the most recent roll call. Stations receiving the TM_TERM PDU with Reason = SUSPEND are expected to remain on the traffic channel for a time period sufficient to allow the link initiator to complete the 3G-ALE multicast call, return to the traffic channel, and send a TM_REQ PDU to start another roll call.
		7 - 63	reserved
CRC	12	any	12-bit Cyclic Redundancy Check (CRC) computed across the remaining 36 bits of each TM PDU, using the generator polynomial $X^{12} + X^{11} + X^9 + X^8 + X^7 + X^6 + X^3 + X^2 + X^1 + 1$, and the procedure described in C.4.12.
NOTE: The destination address has no significance on broadcast links; this field is set to all ones purely by convention. The all ones address vaule is not a reserved broadcast address, and hence can also be used as an individual or multicast address.			

C.5.3.5 Protocol behavior.

The following sections define the behavior of the TM protocol:

- C.5.3.5.1 identifies and defines the events to which the TM entity responds;
- C.5.3.5.2 identifies and defines the actions taken by the TM entity in response to these events;
- C.5.3.5.3 describes the data items used and maintained by a TM entity;
- C.5.3.5.4 provides state diagrams and a state transition table specifying the behavior of the TM entity in terms of these events, actions, and data items; and
- C.5.3.5.5 provides additional information on the timing characteristics of TM protocol behavior.

C.5.3.5.1 Events.

Table C-XXXVIII defines the events to which the TM entity responds. The event names are used in the state diagrams or state transition tables in C.5.3.5, which define the behavior of the TM protocol. Some event names refer to the receipt of PDUs from the TM entity at a remote station; in these cases, either the PDU definition in C.5.3.4 or the ‘description’ field of the table entry describes the manner in which the arrival of a PDU is accomplished through TM’s accepting one or more service primitives from lower-layer entities at the local station. The prefix ‘R:’ in the name of an event indicates that the event is the receipt of a PDU from the remote station. The prefix ‘D:’ indicates that the event is the TM entity’s accepting a service primitive from a higher-layer entity (the primitive is passed ‘downward’), while the prefix ‘U:’ indicates that the event is the TM entity’s accepting a service primitive from a lower-layer entity (the primitive is passed ‘upward’). Event names are used in the state diagrams and the transition table precisely as shown here, with the following exception: italicized words in the event names shown here are substitution variables for which explicit parameter- or field-values are substituted when these event names are used in the state diagrams and the transition table.

TABLE C-XXXVIII. TM events.

Event name	Description
ConfirmTimeout	A TM_CONF PDU was not received in the time period in which it was expected, in the two-way handshake performed to establish a point-to-point traffic link for packet or circuit traffic, as indicated by a timeout of ConfirmTimer.
D:TM_Connect_Req (topology, trafficType, role, addr, reqIntvl)	A TM_Connect_Req service primitive was received from the user process, with the indicated values for the topology, trafficType, role, addr, and reqIntvl parameters. In the state diagrams and state transition table, · '*topology*' is replaced by 'P2P' (point-to-point), 'MC' (multicast), or 'BC' (broadcast), indicating the topology of the traffic link being established · '*role*' is replaced by 'MASTER' or 'SLAVE', where the link master is the station initiating the traffic link · '*trafficType*' is replaced by one of the traffic type values defined in table C-XXII. The values 'PKT' and 'CKT' are used whenever role = SLAVE, since the slave station does not yet know the specific type of traffic that is to be delivered on the traffic link (as this information is not conveyed by 3G ALE). 'pktTraf' is used as an abbreviation for any of the HDL_n or LDL_n traffic types which can be delivered on a packet traffic link; 'cktTraf' is used for any traffic type other than HDL_n or LDL_n, which can be delivered on a circuit traffic link. · '*addr*' is replaced by either 'indAddr' (the remote station participating in a packet or point-to-point circuit link), 'MCaddr' (the address of the called multicast group), or 'BCaddr' (the all-ones broadcast address). Note that addr is ignored whenever topology = BC; although an all-ones broadcast value is transmitted in TM PDUs, this address value has no significance. · '*reqIntvl*'' is used only at slave stations participating in broadcast circuit links. The value of reqIntvl is based on the Countdown field-value of the received LE_BROADCAST PDU.
D:TM_Disconnect_Req (reason) D:TM_Disconnect_Req (reason, period)	A TM_Disconnect_Req service primitive was received from the user process, having the indicated value, or one of the provided list of values, as its reason parameter. '*reason*' may be either a single parameter value (e.g., "ABORT") or a list of two or more possible reason values separated by 'pipe' characters ('|') (e.g., "ABORT|RELINK"). An event name containing the word 'reason' instead of a specific parameter-value applies to any TM_Disconnect_Req service primitive containing any value for the reason parameter. The value of the period parameter determines the maximum length of time that TM will wait for the link to become available before sending a TM_TERM PDU, if the link is busy (as determined by CLC) at the time the TM_Disconnect_Req service primitive is accepted from the user process. The period parameter is shown on the state diagrams only in those situations in which it is used: i.e., on circuit traffic links after the link has been established (and hence could be busy). In all other contexts, the period parameter-value is ignored.
D:TM_Suspend_Req	A TM_Suspend_Req service primitive was received from the user process.
D:TM_Resume_Req	A TM_Resume_Req service primitive was received from the user process.
DropTimeout	The time limit limiting the period through which TM can wait for the traffic link to become idle before sending a TM_TERM PDU in response to a TM_Disconnect_Req has been exceeded, as indicated by a timeout of DropTimer.

TABLE C-XXXVIII. TM events (continued).

Event name	Description
EndOfMCRollCall	Indicates that the time period in which stations participating in a multicast circuit link are expected to transmit their roll-call responses has ended, as indicated by the RollCallTimer.
EndOfUnlink	Indicates that the time period in which stations participating in a multicast circuit link are expected to transmit their unlink roll-call responses has ended, as indicated by the UnlinkRollCallTimer.
MyRCSlot	Indicates that the time-slot allocated to the local station for transmission of its roll call response has arrived, as indicated by the RollCallTimer. See C.5.3.5.5 for a specification of the timing of the multicast roll-call operation. Roll call time slots are assigned to multicast group member stations in the following manner: 1. The individual addresses of the multicast group member stations are placed in a list. 2. If the link master (the station that initiated the link) is a member of the multicast group, its individual address is removed from the list (see note). 3. The list of addresses is sorted by increasing dwell group number, and, for stations in the same dwell group, by increasing member number (so that, for instance, {group #4, member#5} precedes {group #5, member #2}, and {group #5, member #2} precedes {group #5, member #4}). 4. The station whose address is first in the list is assigned the first roll call slot (the slot immediately following the transmission of the TM_REQUEST PDU that initiated the roll call), the one whose address is second gets the second slot, and so forth.
other	Refers to any event not corresponding to any of the explicit event labels on transitions leaving the current state.
R:other	Refers to the receipt of any PDU not described explicitly by any of the event labels on transitions leaving the current state.
R:TM_CONF (pktTraf, srcAddr) R:TM_CONF (cktTraf, srcAddr)	A TM_CONF PDU was received from the station with individual address srcAddr, confirming establishment of the traffic link requested by a preceding TM_REQ PDU. The Traffic Type field value (represented by ‘pktTraf’ or ‘cktTraf’) should be identical to that of the TM_REQ PDU sent most recently by the local station; received TM_CONF PDUs in which this is not the case shall be ignored. If the requested traffic link is a multicast circuit link, then the TM_CONF PDU is a roll call response, which should be received in the correct roll call time-slot of the station having srcAddr as its individual address; any TM_CONF PDU having an incorrect source address for the roll call time slot in which it was received shall be ignored. On point-to-point links, the TM_CONF PDU should be received immediately following the transmission of the TM_REQ PDU to which it is a response; otherwise, a ConfirmTimeout occurs.
R:TM_REQ (pktTraf, srcAddr) R:TM_REQ (cktTraf, srcAddr)	A TM_REQ PDU was received from the station with individual address srcAddr, requesting establishment of a traffic link for delivery of the traffic type represented by ‘pktTraf’ (HDL_n or LDL_n traffic) or ‘cktTraf’ (circuit traffic: i.e., neither HDL_n nor LDL_n).
R:TM_REQ (cktTraf, srcAddr, MCaddr)	A TM_REQ PDU was received from the station with individual address srcAddr, requesting establishment of a multicast circuit link containing members of the multicast group having address MCaddr, for delivery of the traffic type represented by ‘cktTraf’ (circuit traffic).
R:TM_TERM (reason)	A TM_TERM PDU, having RELINK, ABORT, or SIGN_OFF as the value of its reason parameter, was received from the remote station participating in a point-to-point link.

TABLE C-XXXVIII. TM events (continued).

Event name	Description
R:TM_TERM (ABORT, srcAddr)	A TM_TERM PDU was received from the station with address 'srcAddr', with reason = ABORT: the sending station is dropping a currently-established circuit link of which it was the master participant.
R:TM_TERM (SIGN_OFF, srcAddr)	A TM_TERM PDU was received from the station with address 'srcAddr', with reason = SIGN_OFF: the sending station is signing-off a currently-established circuit link in which it was a slave participant.
R:TM_TERM (SUSPEND, srcAddr)	A TM_TERM PDU was received from the station with address 'srcAddr', with reason = SUSPEND: the sending station is suspending a currently-established multicast circuit link of which it was the master participant, while it repeats the multicast call so as to attempt to include additional stations in the circuit link.
R:TM_TERM (UNLINK, srcAddr)	A TM_TERM PDU was received from the station with address 'srcAddr', with reason = UNLINK: the sending station is dropping a currently-established multicast circuit link in which it was the master station, and requesting that the stations present in the circuit respond to an unlink roll-call as they leave the circuit.
R:TM_TERM (ackNak, srcAddr)	A TM_TERM PDU was received from the station with address 'srcAddr', with reason = UNL_ACK or UNL_NAK: the sending station is leaving the current multicast circuit link, and indicating that it did (if reason = UNL_ACK) or did not (if reason = UNL_NAK) successfully receive the traffic transmitted on the circuit.
RequestTimeout	A TM_REQ PDU was not received in the time period in which it was expected, in the course of an attempt to establish a traffic link, as indicated by a timeout of RequestTimer.
ReversalTimeout	A TM_REQ PDU was not received in the time period in which it was expected, in the course of an potential packet traffic link reversal, as indicated by a timeout of ReversalTimer.
U:CLC_Avail_Ind	A CLC_Avail_Ind service primitive was received from the CLC, indicating that the traffic link is available for new traffic (i.e., not busy).
U:CLC_Busy_Ind	A CLC_Busy_Ind service primitive was received from the CLC, indicating that the traffic link is busy (i.e., not available for new traffic).
U:CLC_Idle_Ind	A CLC_Idle_Ind service primitive was received from the CLC, indicating that a traffic timeout occurred: no outgoing or incoming traffic was detected on the circuit link over a time period equal to the traffic timeout interval.
U:xDL_Rcv_Ind	An HDL_Rcv_Ind or LDL_Rcv_Ind service primitive was received from, respectively, the High-Rate or LDL, indicating that the data link has just successfully completed an incoming datagram transfer.
U:xDL_Send_Conf	An HDL_ Send_Conf or LDL_ Send_Conf service primitive was received from, respectively, the High-Rate or LDL, indicating that the data link has just successfully completed an outgoing datagram transfer.
NOTE: It is considered unnecessary for the link master to respond to the roll call, since it has indicated its presence by sending the original TM_REQUEST. No roll call response time slot is assigned to the link master at all, since assigning one and leaving it unused could cause a listening station to erroneously declare the channel unused (and hence available) if it were to listen on the channel during the unused time slot.	

C.5.3.5.2 Actions.

Table C-XXXIX defines the actions which the TM entity can perform. The action name is used in the state diagrams and/or state transition tables used below to define the behavior of the TM protocol. Some action names refer to sending PDUs to the TM entity at a remote station; in these cases, the PDU definition in C.5.3.4 or the 'description' field of the table entry describes the manner in which sending of the PDU is accomplished by issuing one or more service primitives to subordinate entities at the local station. Action names are used in the state diagrams and the transition table precisely as shown here, with the following exception: italicized words in the action names shown here are substitution variables for which explicit parameter- or field-values are substituted when these action names are used in the state diagrams and the transition table.

TABLE C-XXXIX. TM actions.

Action name	Description
D:CLC_Active_Req	Issue a CLC_Active_Req service primitive to the CLC, requesting that it begin monitoring and arbitrating access to the newly-established circuit link.
D:CLC_Idle_Req	Issue a CLC_Idle_Req service primitive to the CLC, requesting that it cease monitoring and arbitrating access to the circuit link.
D:CLC_Set_Priority (prio)	Issue a CLC_Set_Priority service primitive, giving it a new priority value for pending outgoing traffic (if any) for the current circuit link. Where the word 'prio' occurs in place of an explicit value for the prio parameter, this indicates that the prio parameter has assigned to it the value of the priority parameter of the TM_Connect_Req primitive that caused the current traffic link to be established.
InitRollCall	Initialize (empty) RollCallResponses; initialize and start RollCallTimer.
InitUnlink	Initialize (empty) UnlinkResponses; initialize and start UnlinkRollCallTimer.
InitWaitForConfirm	Initialize ConfirmTimer to start timing the time-interval in which an incoming TM_CONFIRM PDU is expected in the establishment of a packet or point-to-point circuit link. The timeout interval duration is $T_{BW1} + 2T_{prop,max} + T_{BW1enc} + 2T_{BW1proc}$ following the emission of the last sample of an outgoing TM_REQUEST PDU, where the 'T_x' duration constants are as defined in C.5.3.5.5.
InitWaitForRequest	Initialize RequestTimer to start timing the time-interval in which an incoming TM_REQUEST PDU is expected in the establishment of a packet or point-to-point circuit traffic link. The timeout interval duration is $T_{tune} + 2T_{sug} + T_{prop,max} + T_{BW1} + T_{BW1proc}$ following the end of the 3G-ALE time slot in which the LE_HANDSHAKE PDU was transmitted or received, where the 'T_x' duration constants are as defined in C.5.3.5.5.
InitWaitForRequest (MC)	Initialize RequestTimer to start timing the time-interval in which an incoming TM_REQUEST PDU is expected in the establishment of a multicast circuit traffic link. The timeout interval duration is $T_{traf\_wait\_mcast} + T_{tune} + 2T_{sug} + T_{prop,max} + T_{BW1} + T_{BW1proc}$ following the end of the 3G-ALE slot in which the LE_HANDSHAKE PDU specifying the traffic channel for the circuit link was received, where $T_{traf\_wait\_mcast}$ is an initial set-up parameter, and the remaining 'T_x' duration constants are as defined in C.5.3.5.5.
InitWaitForRequest (SUSPEND)	Initialize RequestTimer to start timing the time-interval in which an incoming TM_REQUEST PDU is expected after an established multicast circuit traffic link has been suspended. The timeout interval duration is $2T_{BW1} + T_{BW1proc} + T_{traf\_wait\_mcast} + 2T_{dwell} + T_{tune} + 2T_{sug} + T_{prop,max} + T_{BW1} + T_{BW1proc}$ following the instant in which the last sample of the TM_TERM(SUSPEND) PDU was received, where $T_{traf\_wait\_mcast}$ is an initial set-up parameter, and the remaining 'T_x' duration constants are as defined in C.5.3.5.5.
InitWaitForRequest (reqIntvl)	Initialize RequestTimer to start timing the time-interval in which an incoming TM_REQUEST PDU is expected in the establishment of a broadcast traffic link. The value of reqIntvl is supplied by the user process as the reqIntvl parameter-value of the TM_Connect_Req service primitive, and is based on the countdown field-value of the received LE_BROADCAST PDU.

TABLE C-XXXIX. TM actions (continued).

Action name	Description
InitWaitForReversal	Initialize ReversalTimer to start timing the time-interval in which an incoming TM_REQUEST PDU is expected at the former sending station in a packet traffic link reversal. The timeout interval duration is T_{BW1} + $T_{BW1proc}$ following the instant at which the arrival of the first sample of an xDL_ACK PDU would have been expected if the preceding data link transfer had not been already completed. In this case, if a request timeout occurs, TM assumes that the remote station did not attempt to reverse the packet traffic link; it is up to the user process at the remote station to set-up a new traffic link (via 3G ALE and TM) to deliver any packet traffic it has, if an attempted packet link reversal fails.
MarkAbsent (srcAddr)	Record in RollCallResponses the fact that no roll call response was received from the station with address srcAddr.
MarkLinkAvail	Set LinkBusy to FALSE (since the link is "available").
MarkLinkBusy	Set LinkBusy to TRUE.
MarkPresent (srcAddr)	Record in RollCallResponses the fact that a roll call response was received from the station with address srcAddr.
MarkReverseTrafficPending	Set ReverseTrafficPending to TRUE.
none	No action.
RecordUnlinkResponse (ackNak, srcAddr)	Add an entry to UnlinkResponses representing a received unlink roll call response from the station whose individual address is srcAddr, and ackNak is the Reason field value of the response (TM_TERM) PDU: UNL_ACK or UNL_NAK.
S:TM_CONF (pktTraf, srcAddr) S:TM_CONF (cktTraf, srcaddr)	Send a TM_CONF ("Confirm") PDU with destAddr = the individual address of the station that initiated the traffic link being established by sending a TM_REQ PDU, and trafficType = the traffic type announced in the TM_REQ PDU. 'pktTraf' represents an announced packet traffic type (HDL_n or LDL_n); 'cktTraf' represents an announced circuit traffic type (neither HDL_n nor LDL_n).
S:TM_CONF (cktTraf, MCaddr)	Send a TM_CONF PDU in the local station's roll call time slot., with destAddr = the multicast address of the multicast group for which the circuit link is being established, and trafficType equal to the circuit traffic type value announced in the TM_REQ PDU sent by the link master to initiate the roll call operation.
S:TM_REQ (trafficType, destAddr)	Send a TM_REQ PDU requesting establishment of a traffic link, with trafficType = the type of traffic to be delivered on the requested link, and destAddr identifying the stations intended to participate in the link. In the state diagrams and transition table, 'pktTraf' is used to refer to any form of packet traffic delivered by either the High-Rate ('HDL_n') or the LDL ('LDL_n'); 'cktTraf' is used to refer to any traffic type that can be delivered on a circuit link: i.e., any type other than HDL_n or LDL_n. For circuit links, the type of destination address depends on whether the requested link is a point-to-point, multicast, or broadcast link; this is signified in the state diagrams and transition table by the use of the abbreviations 'indAddr', 'MCaddr', and 'BCaddr'.

TABLE C-XXXIX. TM actions (continued).

Action name	Description
S:TM_TERM (reason, addr, howMany)	Send one or more TM_TERM PDUs having the indicated values for the Reason and Dest Addr fields. addr is the station address of the remote participant in a point-to-point link (circuit or packet), the multicast address of the multicast group participating in a multicast circuit link, or the all-ones broadcast address. The howMany term (which does not refer to a PDU field) indicates how many times the TM_TERM PDU is to be sent: once, if no explicit howMany value is provided; three times, if howMany = ‘3x’; and as many times as possible within a High-Rate or LDL forward transmission interval, if howMany = ‘nx’; for example: once, for traffic type HDL_3 or HDL_6; three times, for traffic type HDL_12; seven times, for traffic type HDL_24; once, for traffic type LDL_64 or LDL_128; two times, for traffic type LDL_256; and five times, for traffic type LDL_512. When ‘reason’ occurs in the state diagrams or transition table in place of an explicit Reason field-value, this indicates that this field is to be given the value of the Reason parameter of the TM_Disconnect_Req service primitive most recently received from the user process. When ‘ackNak’ is shown in the reason position, this indicates that the Reason field-value is to be either UNL_ACK or UNL_NAK, depending on whether the station successfully received the traffic transmitted on the multicast circuit which is now being dropped. In this case, the Reason field-value should be the same as the ackNak parameter-value of the TM_Disconnect_Resp service primitive just accepted from the user process.
ScheduleAbort	Set ScheduledAbort to TRUE.
ScheduleSignoff	Set ScheduledSignoff to TRUE.
SetDropTimeout (period)	Set DropTimer to time out and generate a DropTimeout event after an interval of duration period has elapsed.
SetupUnlink (ackNak, watch)	Record whether the traffic transmitted on the multicast circuit now being dropped was received successfully, as indicated by the ackNak parameter-value of the TM_Disconnect_Resp service primitive just accepted from the user process. Also record whether the local station is to remain on the traffic channel long enough to hear all of the unlink roll call responses from the participants in the multicast circuit.
U:TM_Connect_Conf U:TM_Connect_Conf (responses)	Issue a TM_Connect_Conf service primitive to the user process, confirming establishment of the requested traffic link, of which the local station is now link master. If the established link is a multicast circuit link, the ‘responses’ parameter identifies the stations from which roll call responses were received; this parameter is omitted when the link is a point-to-point or broadcast link.

TABLE C-XXXIX. TM actions (continued).

Action name	Description
U:TM_Connect_Ind (trafficType, srcAddr, responses)	Issue a TM_Connect_Ind service primitive to the user process, indicating that a traffic link has been established of which the local station is a non-master participant. The trafficType parameter identifies the type of traffic for which the traffic link is being established, and should habe the same value as the Argument (Traffic Type) field of the TM_REQ PDU that was sent by the station initiating the traffic link. The srcAddr parameter is the individual address of the station that initiated the traffic link. If the established link is a multicast circuit link, the responses parameter identifies the stations from which roll call responses were received; this parameter is omitted when the link is a point-to-point or broadcast link.
U:TM_Disconnect_Conf (reason) U:TM_Disconnect_Conf (reason, responses)	Issue a TM_Disconnect_Conf service primitive with its reason parameter having the indicated value to the user process. Where the word ‘reason’ occurs in place of an explicit value for the reason parameter, this indicates that the reason parameter has assigned to it the value of the reason parameter of the TM_Disconnect_Req primitive to which the TM_Disconnect_Conf primitive is a response. The responses parameter is present only if the link being dropped is a multicast circuit link and if an unlink roll call has been performed; in this case, the value of the responses parameter is a list of the unlink roll call responses received by the local station.
U:TM_Disconnect_Ind (reason)	Issue a TM_Disconnect_Ind service primitive with its reason parameter having the indicated value to the user process. Where the word ‘reason’ occurs in place of an explicit value for the reason parameter, this indicates that the reason parameter has assigned to it the value of the ‘reason’ field of a just-received TM_TERM PDU.

C.5.3.5.3 Data.

Table C-XL defines the data items used and maintained by the TM entity, including buffers, counters, timers, configuration parameters, and so forth. These data items are referred to by the names assigned to them here, in the definitions of TM events and actions presented in the preceding sections. These data items are used in this specification only as expository devices; it is not required for compliance that an implementation contain these data items in the forms described here.

TABLE C-XL. TM data items.

Data item	Description
ConfirmTimer	Timer timing the period in which receipt of a TM_CONF PDU is expected in response to a TM_REQ PDU just sent.
DropTimer	Timer timing the interval TM waits for the traffic link to become available (no longer busy) as indicated by CLC, so that TM can send a TM_TERM PDU in response to a TM_Disconnect_Req.
LinkBusy	Boolean condition variable: is TRUE if and only if CLC has declared the current circuit link to be busy (without since then having declared it to be non-busy — i.e., available for new traffic).
RequestTimer	Timer timing the period in which receipt of a TM_REQ PDU is expected, when the local station is a slave participant in a connection which has just been established by 3G ALE. The timeout period varies depending on the traffic link topology; see the descriptions of the InitWaitForRequest() actions for further details.
ReversalTimer	Timer timing the period in which receipt of a TM_REQ PDU is expected, when the local station has just completed an outgoing data link transfer on a packet traffic link, and is waiting for an indication that the remote station has data link traffic to send on the traffic link.
ReverseTrafficPending	Boolean condition variable; when TRUE, indicates that the non-master participant in a packet traffic link has packet traffic to send to the link initiator, and hence that the link direction will be reversed once delivery of the link initiator's packet traffic has been completed.
RollCallResponses	List of the system addresses of stations intended to participate in the current multicast circuit link (multicast group members) from which valid roll call responses have been received.
RollCallTimer	Timer timing each station's participation in the roll call performed in establishing a multicast circuit link. Provides two time signals (interrupts) to the local station: one when it is time for the local station to send its own roll call response, the other when the time interval for all roll call responses by all participating stations has expired.
ScheduledSignoff	Boolean condition variable; when TRUE, causes the local station (non-master participant in a multicast circuit link) to send a TM_TERM PDU signing-off from the circuit link as soon as the roll call currently in progress is completed.
ScheduledAbort	Boolean condition variable; when TRUE, causes the local station (master of a multicast circuit link) to send a TM_TERM PDU dropping the circuit link as soon as the roll call currently in progress is completed.
UnlinkResponses	List of responses to the optional unlink roll-call performed at the conclusion of a multicast circuit connection, in which each response is in the form of an ordered pair (indAddr, ackNak), where indAddr is the address of a station whose roll call response was heard, and ackNak is the Reason field-value of the TM_TERM PDU sent as the station's response: UNL_ACK or UNL_NAK.
UnlinkRollCallTimer	Timer timing each station's participation in the optional roll call that can be performed just before a multicast circuit linkis dropped. Provides two time signals (interrupts) to the local station: one when it is time for the local station to send its own unlink roll call response (if any), the other when the time interval for all unlink roll call responses by all participating stations has expired.

C.5.3.5.4 Behavior definition.

For the reader's convenience, two equivalent representations of the behavior of the TM protocol are provided in this section: the state transition table in table C-XLI, and the state diagrams figures C-22 through C-25. Due to the complexity of the state-machine behavior, it has been found necessary to represent this behavior on four different state diagrams. Note that the Idle state shown on all four diagrams is the same single state: each diagram depicts only a subset of the transitions entering and leaving the Idle state.

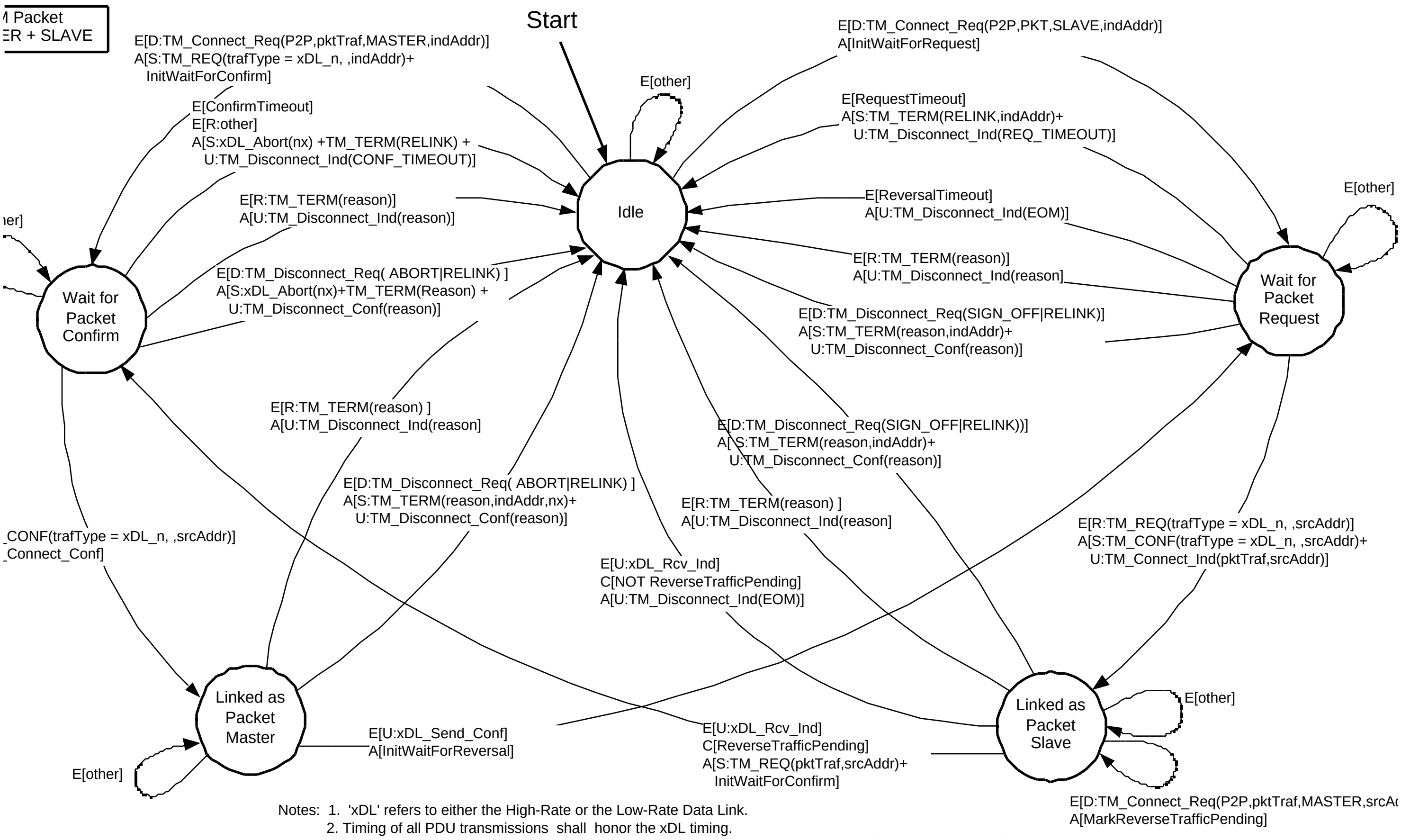
Both the state diagrams and the transition table specify the behavior of the TM entity in terms of the events defined in C.5.3.5.1 and the actions defined in C.5.3.5.2. The conditions gating certain transitions are specified in terms of the data items defined in C.5.3.5.3.

In the state diagrams, each state transition is labeled with an event, an optional condition, and zero or more actions. This indicates that the state transition occurs whenever the event occurs and the condition obtains (is TRUE), causing the associated actions to be performed. In the diagram,

- the name of each event is shown in square brackets preceded by the letter ‘E’;
- the description of each condition is shown in square brackets preceded by the letter ‘C’; and
- the names of the actions associated with a transition are shown in square brackets preceded by the letter ‘A’.

Where a transition is labeled with two or more events, this indicates that the transition occurs whenever any of the events occurs.

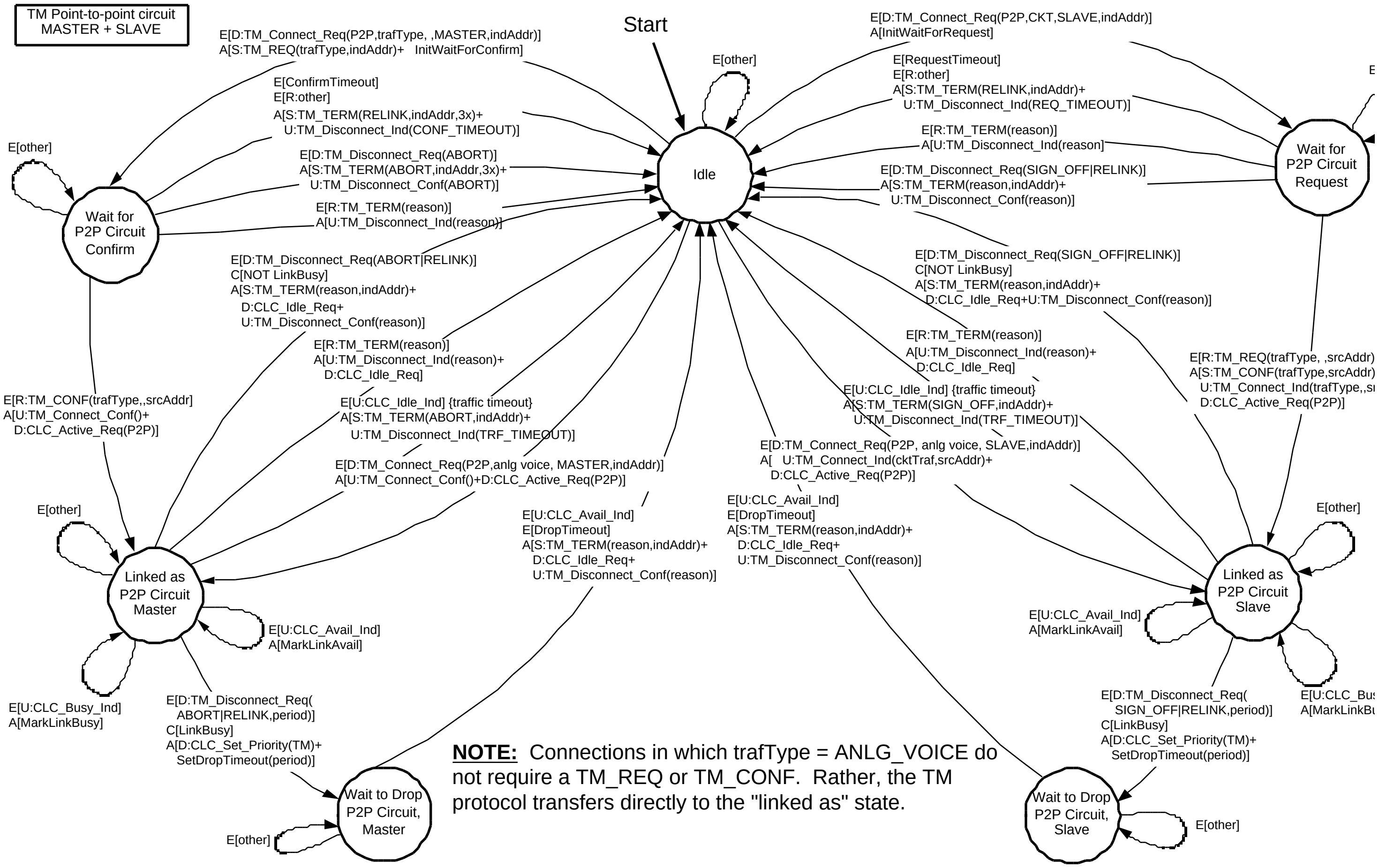
In the state diagrams and the state transition table, text within text boxes or braces (“{}”) is commentary and not part of the formal state machine definition.



/I Packet
ER + SLAVE
Start
E[D:TM_Connect_Req(P2P,pktTraf,MASTER,indAddr)]
A[S:TM_REQ(trafType = xDL_n, ,indAddr)+
InitWaitForConfirm]
E[other]
E[D:TM_Connect_Req(P2P,PKT,SLAVE,indAddr)]
A[InitWaitForRequest]
E[ConfirmTimeout]
E[R:other]
A[S:xDL_Abort(nx) +TM_TERM(RELINK) +
U:TM_Disconnect_Ind(CONF_TIMEOUT)]
E[RequestTimeout]
A[S:TM_TERM(RELINK,indAddr)+
U:TM_Disconnect_Ind(REQ_TIMEOUT)]
Idle
E[R:TM_TERM(reason)]
A[U:TM_Disconnect_Ind(reason)]
E[ReversalTimeout]
A[U:TM_Disconnect_Ind(EOM)]
E[other]
ıer]
E[R:TM_TERM(reason)]
A[U:TM_Disconnect_Ind(reason]
E[D:TM_Disconnect_Req(ABORT|RELINK)]
A[S:xDL_Abort(nx)+TM_TERM(Reason) +
U:TM_Disconnect_Conf(reason)]
Wait for
Packet
Confirm
Wait for
Packet
Request
E[D:TM_Disconnect_Req(SIGN_OFF|RELINK)]
A[S:TM_TERM(reason,indAddr)+
U:TM_Disconnect_Conf(reason)]
E[R:TM_TERM(reason)]
A[U:TM_Disconnect_Ind(reason]
E[D:TM_Disconnect_Req(SIGN_OFF|RELINK))]
A[S:TM_TERM(reason,indAddr)+
U:TM_Disconnect_Conf(reason)]
E[D:TM_Disconnect_Req(ABORT|RELINK)]
A[S:TM_TERM(reason,indAddr,nx)+
U:TM_Disconnect_Conf(reason)]
E[R:TM_TERM(reason)]
A[U:TM_Disconnect_Ind(reason]
_CONF(trafType = xDL_n, ,srcAddr)]
_Connect_Conf]
E[R:TM_REQ(trafType = xDL_n, ,srcAddr)]
A[S:TM_CONF(trafType = xDL_n, ,srcAddr)+
U:TM_Connect_Ind(pktTraf,srcAddr)]
E[U:xDL_Rcv_Ind]
C[NOT ReverseTrafficPending]
A[U:TM_Disconnect_Ind(EOM)]
Linked as
Packet
Master
E[U:xDL_Send_Conf]
A[InitWaitForReversal]
E[U:xDL_Rcv_Ind]
C[ReverseTrafficPending]
A[S:TM_REQ(pktTraf,srcAddr)+
InitWaitForConfirm]
Linked as
Packet
Slave
E[other]
E[other]
E[D:TM_Connect_Req(P2P,pktTraf,MASTER,srcA
A[MarkReverseTrafficPending]

Notes: 1. 'xDL' refers to either the High-Rate or the Low-Rate Data Link.
2. Timing of all PDU transmissions shall honor the xDL timing.
3. Aborting an xDL Packet connection shall be accomplished by an xDL_Abort PDU (sent by xDL protocol) followed by a TM_Term PDU issued by TM protocol. It is assumed that the E[D:TM_Disconnect_Req (ABORT) is issued by the user process AFTER the xDL_Abort is complete.

FIGURE C-22. TM state diagram: packet.


TM Point-to-point circuit
MASTER + SLAVE
Start
Idle
E[other]
E[D:TM_Connect_Req(P2P,trafType, ,MASTER,indAddr)]
A[S:TM_REQ(trafType,indAddr)+ InitWaitForConfirm]
E[ConfirmTimeout]
E[R:other]
A[S:TM_TERM(RELINK,indAddr,3x)+
U:TM_Disconnect_Ind(CONF_TIMEOUT)]
E[D:TM_Disconnect_Req(ABORT)]
A[S:TM_TERM(ABORT,indAddr,3x)+
U:TM_Disconnect_Conf(ABORT)]
E[R:TM_TERM(reason)]
A[U:TM_Disconnect_Ind(reason)]
E[other]
Wait for
P2P Circuit
Confirm
E[D:TM_Disconnect_Req(ABORT|RELINK)]
C[NOT LinkBusy]
A[S:TM_TERM(reason,indAddr)+
D:CLC_Idle_Req+
U:TM_Disconnect_Conf(reason)]
E[R:TM_TERM(reason)]
A[U:TM_Disconnect_Ind(reason)+
D:CLC_Idle_Req]
E[R:TM_CONF(trafType,,srcAddr]
A[U:TM_Connect_Conf()+
D:CLC_Active_Req(P2P)]
E[U:CLC_Idle_Ind] {traffic timeout}
A[S:TM_TERM(ABORT,indAddr)+
U:TM_Disconnect_Ind(TRF_TIMEOUT)]
E[D:TM_Connect_Req(P2P,anlg voice, MASTER,indAddr)]
A[U:TM_Connect_Conf()+D:CLC_Active_Req(P2P)]
E[other]
Linked as
P2P Circuit
Master
E[U:CLC_Avail_Ind]
A[MarkLinkAvail]
E[U:CLC_Busy_Ind]
A[MarkLinkBusy]
E[D:TM_Disconnect_Req(
ABORT|RELINK,period)]
C[LinkBusy]
A[D:CLC_Set_Priority(TM)+
SetDropTimeout(period)]
E[U:CLC_Avail_Ind]
E[DropTimeout]
A[S:TM_TERM(reason,indAddr)+
D:CLC_Idle_Req+
U:TM_Disconnect_Conf(reason)]
Wait to Drop
P2P Circuit,
Master
E[other]
E[D:TM_Connect_Req(P2P,CKT,SLAVE,indAddr)]
A[InitWaitForRequest]
E[RequestTimeout]
E[R:other]
A[S:TM_TERM(RELINK,indAddr)+
U:TM_Disconnect_Ind(REQ_TIMEOUT)]
E[R:TM_TERM(reason)]
A[U:TM_Disconnect_Ind(reason]
E[D:TM_Disconnect_Req(SIGN_OFF|RELINK)]
A[S:TM_TERM(reason,indAddr)+
U:TM_Disconnect_Conf(reason)]
E
Wait for
P2P Circuit
Request
E[D:TM_Disconnect_Req(SIGN_OFF|RELINK)]
C[NOT LinkBusy]
A[S:TM_TERM(reason,indAddr)+
D:CLC_Idle_Req+U:TM_Disconnect_Conf(reason)]
E[R:TM_TERM(reason)]
A[U:TM_Disconnect_Ind(reason)+
D:CLC_Idle_Req]
E[R:TM_REQ(trafType, ,srcAddr)
A[S:TM_CONF(trafType,srcAddr)
U:TM_Connect_Ind(trafType,,sr
D:CLC_Active_Req(P2P)]
E[U:CLC_Idle_Ind] {traffic timeout}
A[S:TM_TERM(SIGN_OFF,indAddr)+
U:TM_Disconnect_Ind(TRF_TIMEOUT)]
E[D:TM_Connect_Req(P2P, anlg voice, SLAVE,indAddr)]
A[U:TM_Connect_Ind(cktTraf,srcAddr)+
D:CLC_Active_Req(P2P)]
E[U:CLC_Avail_Ind]
E[DropTimeout]
A[S:TM_TERM(reason,indAddr)+
D:CLC_Idle_Req+
U:TM_Disconnect_Conf(reason)]
E[other]
Linked as
P2P Circuit
Slave
E[U:CLC_Avail_Ind]
A[MarkLinkAvail]
E[D:TM_Disconnect_Req(
SIGN_OFF|RELINK,period)]
C[LinkBusy]
A[D:CLC_Set_Priority(TM)+
SetDropTimeout(period)]
E[U:CLC_Bus
A[MarkLinkBu
Wait to Drop
P2P Circuit,
Slave
E[other]

NOTE: Connections in which trafType = ANLG_VOICE do not require a TM_REQ or TM_CONF. Rather, the TM protocol transfers directly to the "linked as" state.

FIGURE C-23. TM state diagram: point-to-point circuit.

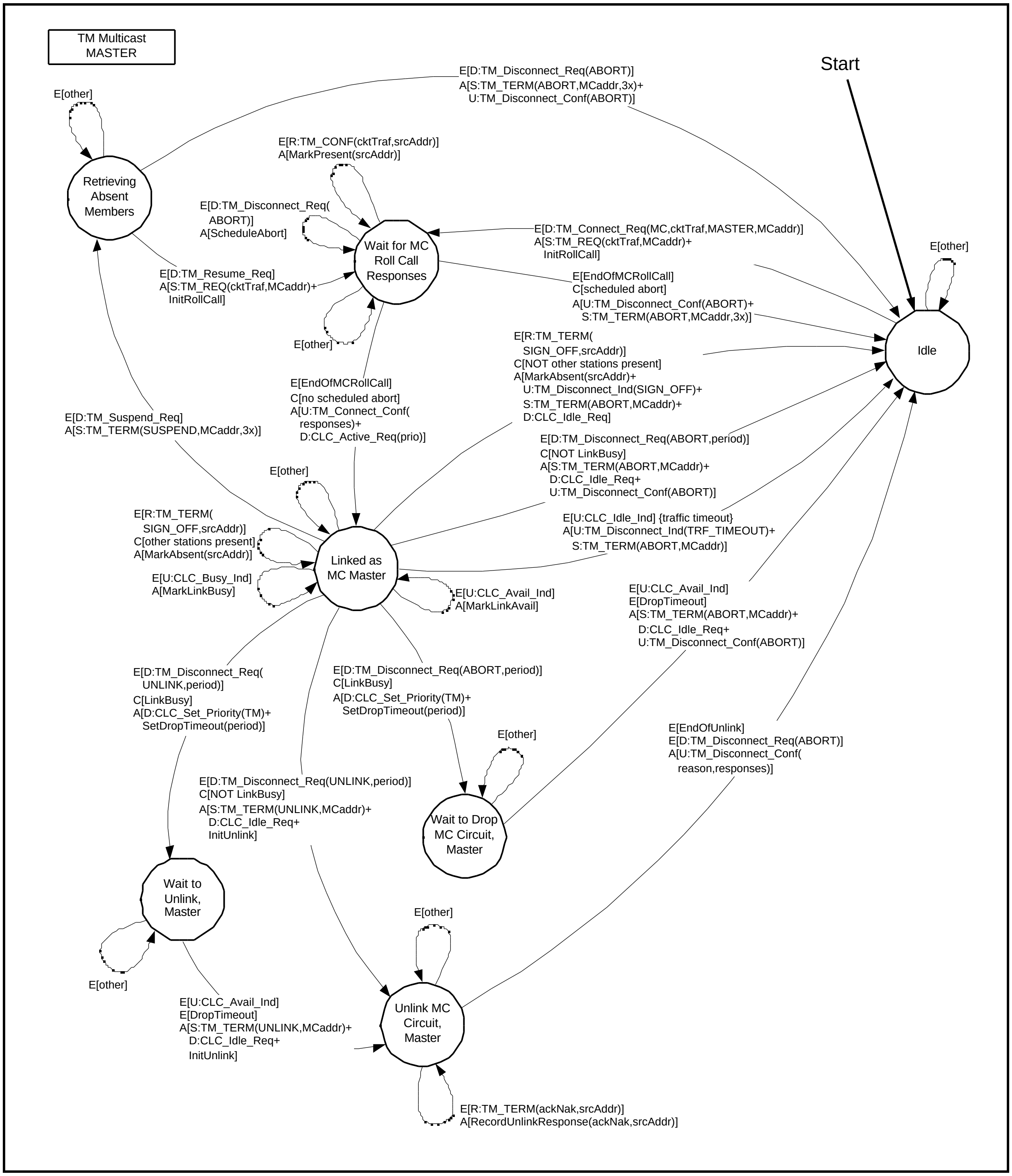


FIGURE C-24a. TM state diagram: multicast circuit (master).

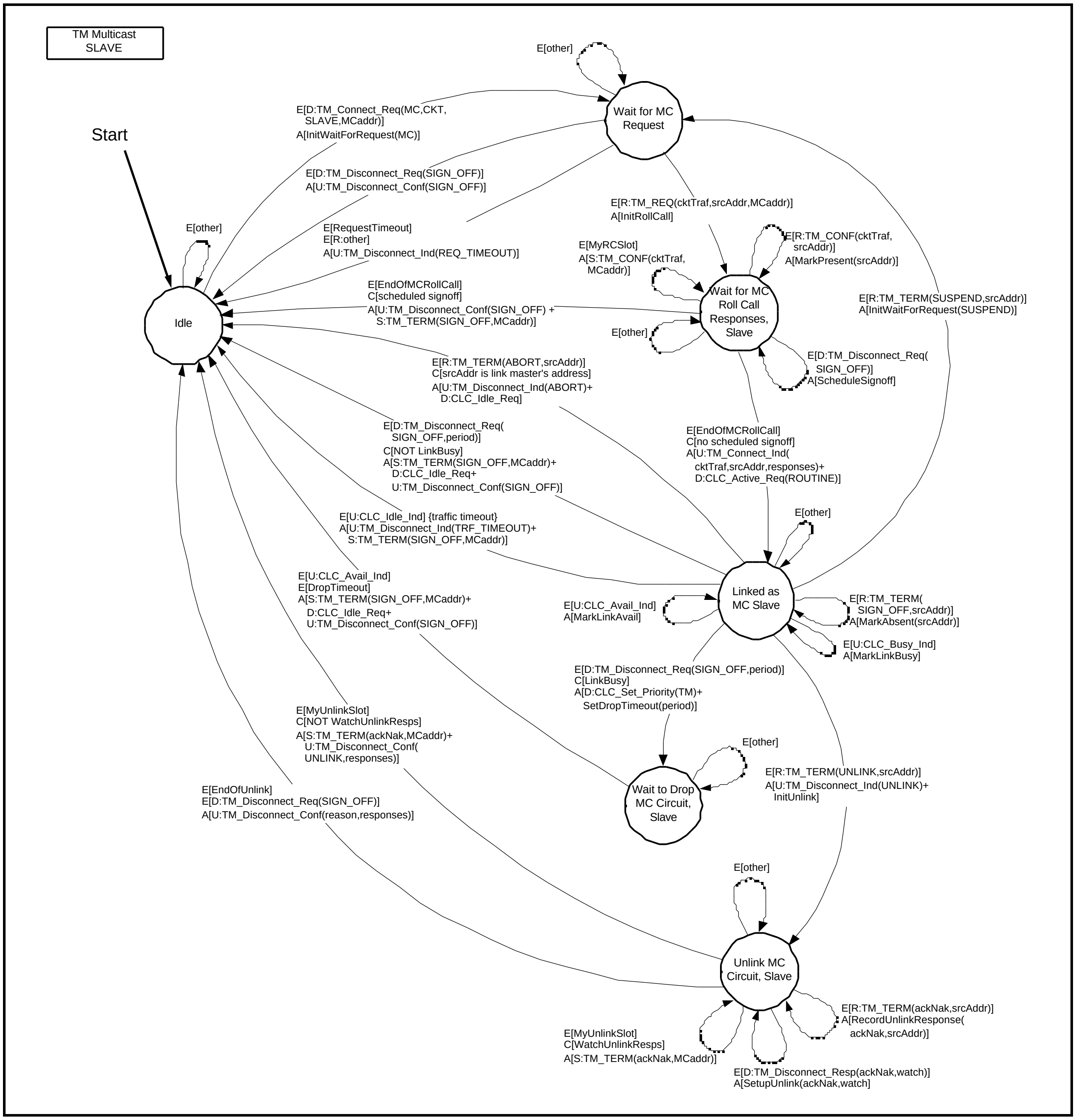

TM Multicast
SLAVE
Start
Idle
E[other]
Wait for MC
Request
E[other]
E[D:TM_Connect_Req(MC,CKT,
SLAVE,MCaddr)]
A[InitWaitForRequest(MC)]
E[D:TM_Disconnect_Req(SIGN_OFF)]
A[U:TM_Disconnect_Conf(SIGN_OFF)]
E[RequestTimeout]
E[R:other]
A[U:TM_Disconnect_Ind(REQ_TIMEOUT)]
E[R:TM_REQ(cktTraf,srcAddr,MCaddr)]
A[InitRollCall]
Wait for MC
Roll Call
Responses,
Slave
E[MyRCSlot]
A[S:TM_CONF(cktTraf,
MCaddr)]
E[R:TM_CONF(cktTraf,
srcAddr)]
A[MarkPresent(srcAddr)]
E[other]
E[D:TM_Disconnect_Req(
SIGN_OFF)]
A[ScheduleSignoff]
E[R:TM_TERM(SUSPEND,srcAddr)]
A[InitWaitForRequest(SUSPEND)]
E[EndOfMCRollCall]
C[scheduled signoff]
A[U:TM_Disconnect_Conf(SIGN_OFF) +
S:TM_TERM(SIGN_OFF,MCaddr)]
E[R:TM_TERM(ABORT,srcAddr)]
C[srcAddr is link master's address]
A[U:TM_Disconnect_Ind(ABORT)+
D:CLC_Idle_Req]
E[D:TM_Disconnect_Req(
SIGN_OFF,period)]
C[NOT LinkBusy]
A[S:TM_TERM(SIGN_OFF,MCaddr)+
D:CLC_Idle_Req+
U:TM_Disconnect_Conf(SIGN_OFF)]
E[EndOfMCRollCall]
C[no scheduled signoff]
A[U:TM_Connect_Ind(
cktTraf,srcAddr,responses)+
D:CLC_Active_Req(ROUTINE)]
E[U:CLC_Idle_Ind] {traffic timeout}
A[U:TM_Disconnect_Ind(TRF_TIMEOUT)+
S:TM_TERM(SIGN_OFF,MCaddr)]
E[other]
Linked as
MC Slave
E[U:CLC_Avail_Ind]
E[DropTimeout]
A[S:TM_TERM(SIGN_OFF,MCaddr)+
D:CLC_Idle_Req+
U:TM_Disconnect_Conf(SIGN_OFF)]
E[U:CLC_Avail_Ind]
A[MarkLinkAvail]
E[R:TM_TERM(
SIGN_OFF,srcAddr)]
A[MarkAbsent(srcAddr)]
E[U:CLC_Busy_Ind]
A[MarkLinkBusy]
E[D:TM_Disconnect_Req(SIGN_OFF,period)]
C[LinkBusy]
A[D:CLC_Set_Priority(TM)+
SetDropTimeout(period)]
E[MyUnlinkSlot]
C[NOT WatchUnlinkResps]
A[S:TM_TERM(ackNak,MCaddr)+
U:TM_Disconnect_Conf(
UNLINK,responses)]
E[other]
Wait to Drop
MC Circuit,
Slave
E[R:TM_TERM(UNLINK,srcAddr)]
A[U:TM_Disconnect_Ind(UNLINK)+
InitUnlink]
E[EndOfUnlink]
E[D:TM_Disconnect_Req(SIGN_OFF)]
A[U:TM_Disconnect_Conf(reason,responses)]
E[other]
Unlink MC
Circuit, Slave
E[R:TM_TERM(ackNak,srcAddr)]
A[RecordUnlinkResponse(
ackNak,srcAddr)]
E[MyUnlinkSlot]
C[WatchUnlinkResps]
A[S:TM_TERM(ackNak,MCaddr)]
E[D:TM_Disconnect_Resp(ackNak,watch)]
A[SetupUnlink(ackNak,watch)]

FIGURE C-24b. TM state diagram: multicast circuit (slave).

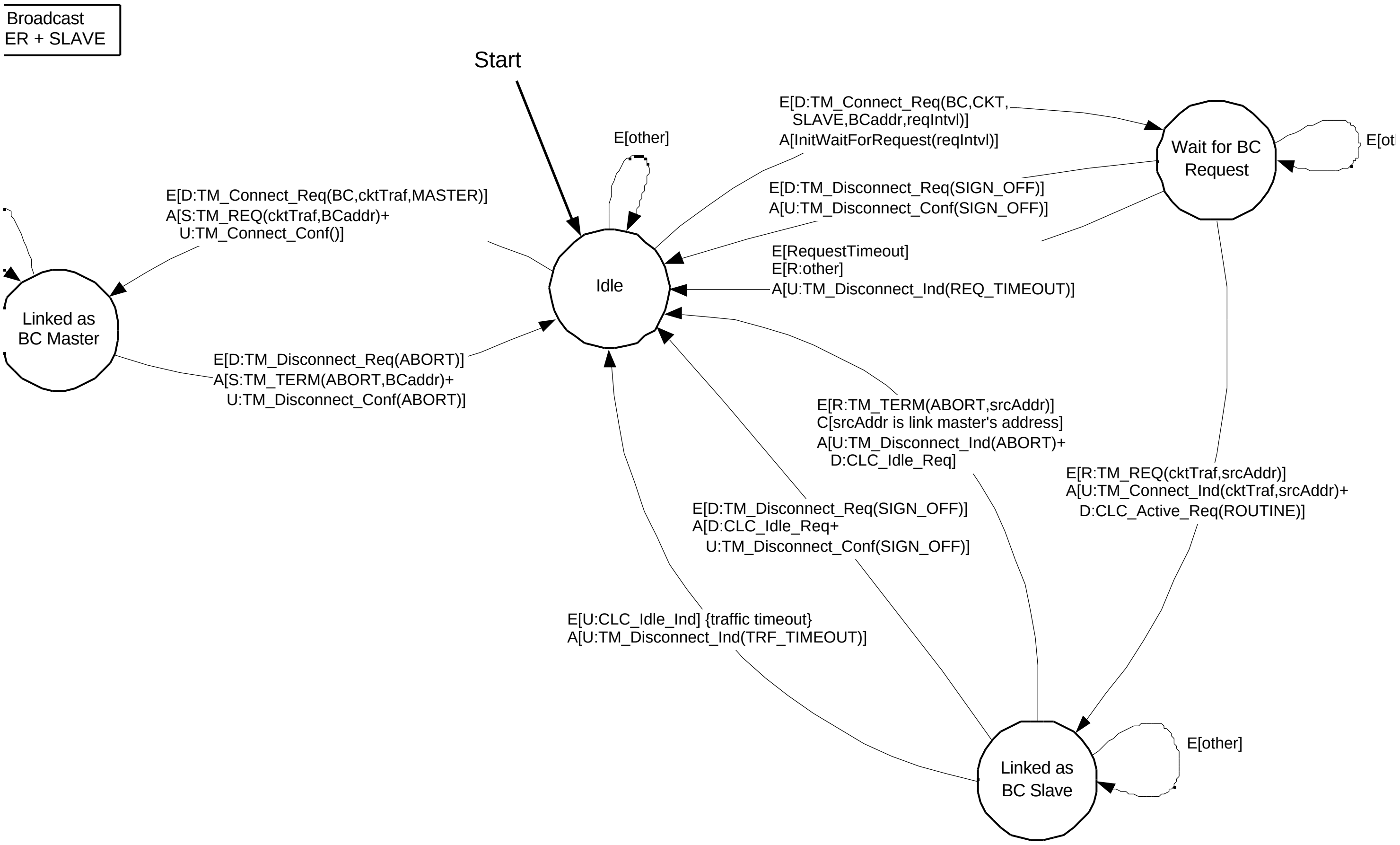

Broadcast
ER + SLAVE
Start
Idle
E[other]
Wait for BC Request
E[ot
Linked as BC Master
Linked as BC Slave
E[other]
E[D:TM_Connect_Req(BC,CKT, SLAVE,BCaddr,reqIntvl)]
A[InitWaitForRequest(reqIntvl)]
E[D:TM_Connect_Req(BC,cktTraf,MASTER)]
A[S:TM_REQ(cktTraf,BCaddr)+ U:TM_Connect_Conf()]
E[D:TM_Disconnect_Req(SIGN_OFF)]
A[U:TM_Disconnect_Conf(SIGN_OFF)]
E[RequestTimeout]
E[R:other]
A[U:TM_Disconnect_Ind(REQ_TIMEOUT)]
E[D:TM_Disconnect_Req(ABORT)]
A[S:TM_TERM(ABORT,BCaddr)+ U:TM_Disconnect_Conf(ABORT)]
E[R:TM_TERM(ABORT,srcAddr)]
C[srcAddr is link master's address]
A[U:TM_Disconnect_Ind(ABORT)+ D:CLC_Idle_Req]
E[R:TM_REQ(cktTraf,srcAddr)]
A[U:TM_Connect_Ind(cktTraf,srcAddr)+ D:CLC_Active_Req(ROUTINE)]
E[D:TM_Disconnect_Req(SIGN_OFF)]
A[D:CLC_Idle_Req+ U:TM_Disconnect_Conf(SIGN_OFF)]
E[U:CLC_Idle_Ind] {traffic timeout}
A[U:TM_Disconnect_Ind(TRF_TIMEOUT)]

FIGURE C-25. TM state diagram: broadcast circuit.

TABLE C-XLI. TM state transition table.

State	Event	Condition	Action	Next State
Idle	D:TM_Connect_Req (P2P, pktTraf, MASTER, indAddr)		S:TM_REQ (pktTraf, indAddr)+ InitWaitForConfirm	Wait for Packet Confirm
	D:TM_Connect_Req (P2P, PKT, SLAVE, indAddr)		InitWaitForRequest	Wait for Packet Request
	D:TM_Connect_Req (P2P, cktTraf, MASTER, indAddr)		S:TM_REQ (cktTraf, indAddr)+ InitWaitForConfirm	Wait for P2P Circuit Confirm
	D:TM_Connect_Req (P2P, CKT, SLAVE, indAddr)		InitWaitForRequest	Wait for P2P Circuit Request
	D:TM_Connect_Req (P2P, ANLG_VOICE, SLAVE, indAddr)		U:TM_Connect_Ind (cktTraf, srcAddr)	Linked as P2P circuit slave
	D:TM_Connect_Req (MC, cktTraf, MASTER, MCaddr)		S:TM_REQ (cktTraf, MCaddr)+ InitRollCall	Wait for MC Roll Call Responses
	D:TM_Connect_Req (MC, ANLG_VOICE, MASTER, MCaddr)		U:TM_Connect_Ind (cktTraf, srcAddr)	Linked as P2P circuit master
	D:TM_Connect_Req (MC, CKT, SLAVE, MCaddr)		InitWaitForRequest(MC)	Wait for MC Request
	D:TM_Connect_Req (BC, cktTraf, MASTER)		S:TM_REQ (cktTraf, BCaddr)+ U:TM_Connect_Conf	Linked as BC Master
	D:TM_Connect_Req (BC, CKT, SLAVE, BCaddr reqIntvl)		InitWaitForRequest (reqIntvl)	Wait for BC Request
	other		none	Idle
Wait for Packet Confirm	R:TM_CONF (pktTraf, srcAddr)		U:TM_Connect_Conf	Linked as Packet Master
	ConfirmTimeout R:other		S:TM_TERM (RELINK, indAddr, nx)+ U:TM_Disconnect_Ind (CONF_TIMEOUT)	Idle
	D:TM_Disconnect_Req (ABORT | RELINK)		S:TM_TERM (reason, indAddr, nx)+ U:TM_Disconnect_Conf (reason)	Idle
	R:TM_TERM (reason)		U:TM_Disconnect_Ind (reason)	Idle
	other		none	Wait for Packet Confirm
Linked as Packet Master	U:xDL_Send_Conf		InitWaitForReversal	Wait for Packet Request
	D:TM_Disconnect_Req (ABORT | RELINK)		S:TM_TERM (reason, indAddr, nx)+ U:TM_Disconnect_Conf (reason)	Idle
	R:TM_TERM (reason)		U:TM_Disconnect_Ind (reason)	Idle
	other		none	Linked as Packet Master
Wait for Packet Request	R:TM_REQ (pktTraf, srcAddr)		S:TM_CONF (pktTraf, srcAddr)+ U:TM_Connect_Ind (pktTraf, srcAddr)	Linked as Packet Slave
	D:TM_Disconnect_Req (SIGN_OFF | RELINK)		S:TM_TERM (reason, indAddr)+ U:TM_Disconnect_Conf (reason)	Idle
	R:TM_TERM (reason)		U:TM_Disconnect_Ind (reason)	Idle
	RequestTimeout		S:TM_TERM (RELINK, indAddr)+ U:TM_Disconnect_Ind (REQ_TIMEOUT)	Idle
	ReversalTimeout		U:TM_Disconnect_Ind (EOM)	Idle
	other		none	Wait for Packet Request

TABLE C-XLI. TM state transition table (continued).

State	Event	Condition	Action	Next State
Linked as Packet Slave	U:xDL_Rcv_Ind	NOT Reverse Traffic Pending	U:TM_Disconnect_Ind (EOM)	Idle
	U:xDL_Rcv_Ind	Reverse Traffic Pending	S:TM_REQ (pktTraf, srcAddr)+ InitWaitForConfirm	Wait for Packet Confirm
	D:TM_Disconnect_Req (SIGN_OFF | RELINK)		S:TM_TERM (reason, indAddr)+ U:TM_Disconnect_Conf (reason)	Idle
	R:TM_TERM (reason)		U:TM_Disconnect_Ind (reason)	Idle
	D:TM_Connect_Req (P2P, pktTraf, MASTER, srcAddr)		MarkReverseTrafficPending	Linked as Packet Slave
	other		none	Linked as Packet Slave
Wait for P2P Circuit Confirm	R:TM_CONF (cktTraf, srcAddr)		U:TM_Connect_Conf+ D:CLC_Active_Req (P2P)	Linked as P2P Circuit Master
	D:TM_Disconnect_Req (ABORT)		S:TM_TERM (ABORT, indAddr, 3x)+ U:TM_Disconnect_Conf (ABORT)	Idle
	R:TM_TERM (reason)		U:TM_Disconnect_Ind (reason)	Idle
	ConfirmTimeout R:other		S:TM_TERM (RELINK, indAddr, 3x)+ U:TM_Disconnect_Ind (CONF_TIMEOUT)	Idle
	other		none	Wait for P2P Circuit Confirm
Linked as P2P Circuit Master	R:TM_TERM (reason)		U:TM_Disconnect_Ind (reason)+ D:CLC_Idle_Req	Idle
	D:TM_Disconnect_Req (ABORT | RELINK)	NOT LinkBusy	S:TM_TERM (reason, indAddr)+ D:CLC_Idle_Req+ U:TM_Disconnect_Conf (reason)	Idle
	D:TM_Disconnect_Req (ABORT | RELINK, period)	LinkBusy	D:CLC_Set_Priority (TM)+ SetDropTimeout (period)	Wait to Drop P2P Circuit, Master
	U:CLC_Idle_Ind {traffic timeout}		S:TM_TERM (ABORT, indAddr)+ U:TM_Disconnect_Ind (TRF_TIMEOUT)	Idle
	U:CLC_Busy_Ind		MarkLinkBusy	Linked as P2P Circuit Master
	U:CLC_Avail_Ind		MarkLinkAvail	Linked as P2P Circuit Master
	other		none	Linked as P2P Circuit Master
Wait to Drop P2P Circuit, Master	U:CLC_Avail_Ind DropTimeout		S:TM_TERM (reason, indAddr)+ D:CLC_Idle_Req+ U:TM_Disconnect_Conf (reason)	Idle
	other		none	Wait to Drop P2P Circuit, Master

TABLE C-XLI. TM state transition table (continued).

State	Event	Condition	Action	Next State
Wait for P2P Circuit Request	R:TM_REQ (cktTraf, srcAddr)		S:TM_CONF (cktTraf, srcAddr)+ U:TM_Connect_Ind (cktTraf, srcAddr)+ D:CLC_Active_Req (P2P)	Linked as P2P Circuit Slave
	RequestTimeout R:other		S:TM_TERM (RELINK, indAddr)+ U:TM_Disconnect_Ind (REQ_TIMEOUT)	Idle
	D:TM_Disconnect_Req (SIGN_OFF | RELINK)		S:TM_TERM (reason, indAddr)+ U:TM_Disconnect_Conf (reason)	Idle
	R:TM_TERM (reason)		U:TM_Disconnect_Ind (reason)	Idle
	other		none	Wait for P2P Circuit Request
Linked as P2P Circuit Slave	D:TM_Disconnect_Req (SIGN_OFF | RELINK)	NOT LinkBusy	S:TM_TERM (reason, indAddr)+ D:CLC_Idle_Req+ U:TM_Disconnect_Conf (reason)	Idle
	R:TM_TERM (reason)		U:TM_Disconnect_Ind (reason)+ D:CLC_Idle_Req	Idle
	U:CLC_Idle_Ind {traffic timeout}		S:TM_TERM (SIGN_OFF, indAddr)+ U:TM_Disconnect_Ind (TRF_TIMEOUT)	Idle
	D:TM_Disconnect_Req (SIGN_OFF | RELINK, period)	LinkBusy	D:CLC_Set_Priority (TM)+ SetDropTimeout (period)	Wait to Drop P2P Circuit, Slave
	U:CLC_Busy_Ind		MarkLinkBusy	Linked as P2P Circuit Slave
	U:CLC_Avail_Ind		MarkLinkAvail	Linked as P2P Circuit Slave
	other		none	Linked as P2P Circuit Slave
Wait to Drop P2P Circuit, Slave	U:CLC_Avail_Ind DropTimeout		S:TM_TERM (reason, indAddr)+ D:CLC_Idle_Req+ U:TM_Disconnect_Conf (reason)	Idle
	other		none	Wait to Drop P2P Circuit, Slave
Wait for MC Roll Call Responses	EndOfMCRollCall	NOT Scheduled Abort	U:TM_Connect_Conf (responses)+ D:CLC_Active_Req (prio)	Linked as MC Master
	R:TM_CONF (cktTraf, srcAddr)		MarkPresent (srcAddr)	Wait for MC Roll Call Responses
	D:TM_Disconnect_Req (ABORT)		ScheduleAbort	Wait for MC Roll Call Responses
	other		none	Wait for MC Roll Call Responses
	EndOfMCRollCall	Scheduled Abort	U:TM_Disconnect_Conf (ABORT)+ S:TM_TERM (ABORT, MCaddr, 3x)	Idle

TABLE C-XLI. TM state transition table (continued).

State	Event	Condition	Action	Next State
Linked as MC Master	R:TM_TERM (SIGN_OFF, srcAddr)	other stations present	MarkAbsent (srcAddr)	Linked as MC Master
	R:TM_TERM (SIGN_OFF, srcAddr)	NOT other stations present	MarkAbsent (srcAddr)+ U:TM_Disconnect_Ind (SIGN_OFF)+ S:TM_TERM (ABORT, MCaddr)+ D:CLC_Idle_Req	Idle
	D:TM_Disconnect_Req (ABORT, period)	NOT LinkBusy	S:TM_TERM (ABORT, MCaddr)+ D:CLC_Idle_Req+ U:TM_Disconnect_Conf (ABORT)	Idle
	D:TM_Disconnect_Req (ABORT, period)	LinkBusy	D:CLC_Set_Priority (TM)+ SetDropTimeout(period)	Wait to Drop MC Circuit, Master
	U:CLC_Idle_Ind {traffic timeout}		U:TM_Disconnect_Ind (TRF_TIMEOUT)+ S:TM_TERM (ABORT, MCaddr)	Idle
	U:CLC_Busy_Ind		MarkLinkBusy	Linked as MC Master
	U:CLC_Avail_Ind		MarkLinkAvail	Linked as MC Master
	D:TM_Suspend_Req		S:TM_TERM (SUSPEND, MCaddr, 3x)	Retrieving Absent Members
	D:TM_Disconnect_Req (UNLINK, period)	NOT LinkBusy	S:TM_TERM (UNLINK, MCaddr)+ D:CLC_Idle_Req+ InitUnlink	Unlink MC Circuit, Master
	D:TM_Disconnect_Req (UNLINK, period)	LinkBusy	D:CLC_Set_Priority (TM)+ SetDropTimeout (period)	Wait to Unlink, Master
	other		none	Linked as MC Master
Retrieve Absent Members	D:TM_Resume_Req		S:TM_REQ (cktTraf, MCaddr)+ InitRollCall	Wait for MC Roll Call Responses
	D:TM_Disconnect_Req (ABORT)		S:TM_TERM (ABORT, MCaddr, 3x)+ U:TM_Disconnect_Conf (ABORT)	Idle
	other		none	Retrieve Absent Members
Wait to Drop MC Circuit, Master	U:CLC_Avail_Ind DropTimeout		S:TM_TERM (ABORT, MCaddr)+ D:CLC_Idle_Req+ U:TM_Disconnect_Conf (ABORT)	Idle
	other		none	Wait to Drop MC Circuit, Master
Unlink MC Circuit, Master	R:TM_TERM (ackNak, srcAddr)		RecordUnlinkResponse (ackNak, srcAddr)	Unlink MC Circuit, Master
	EndOfUnlink D:TM_Disconnect_Req (ABORT)		U:TM_Disconnect_Conf (reason, responses)	Idle
	other		none	Unlink MC Circuit, Master
Wait to Unlink, Master	U:CLC_Avail_Ind DropTimeout		S:TM_TERM (UNLINK, MCaddr)+ D:CLC_Idle_Req+ InitUnlink	Unlink MC Circuit, Master
	other		none	Wait to Unlink, Master

TABLE C-XLI. TM state transition table (continued).

State	Event	Condition	Action	Next State
Wait for MC Request	R:TM_REQ (cktTraf, srcAddr, MCaddr)		InitRollCall	Wait for MC Roll Call Responses, Slave
	RequestTimeout R:other		U:TM_Disconnect_Ind (REQ_TIMEOUT)	Idle
	D:TM_Disconnect_Req (SIGN_OFF)		U:TM_Disconnect_Conf (SIGN_OFF)	Idle
	other		none	Wait for MC Request
Wait for MC Roll Call Respons es, Slave	EndOfMCRollCall	NOT Scheduled Signoff	U:TM_Connect_Ind (cktTraf, srcAddr, responses)+ D:CLC_Active_Req (ROUTINE)	Linked as MC Slave
	R:TM_CONF (cktTraf, srcAddr)		MarkPresent (srcAddr)	Wait for MC Roll Call Responses, Slave
	MyRCSlot		S:TM_CONF (cktTraf, MCaddr)	Wait for MC Roll Call Responses, Slave
	D:TM_Disconnect_Req (SIGN_OFF)		ScheduleSignoff	Wait for MC Roll Call Responses, Slave
	other		none	Wait for MC Roll Call Responses, Slave
	EndOfMCRollCall	Scheduled Signoff	U:TM_Disconnect_Conf (SIGN_OFF)+ S:TM_TERM (SIGN_OFF, MCaddr)	Idle
Linked as MC Slave	R:TM_TERM (SIGN_OFF, srcAddr)		MarkAbsent (srcAddr)	Linked as MC Slave
	D:TM_Disconnect_Req (SIGN_OFF, period)	NOT LinkBusy	S:TM_TERM (SIGN_OFF, MCaddr)+ D:CLC_Idle_Req+ U:TM_Disconnect_Conf (SIGN_OFF)	Idle
	D:TM_Disconnect_Req (SIGN_OFF, period)	LinkBusy	D:CLC_Set_Priority (TM)+ SetDropTimeout (period)	Wait to Drop MC Circuit, Slave
	R:TM_TERM (ABORT, srcAddr)	srcAddr is link master's address	U:TM_Disconnect_Ind (ABORT)+ D:CLC_Idle_Req	Idle
	U:CLC_Idle_Ind {traffic timeout}		U:TM_Disconnect_Ind (TRF_TIMEOUT) + S:TM_TERM (SIGN_OFF, MCaddr)	Idle
	U:CLC_Busy_Ind		MarkLinkBusy	Linked as MC Slave
	U:CLC_Avail_Ind		MarkLinkAvail	Linked as MC Slave
	R:TM_TERM (SUSPEND, srcAddr)		InitWaitForRequest(SUSPEND)	Wait for MC Request
	R:TM_TERM (UNLINK, srcAddr)		U:TM_Disconnect_Ind (UNLINK)+ InitSlaveUnlink	Unlink MC Circuit, Slave
	other		none	Linked as MC Slave
Wait to Drop MC Circuit, Slave	U:CLC_Avail_Ind DropTimeout		S:TM_TERM (SIGN_OFF, MCaddr)+ D:CLC_Idle_Req+ U:TM_Disconnect_Conf (SIGN_OFF)	Idle
	other		none	Wait to Drop MC Circuit, Slave

TABLE C-XLI. TM state transition table (continued).

State	Event	Condition	Action	Next State
Unlink MC Circuit, Slave	D:TM_Disconnect_Resp (ackNak, watch)		SetupUnlink (ackNak, watch)	Unlink MC Circuit, Slave
	R:TM_TERM (ackNak, srcAddr)		RecordUnlinkResponse (ackNak, srcAddr)	Unlink MC Circuit, Slave
	MyUnlinkSlot	WatchUnlink Resps	S:TM_TERM (ackNak, MCaddr)	Unlink MC Circuit, Slave
	MyUnlinkSlot	NOT Watch UnlinkResps	S:TM_TERM (ackNak, MCaddr)+ U:TM_Disconnect_Conf (UNLINK, responses)	Idle
	EndOfUnlink D:TM_Disconnect_Req (SIGN_OFF)		U:TM_Disconnect_Conf (reason, responses)	Idle
	other		none	Unlink MC Circuit, Slave
Linked as BC Master	D:TM_Disconnect_Req (ABORT)		S:TM_TERM (ABORT, BCaddr)+ U:TM_Disconnect_Conf (ABORT)	Idle
	other		none	Linked as BC Master
Wait for BC Request	R:TM_REQ (cktTraf, srcAddr)		U:TM_Connect_Ind (cktTraf, srcAddr)+ D:CLC_Active_Req (ROUTINE) {CLC used only for traffic timeout}	Linked as BC Slave
	RequestTimeout R:other		U:TM_Disconnect_Ind (REQ_TIMEOUT)	Idle
	D:TM_Disconnect_Req (SIGN_OFF)		U:TM_Disconnect_Conf (SIGN_OFF)	Idle
	other		none	Wait for BC Request
Linked as BC Slave	D:TM_Disconnect_Req (SIGN_OFF)		D:CLC_Idle_Req+ U:TM_Disconnect_Conf (SIGN_OFF)	Idle
	R:TM_TERM (ABORT, srcAddr)	srcAddr is link master's address	U:TM_Disconnect_Ind (ABORT)+ D:CLC_Idle_Req	Idle
	U:CLC_Idle_Ind (traffic timeout}		U:TM_Disconnect_Ind (TRF_TIMEOUT)	Idle
	other		none	Linked as BC Slave

C.5.3.5.5 Timing characteristics.

The protocol timing characteristics vary depending on which kind of traffic link is being established. The sub-sections of this section describe the timing characteristics applying to establishment and execution of, respectively, point-to-point packet traffic links, point-to-point circuit links, and multicast circuit links.

Table C-XLII gives definitions of time intervals used in presenting the protocol timing characteristics.

TABLE C-XLII. Protocol time-intervals.

Interval	# 32 Frames	# 48 Frames	# PSK Symbol times	Duration (ms., approx.)	Description
T_{slot}		45	2160	900	Duration of 3G-ALE slot.
T_{dwell}		270	12960	5400	Duration of 3G-ALE dwell.
T_{sug}	8		256	106.67	LE sync uncertainty guard interval
T_{tune}	45		1600	666.67	TM coupler tune allowance
$T_{prop,max}$	6		192	80.00	maximum propagation delay
$T_{BW0proc}$	8		128	53.33	BW0 processing time (after last sample is received)
T_{BW1enc}	5		160	66.67	BW1 encoding time (prior to emission of first sample)
T_{BW1}	98		3136	1306.67	BW1 transmission duration (TLC+preamble+data)
$T_{BW1proc}$	10		320	133.33	BW1 processing time (after last sample is received)
T_{BW2enc}	22		704	293.33	BW2 encoding time (prior to emission of first sample)
T_{BW2}	2	5+20n	304+960n	126.67+ (n*400.00)	BW2 transmission duration -- n packets per transmission (n = 3, 6, 12, or 24)
$T_{BW2(3)}$	2	65	3184	1326.67	BW2 transmission duration (3 packets per frame)
$T_{BW2(6)}$	2	125	6064	2526.67	BW2 transmission duration (6 packets per frame)
$T_{BW2(12)}$	2	245	11824	4926.67	BW2 transmission duration (12 packets per frame)
$T_{BW2(24)}$	2	485	23344	9726.67	BW2 transmission duration (24 packets per frame)
$T_{BW2proc}$		11	528	220.00	BW2 processing time (after last sample received)
T_{BW3enc}	5		160	66.67	BW3 encoding time (prior to emission of first sample)
T_{BW3}	28+n		896+32n	373.33+ (n*13.33)	BW3 transmission duration (preamble+data) – n payload bytes per transmission (n = 64, 128, 256, or 512)
$T_{BW3(64)}$	92		2944	1226.67	BW3 transmission duration (preamble+data) – 64 payload bytes per transmission
$T_{BW3(128)}$	156		4992	2080.00	BW3 transmission duration (preamble+data) – 128 payload bytes per transmission
$T_{BW3(256)}$	284		9088	3786.67	BW3 transmission duration (preamble+data) – 256 payload bytes per transmission
$T_{BW3(512)}$	540		17280	7200.00	BW3 transmission duration (preamble+data) – 512 payload bytes per transmission
$T_{BW3proc}$	5		160	66.67	BW3 processing time (after last sample is received)
T_{BW4enc}	5		160	66.67	BW4 encoding time (prior to emission of first sample)
T_{BW4}	48		1536	640.00	BW4 transmission duration (TLC+data)
$T_{BW4proc}$	5		160	66.67	BW4 processing time (after last sample is received)

TABLE C-XLII. Protocol time-intervals (continued).

Interval	# 32 Frames	# 48 Frames	# PSK Symbol times	Duration (ms., approx.)	Description
$T_{TM,Master}$					Duration of the initial TM handshake at the link master station. $T_{TM,Master} = T_{sug} + 2T_{BW1} + 2T_{prop,max} + 2T_{BW1proc} + T_{BW1enc.}$
$T_{RTFD, HDL}$					Time from start of TM_REQUEST PDU to start of first HDL_DATA PDU. $T_{RTFD, HDL} = 2T_{BW1} + 2T_{prop,max} + 2T_{BW1proc} + T_{BW1enc} + T_{BW2enc}$.
$T_{CTFA(n), HDL}$					Time from start of TM_CONFIRM PDU to start of first HDL_ACK PDU. $T_{CTFA, HDL} = T_{BW1} + T_{BW1proc} + 2T_{prop,max} + T_{BW2enc} + T_{BW2(n)} + T_{BW2proc} + T_{BW1enc}$.
$T_{HDL(n)}$					Period of HDL protocol. $T_{HDL(n)} = T_{BW2enc} + T_{BW2(n)} + 2T_{prop,max} + T_{BW2proc} + T_{BW1enc} + T_{BW1} + T_{BW1proc}$.
$T_{RTFD, LDL}$					Time from start of TM_REQUEST PDU to start of first LDL_DATA PDU. $T_{RTFD, LDL} = 2T_{BW1} + 2T_{prop,max} + 2T_{BW1proc} + T_{BW1enc} + T_{BW3enc}$.
$T_{CTFA(n), LDL}$					Time from start of TM_CONFIRM PDU to start of first LDL_ACK PDU. $T_{CTFA, LDL(n)} = T_{BW1} + T_{BW1proc} + 2T_{prop,max} + T_{BW3enc} + T_{BW3(n)} + T_{BW3proc} + T_{BW4enc}$.
$T_{LDL(n)}$					Period of LDL protocol. $T_{LDL(n)} = T_{BW3enc} + T_{BW3(n)} + 2T_{prop,max} + T_{BW3proc} + T_{BW4enc} + T_{BW4} + T_{BW4proc}$.
$T_{rc\ slot,TM}$					Duration of TM roll call slot. $T_{rc\ slot,TM} = T_{BW1enc} + T_{BW1} + 2{*}T_{prop,max} + T_{BW1proc}$
T_{CLenc}	15		2400	1000.00	MIL-STD-188-110 serial tone (see note) modem transmit startup delay: the permitted time interval between presentation of the first bit of data to the modem for modulation (when RTS is asserted), and emission of the first time-sample of the modem preamble. Note that this delay is not required for encoding per se, since MIL-STD-188-110 modems typically start emitting the modem preamble simultaneously with encoding the first bit of traffic data. The modem startup delay defined here is a characteristic of modem implementations rather than of the MIL-STD-188-110 standard.
T_{CLpre}	15		480	200.00	Duration of a single period of the Mil-Std-188-110A serial tone modem preamble: the minimum portion of the modem preamble that must be received and processed to successfully detect and acquire sync with an incoming modem transmission.
T_{CLproc}	15		480	200.00	Mil-Std-188-110A serial tone modem acquisition processing delay: the processing time required following receipt of the last sample of a 200 ms. Preamble period, before the receiving modem can declare modem signal presence based on having acquired the preamble.
NOTE: Timing analyses for circuit links assume that these links are used for bit-pipe delivery of data using MIL-STD-188-110 serial tone modems; the timings used for delivery of other kinds of traffic on circuit links are the same.					

C.5.3.5.5.1 Point-to-point packet link.

This section will first provide a description of point-to-point packet link timing. Following this, point-to-point packet link timing requirements will be given.

C.5.3.5.5.1.1 Point-to-point packet link timing description.

The contents of this section are for informational purposes only.

The TM phase of the point-to-point packet link is considered to begin at the end of the 3G-ALE time-slot in which the LE_COMMENCE PDU (see note) is transmitted. Since only a two-way TM handshake is performed, it is not possible for both stations to estimate the propagation delay between them. Instead, in each direction, the TM handshake signalling is used to establish the timing for all subsequent signalling in that direction. In the forward direction, the first xDL_DATA PDU (High-Rate or Low-Rate) is sent at a fixed time interval known a priori to both stations, following the transmission of the TM_REQUEST PDU. Likewise, in the reverse direction, the first xDL_ACK PDU is sent at a fixed time interval known a priori to both stations, following the transmission of the TM_CONFIRM PDU. The entire process is depicted on figure C-26 through figure C-30, and analyzed in further detail below, using the HDL protocol for the purpose of illustration (i.e., an LE_HANDSHAKE PDU in which the Command field's value is "Commence Traffic").

The linking activity proceeds as follows:

- First, an LE handshake is performed. This handshake establishes a link time reference, T_{link}, for both the Master and the Slave. This time reference is defined as the start of the LE slot immediately following the LE slot in which the LE_COMMENCE PDU was transmitted. If TOD offset exists between the Master and the Slave (i.e. $T_{deltaTOD} \neq 0$), T_{link} will not be the same for Master and Slave, and so we introduce individual T_{link} values for each station, $T_{link,Master}$ and $T_{link,Slave}$. Figure C-27 through figure C-30 show examples of $T_{deltaTOD}$ having a non-zero value, and thus $T_{link,Master} \neq T_{link,Slave}$. $T_{link,Master}$ and $T_{link,Slave}$ can differ by no more than T_{sug}.
- Next, Master and Slave are given an opportunity to change to the traffic channel and tune, if necessary. This opportunity lasts T_{tune} seconds.
- Next, a TM handshake is performed. As was done for LE timing, the Master begins emission of the TM_REQUEST PDU T_{sug} seconds into the TM time slot. (The reason for this is shown on figure C-30.) Unlike CM handshakes, the response (in this case the TM_CONFIRM) is always emitted a fixed duration after reception of the first TM PDU of the handshake. As shown in the figures, this fixed response latency is $T_{BW1proc} + T_{BW1enc}$. Combining this fixed response latency with the duration of the two BW1 waveforms, a second processing delay for the Master to process the response, the sync uncertainty guard time, and worst-case propagation delay gives the duration of $T_{TM,Master}$ as defined in table C-XLI and as shown in the figures. Note that $T_{TM,Master}$ is fixed in duration, but $T_{TM,Slave}$ is not. $T_{TM,Slave}$ is equal to $T_{TM,Master} - T_{deltaTOD} + T_{prop}$. The fact that $T_{TM,Slave}$ is variable does not complicate matters for the Slave station,

however, since the TM_REQUEST and the first forward transmission of the data link protocol are separated by a fixed amount of time, $T_{RTFD, HDL}$. As a result, the Slave need only measure the time of arrival of the TM_REQUEST PDU to know when to expect the first forward transmission of the data link protocol and thus $T_{TM,Slave.}$ ($T_{TM,Slave}$ terminates T_{BW2enc} seconds prior to the expected arrival of the first forward transmission.) Similarly, the Master need only measure the time of arrival of the TM_CONFIRM PDU to know when to expect the first HDL_ACK PDU, as these two events will be delayed by $T_{CTFA(n), HDL.}$

- Next, the data link protocol executes a succession of forward transmission / acknowledge handshakes. These handshakes occur with a period of $T_{HDL(n)}$. $T_{HDL(n)}$ is designed to account for waveform encoding and processing delays, and for worst-case propagation delay. $T_{HDL(n)}$ is defined in table C-III. $T_{HDL(n)}$ depends on the size of the forward transmission as established in the TM handshake. Note that the data link protocol time slots of the Slave are delayed T_{prop} with respect to the data link protocol time slots of the Master.

- Finally, the data link protocol concludes when the Master issues HDL_EOM PDU(s) (as many as can be concatenated without exceeding $T_{BW2(n)}$). If reverse traffic is pending, the Slave issues a TM_REQUEST PDU starting at the same time it would have issued an HDL_ACK PDU, the roles of Master and Slave reverse, and the timing proceeds as just described.

A similar analysis defines the timing structure for point-to-point packet traffic links using the LDL protocol, with the following substitutions:

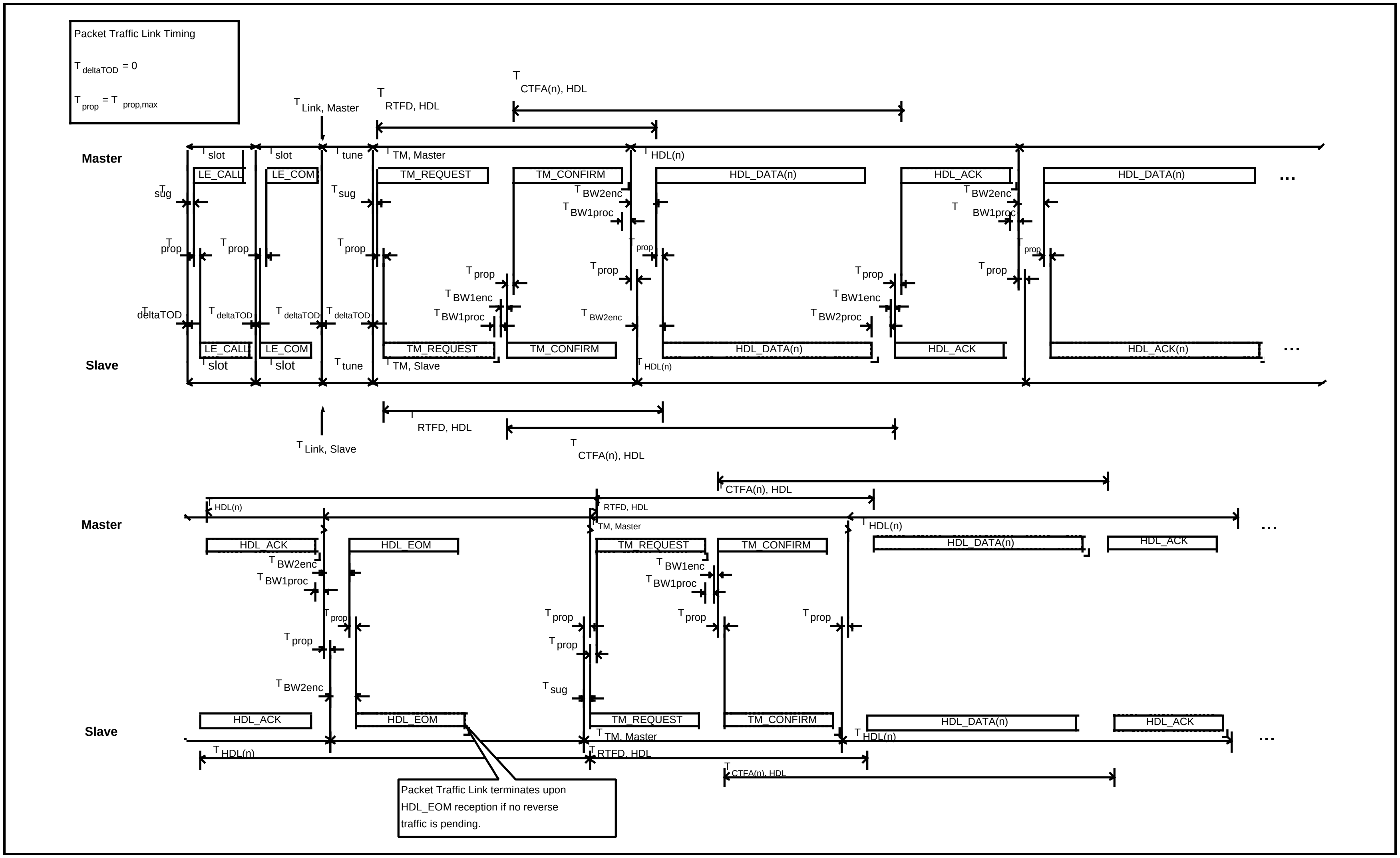
- TBW2enc is replaced by TBW3enc
- TBW2(n) is replaced by TBW3(n)
- TBW2proc is replaced by TBW3proc
- TBW1enc is replaced by TBW4enc
- TBW1 is replaced by TBW4
- TBW1proc is replaced by TBW4proc
- THDL(n) is replaced by TLDL(n)
- TRTFD, HDL is replaced by TRTFD, LDL
- TCTFA(n), HDL is replaced by TCTFA(n), LDL

C.5.3.5.5.1.2 <u>Point-to-point packet link timing requirements.</u>

The following requirements apply to point-to-point packet link timing:

1. Stations shall reckon the start of a link as the start of the 3G-ALE slot immediately following the slot in which the LE_COMMENCE PDU was transmitted.

2. For T_{tune} seconds after the start of the link, stations shall change to the traffic channel, if necessary, and tune, if necessary.
3. The Master shall begin emission of the TM_REQUEST PDU $T_{tune} + T_{sug}$ seconds after the start of the link.
4. The Slave shall begin emission of its response TM PDU $T_{BW1proc} + T_{BW1enc}$ seconds after the end of the TM_REQUEST PDU as observed by the Slave.
5. The Master shall begin emission of the first data link protocol HDL_DATA PDU $T_{RTFD, HDL}$ seconds after emission of the TM_REQUEST PDU began. Thereafter, the Master shall begin emissions of HDL_DATA PDUs $T_{HDL(n)}$ seconds after the emission of the previous HDL_DATA PDU began.
6. The Slave shall begin emission of the first data link protocol HDL_ACK PDU $T_{CTFA(n), HDL}$ seconds after emission of its response TM PDU began. Thereafter, the Slave shall begin emissions of HDL_ACK PDUs $T_{HDL(n)}$ seconds after the emission of the previous HDL_ACK PDU began.
7. The Master shall begin emission of the first HDL_EOM PDU $T_{HDL(n)}$ seconds after the emission of the previous HDL_DATA PDU began.
8. If reverse traffic is pending, the Slave shall begin emission of the TM_REQUEST PDU $T_{HDL(n)}$ seconds after the emission of the previous HDL_ACK PDU began. At this point the roles of Master and Slave reverse, and point-to-point packet link timing requirements 4-7 apply.
9. The Master shall begin emission of the first data link protocol LDL_DATA PDU $T_{RTFD, LDL}$ seconds after emission of the TM_REQUEST PDU began. Thereafter, the Master shall begin emissions of LDL_DATA PDUs $T_{LDL(n)}$ seconds after the emission of the previous LDL_DATA PDU began.
10. The Slave shall begin emission of the first data link protocol LDL_ACK PDU $T_{CTFA(n), LDL}$ seconds after emission of its response TM PDU began. Thereafter, the Slave shall begin emissions of LDL_ACK PDUs $T_{LDL(n)}$ seconds after the emission of the previous LDL_ACK PDU began.
11. The Master shall begin emission of the first LDL_EOM PDU $T_{LDL(n)}$ seconds after the emission of the previous LDL_DATA PDU began.
12. If reverse traffic is pending, the Slave shall begin emission of the TM_REQUEST PDU $T_{LDL(n)}$ seconds after the emission of the previous LDL_ACK PDU began. At this point the roles of Master and Slave reverse, and point-to-point packet link timing requirements 4 and 9-11 apply.



Packet Traffic Link terminates upon HDL_EOM reception if no reverse traffic is pending.

FIGURE C-26. Point-to-point packet link timing example for $T_{deltaTOD}$ =0, T_{prop} = $T_{prop,max}$.

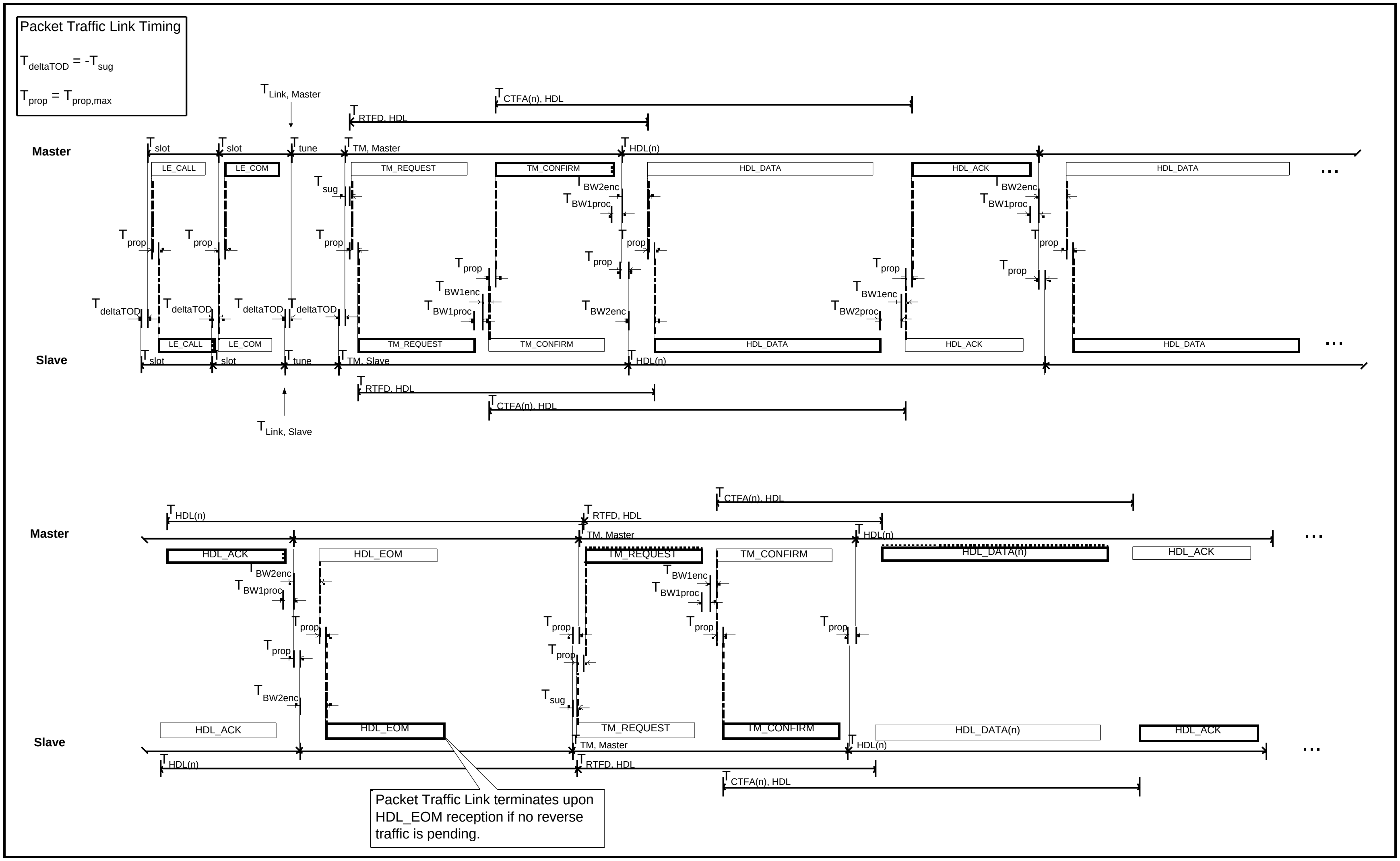
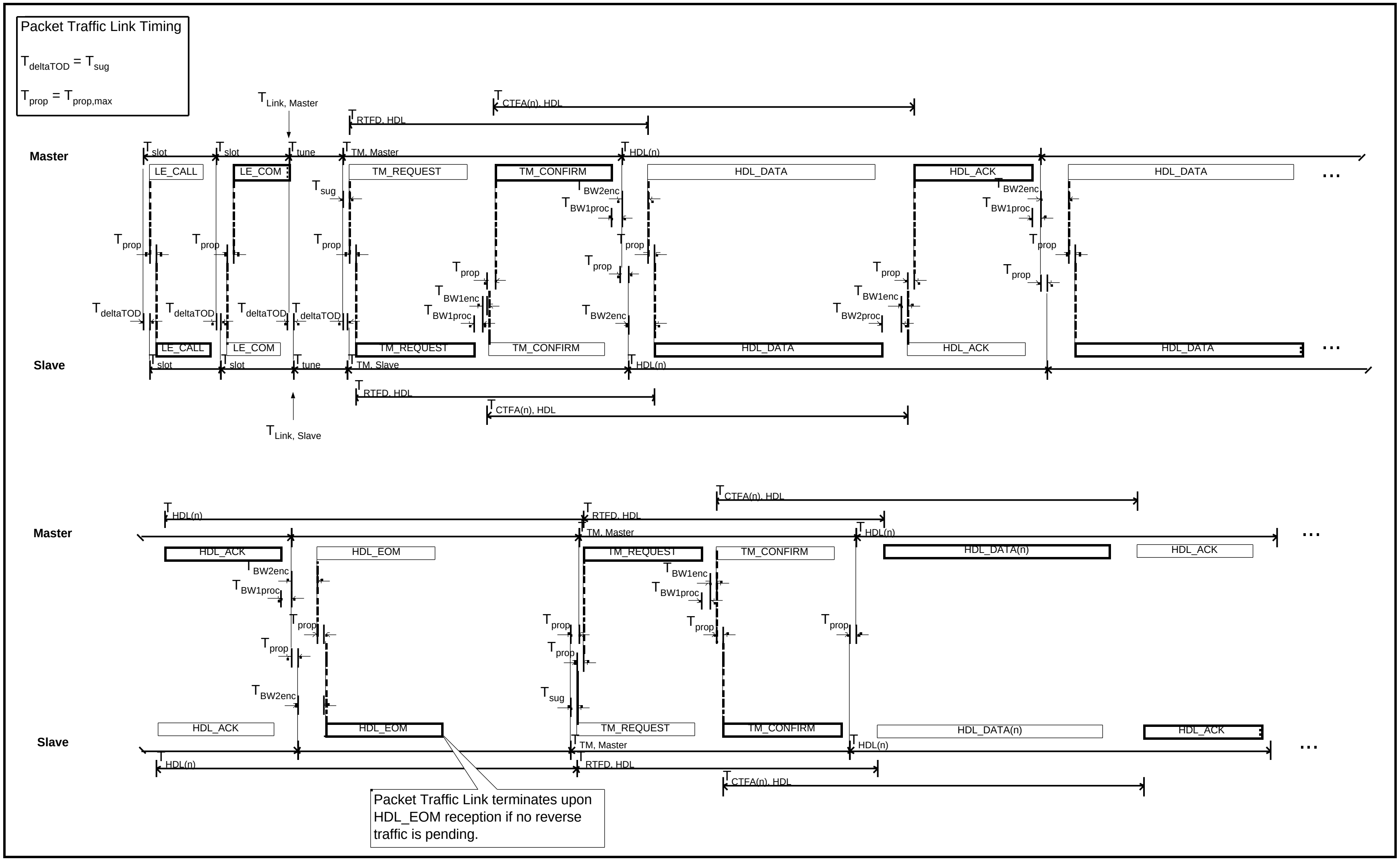
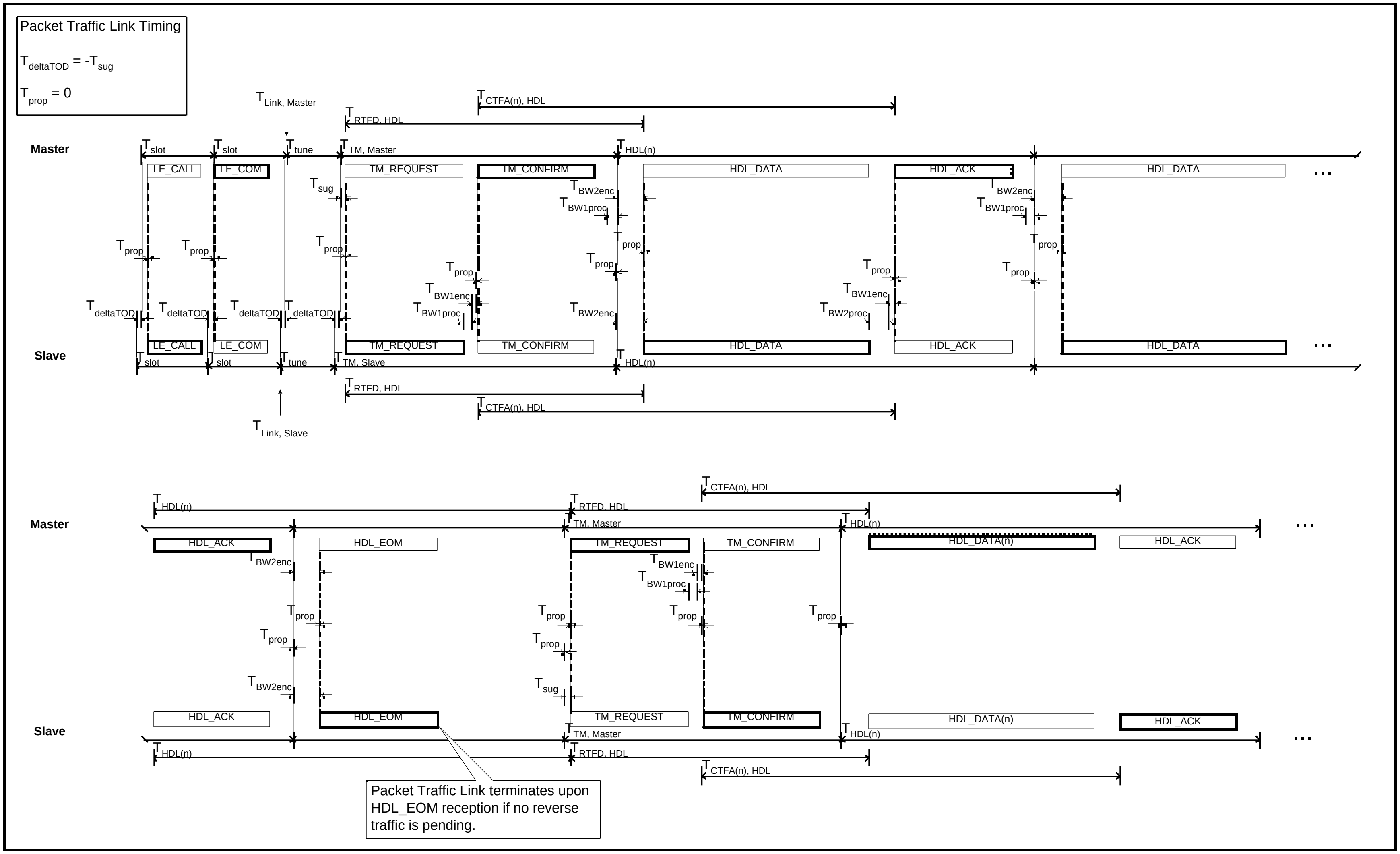


FIGURE C-27. Point-to-point packet link timing example for $T_{deltaTOD} = -T_{sug}$, $T_{prop} = T_{prop,max}$.



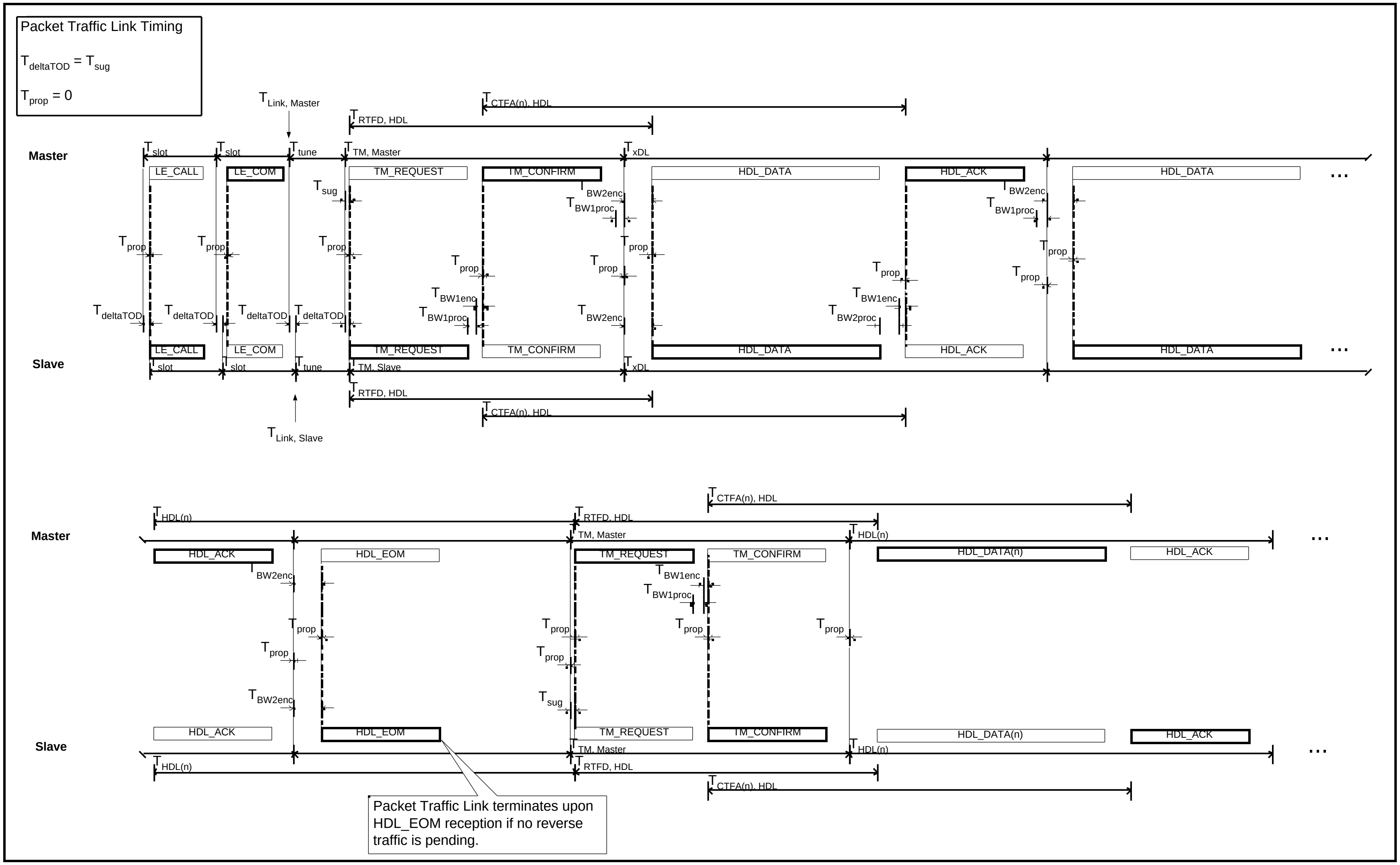
Packet Traffic Link terminates upon HDL_EOM reception if no reverse traffic is pending.

FIGURE C-28. Point-to-point packet link timing example for $T_{deltaTOD} = T_{sug}$, $T_{prop} = T_{prop,max}$.



Packet Traffic Link terminates upon HDL_EOM reception if no reverse traffic is pending.

FIGURE C-29. Point-to-point packet link timing example for $T_{deltaTOD}$ = -T_{sug}, T_{prop} = 0.



Packet Traffic Link terminates upon HDL_EOM reception if no reverse traffic is pending.

FIGURE C-30. Point-to-point packet link timing example for $T_{deltaTOD}$ = T_{sug}, T_{prop} = 0.

C.5.3.5.5.2 Point-to-point circuit links.

This section will first provide a description of point-to-point circuit link timing. Following this, point-to-point circuit link timing requirements will be given.

C.5.3.5.5.2.1 Point-to-point circuit link timing description.

The contents of this section are for informational purposes only.

Figure C-31 provides an example point-to-point circuit link. The linking activity proceeds as follows:

- First, an LE handshake is performed. See the section on point-to-point packet link timing for further explanation.
- Next, Master and Slave are given an opportunity to change to the traffic channel and tune, if necessary. This opportunity lasts T_{tune} seconds.
- Next, a TM handshake is performed. This handshake is identical to that performed for point-to-point packet links. The Slave is able to determine when to expect the start of the Master's transmission based on when it receives the TM_REQUEST PDU from the Master (the Master's transmission starts $-T_{BW1} - T_{sug} + T_{TM,Master} + T_{CLenc}$ after the end of the TM_REQUEST PDU as observed by the Slave).
- Finally, the Master transmits its message.

C.5.3.5.5.2.2 Point-to-point circuit link timing requirements.

The following requirements apply to point-to-point circuit link timing:

1. Stations shall reckon the start of a link as the start of the 3G-ALE slot immediately following the slot in which the LE_COMMENCE PDU was transmitted.
2. For T_{tune} seconds after the start of the link, stations shall change to the traffic channel, if necessary, and tune, if necessary.
3. The Master shall begin emission of the TM_REQUEST PDU $T_{tune} + T_{sug}$ seconds after the start of the link.
4. The Slave shall begin emission of its response TM PDU $T_{BW1proc} + T_{BW1enc}$ seconds after the end of the TM_REQUEST PDU as observed by the Slave.
5. The Master shall begin transmission of its message $T_{tune} + T_{TM,Master} + T_{CLenc}$ seconds after the start of the link.

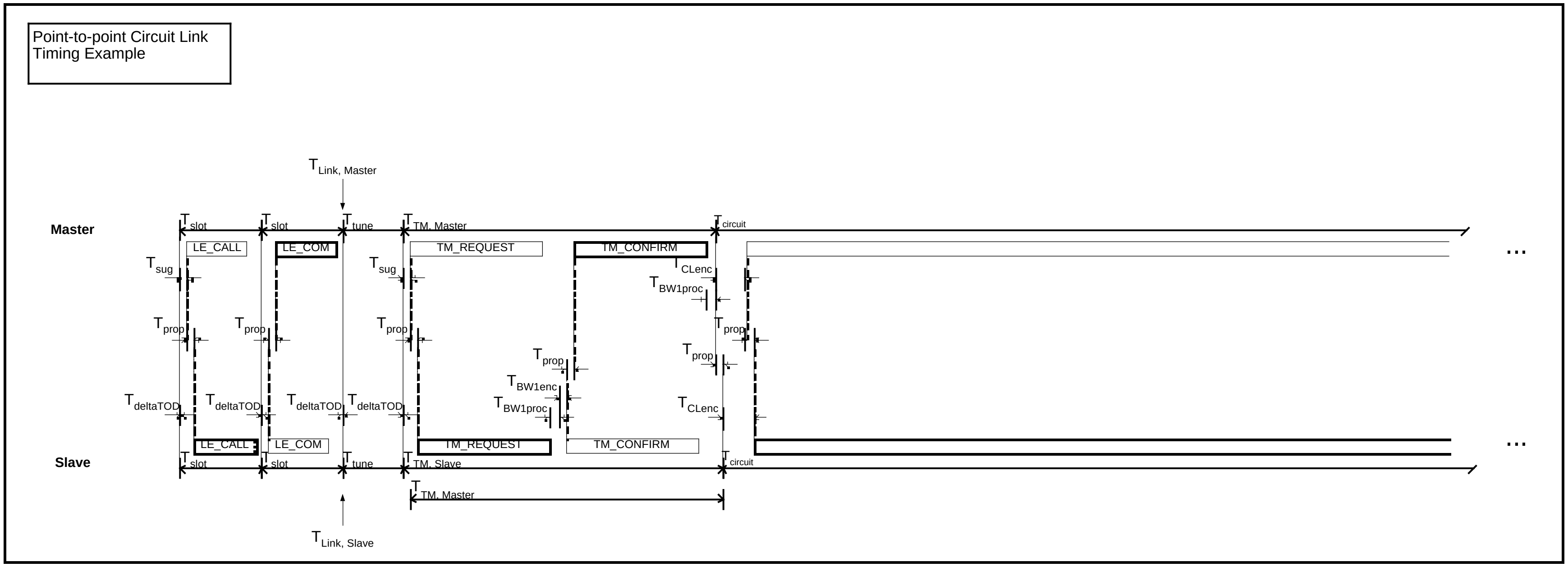


FIGURE C-31. Point-to-point circuit link timing example.

C.5.3.5.5.3 Multicast circuit link.

This section will first provide a description of multicast circuit link timing. Following this, multicast circuit link timing requirements will be given.

C.5.3.5.5.3.1 Multicast circuit link timing description.

The contents of this section are for informational purposes only. Figure C-32 shows an example multicast circuit link involving a Master and 3 Slaves. The linking activity for this example proceeds as follows:

- First, an LE handshake is performed. This handshake establishes a link time reference, T_{link}, for both the Master and the Slaves. This time reference is defined as the start of the LE slot immediately following the LE slot in which the LE_COMMENCE PDU was transmitted.
- Next, Master and Slaves are given an opportunity to change to the traffic channel and tune, if necessary. This opportunity lasts T_{tune} seconds.
- Next, a TM roll-call handshake is performed. The roll-call handshake begins with a two-way handshake between the Master and Slave 1 which is identical to the two-way handshake performed for point-to-point links. The remaining Slaves determine roll call slot timing based on when they receive the TM_REQUEST PDU from the Master (the first roll call slot starts $-T_{BW1} - T_{sug} + T_{TM,Master}$ after the end of the TM_REQUEST PDU as observed by each Slave). Each roll call slot $T_{rc\ slot,TM}$ seconds in duration. This roll call slot timing is designed to allow Slaves the time required to process a PDU arriving in the roll call slot immediately preceding the Slave’s own roll call slot. (See figure C-32. Observe that Slave 3 is given $T_{BW1proc}$ seconds to process the preceding PDU.)
- Finally, the multicast is transmitted by the Master T_{CLenc} seconds after the end of the last roll call slot.

C.5.3.5.5.3.2 Multicast circuit link timing requirements.

The following requirements apply to multicast circuit link timing:

1. Stations shall reckon the start of a link as the start of the 3G-ALE slot immediately following the slot in which the LE_COMMENCE PDU was transmitted.

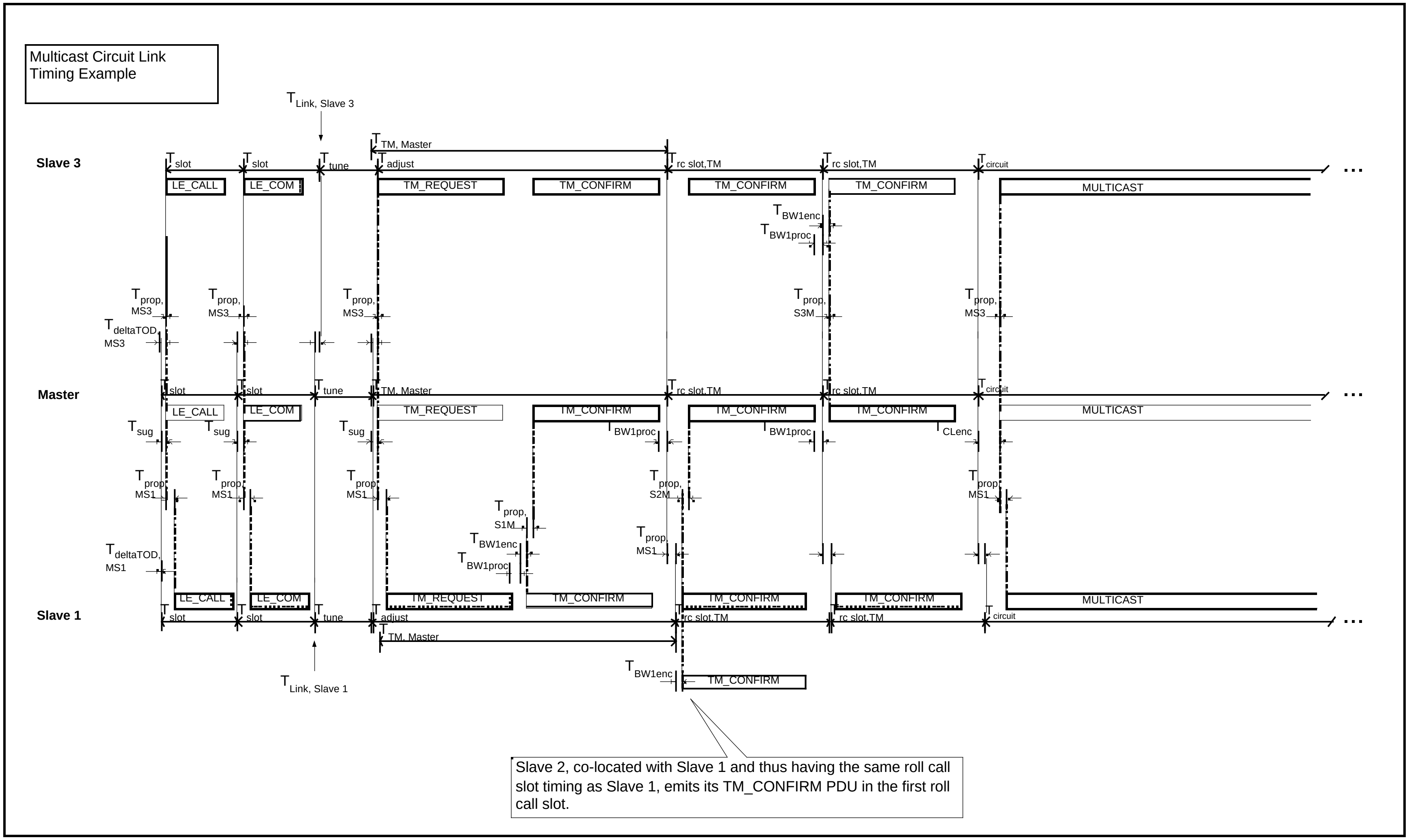
2. For T_{tune} seconds after the start of the link, stations shall change to the traffic channel, if necessary, and tune, if necessary.

3. The Master shall begin emission of the TM_REQUEST PDU $T_{tune} + T_{sug}$ seconds after the start of the link.

4. Slave 1 shall begin emission of its response TM PDU $T_{BW1proc} + T_{BW1enc}$ seconds after the end of the TM_REQUEST PDU as observed by Slave 1.

5. Slaves 2 through n, where n is the number of members of the multicast group, shall begin emission of their respective response PDUs $2*T_{BW1proc} + 2*T_{BW1enc} + T_{BW1} + 2*T_{prop,max} + (m-2)*T_{rc\ slot,TM}$ seconds after the end of the TM_REQUEST PDU as observed by the Slave, where m is the position of the Slave in the multicast group (e.g. m = 2 for Slave 2).

6. If, following the roll call, the Master elects to proceed with the multicast, the Master shall begin the multicast $T_{tune} + T_{TM,Master} + (n-1)*T_{rc\ slot,TM} + T_{CLenc}$ seconds after the start of the link. Otherwise, if the Master elects to transmit TM protocol PDU(s), the Master shall begin emission of such PDUs $T_{tune} + T_{TM,Master} + (n-1)*T_{rc\ slot,TM} + T_{BW1enc}$ seconds after the start of the link. Again, n is the number of members in the multicast group.



Slave 2, co-located with Slave 1 and thus having the same roll call slot timing as Slave 1, emits its TM_CONFIRM PDU in the first roll call slot.

FIGURE C-32. Multicast circuit link timing example.

C.5.4 HDL protocol.

C.5.4.1 Overview.

The HDL is used to provide acknowledged point-to-point delivery of datagrams from a transmitting station to a receiving station across an already-established HF link, with selective retransmission (ARQ) of data received in error. The datagram passed to the HDL for delivery is a finite-length ordered sequence of 8-bit data bytes (octets). The HDL protocol is best suited to delivering relatively large datagrams under good to fair HF channel conditions. By contrast, the LDL protocol described in section C.5.5 provides better performance for all datagram lengths under fair to very poor HF channel conditions, and under all channel conditions for short datagrams.

C.5.4.2 Data object types.

The terms defined in table C-XLIII are used to refer to specific types of data objects in defining the HDL protocol.

TABLE C-XLIII. HDL data object types.

Data object type	Definition
datagram	an ordered sequence of 8-bit data bytes (octets). No limit on the size of the datagram.
data segment	an ordered sequence of 8-bit data bytes (octets) that occur consecutively within a datagram, of length *sl* where $1 \leq sl \leq 233$.
sequence number	a 17-bit data object having the format defined in table C-XLVI, which indicates the position occupied by a data segment within a datagram, and, when the data segment includes the last data byte of the datagram, the number of bytes of payload data from the datagram in the data segment.
data packet	the combination of a data segment with the sequence number indicating its position within a datagram. If the data segment is of length less than 233 bytes (because it includes the last data byte of the datagram*)*, a sequence of null data bytes (of value zero) is appended to the data segment so as to extend it to length 233 in constructing the data packet, and the value of the sequence number indicates how many of the 233 bytes in the extended data segment contain payload data from the datagram.
tx frame	a sequence of np data packets, where np = 24, 12, 6, or 3, and the value of *np* is determined by the numPkts parameter of the most recent HDL_Send_Req or HDL_Rcv_Req service primitive. Same as an HDL_DATA PDU as defined in C.5.4.4.1.
packet index	a number indicating the ordinal position of a data packet within a *tx frame*, such that packet *index* = 0 indicates that the data packet occupies the first position in the *tx frame*.
indexed packet	the combination of a data packet with the packet index indicating the data *packet*'s ordinal position within a specific *tx frame*.
rx frame	a sequence of *np* indexed packets, where $0 \leq np \leq 24$, which includes an indexed packet containing each data packet that was received without errors (as determined by checking its CRC) in an incoming *tx frame*. *np* is never greater than the value of the numPkts parameter of the most recent HDL_Rcv_Req service primitive, but may be less due to packets' having been omitted from the *rx frame* (by the BW2 receiver) due to their having been received containing errors.

C.5.4.3 Service primitives.

Table C-XLIV describes the service primitives exchanged between the HDL protocol entity HDL and one or more user processes at HDL's upper interface. These service primitives are used in this specification only as expository devices, and are not required to be present in any compliant implementation of the protocol. Note that there is no requirement that implementations of the waveforms and protocols defined in this Appendix contain precisely these service primitives; nor are the services primitives defined below necessarily all of the service primitives that would be required in an implementation of these waveforms and protocols.

C.5.4.4 PDUs.

The sub-sections of this section describe the PDUs exchanged between a HDL protocol HDL entity and its remote peer entities.

C.5.4.4.1 HDL_DATA.

An HDL_DATA PDU is a tx frame as defined in table C-XLIII, in which the format and contents of each data packet are as shown in table C-XLV. Table C-XLVI specifies the format and contents of the Sequence Number field of each data packet.

TABLE C-XLIV. HDL service primitives.

Name	Attribute	Values	Description
HDL_Send_Req	Overview	HDL Send Request: generated by the user process when it has a datagram to send on an already-established HF link.	
	Parameters	datagram	datagram to be delivered, as described in table C-XLIII.
		numPkts	the number *np* of data packets to be transmitted in each forward transmission by BW2 (i.e., each tx frame), where *np* = 24, 12, 6, or 3.
	Originator	User process	
	Preconditions	TM has just completed a successful point-to-point HF link establishment with the intended datagram recipient.	
HDL_Rcv_Req	Overview	HDL Receive Request: generated by the user process to request that HDL perform the processing required to receive an expected incoming datagram.	
	Parameters	numPkts	the number *np* of data packets that will be received in each incoming transmission by BW2, where *np* = 24, 12, 6, or 3.
	Originator	User process	
	Preconditions	TM has just completed a successful point-to-point HF link establishment with the expected datagram sender.	
HDL_Rcv_Ind	Overview	HDL Receive Indication: issued by HDL when HDL has a successfully-received datagram to give to the local user process.	
	Parameters	datagram	datagram just received, as described in table C-XLIII; identical to the original datagram parameter-value of the HDL_Send_Req primitive at the remote station.
	Originator	HDL	
	Preconditions	HDL has received an HDL_Rcv_Req since the last outgoing or incoming datagram transfer.	
HDL_Send_Conf	Overview	HDL Send Confirm: Issued by HDL when HDL has completed successful delivery of a datagram to the remote station.	
	Parameters	(none)	
	Originator	HDL	
	Preconditions	HDL was requested to deliver the datagram by the user process by means of an HDL_Send_Req service primitive.	
HDL_Abort_Req	Overview	HDL Abort Request: used by the user process to terminate a HDL protocol data transfer that is currently in progress. The purpose of this service primitive is only to cause the HDL entity to cease attempting to send or receive the current datagram; coordinating the data transfer termination with the remote station is the responsibility of TM.	
	Parameters	(none)	
	Originator	User process	
	Preconditions	Either an outgoing or an incoming data transfer is in progress, using the HDL protocol.	
HDL_Failure_Ind	Overview	HDL Failure Indication: Issued by HDL when HDL is unable to complete delivery of an outgoing or incoming datagram.	
	Parameters	(none)	
	Originator	HDL	
	Preconditions	Either an outgoing or an incoming data transfer is in progress, using the HDL protocol.	

TABLE C-XLV. Data packet format.

Field name	Size (bits)	Values	Description
Payload	1864 (fixed)	any	Contains a data segment as defined in table C-XLIII, followed by however many zero bytes are needed to fill the Payload field to length 233 bytes. Whenever the field contains fewer than 233 bytes of payload data (from the outgoing datagram), the value of the Sequence Number field indicates how many bytes of payload data are present.
Sequence Number	17 (fixed)	As defined by table C-XLVI.	

The bytes of the Payload field are transmitted in the order of their occurrence in the datagram; the bits of each byte are transmitted in order of significance, starting with the most significant bit.

HDL_DATA PDUs are transmitted using the BW2 burst waveform described by section C.5.1.5.

TABLE C-XLVI. Sequence number field definition.

Case	bit 16 (EOM)	bit 15 (SOM)	bits 14-6	bits 5 - 0
only packet in datagram	1	1	Packet Byte Count: 9 bits, (number of user bytes in packet –1)	0
last packet in datagram	1	0	Packet Byte Count: 9 bits, (number of user bytes in packet –1)	0…63 (see Packet Number below)
first packet in datagram	0	1	0	0
packet in interior of datagram	0	0	Packet Byte Count: 9 bits, (number of user bytes in packet –1)	Packet Number: 6 bits, counts up from zero and wraps (0,1,..,63,0).

Following the last bit of the Payload field-value, the bits of the Sequence Number field are transmitted in order of significance within the 17-bit field-value, starting with the most significant bit (bit 16).

C.5.4.4.2 HDL_ACK.

The HDL_ACK PDU is used to transfer selective acknowledgements, in the form of an ack bit-mask containing a single bit for each data packet in an HDL_DATA PDU, from the receiving station to the sending station in a HDL transfer. Table C-XLVII specifies the format and contents of the HDL_ACK PDU.

TABLE C-XLVII. HDL_ACK PDU format.

Field name	Size (bits)	Values	Description
Protocol	3	000_2 = HDL	identifies this PDU as an HDL PDU
Type	1	0_2 = ACK	Identifies this PDU as an HDL_ACK PDU
Ack Bit-Mask	24	any	Contains one ack bit for each of the data packets that can be received in an HDL_DATA PDU. Each bit indicates whether the corresponding data packet was received without errors; 1 = ACK (was received successfully); 0 = NAK (not received successfully).
Reserved	4	0000_2 (fixed)	Reserved for future use.
CRC	16	any	16-bit Cyclic Redundancy Check (CRC) computed across the values of the Type and Ack Bit-Mask fields, using the generator polynomial $X^{16} + X^{15} + X^{11} + X^8 + X^6 + X^5 + X^4 + X^3 + X^1 + 1$, and the procedure described in C.4.12.

The fields of the HDL_ACK PDU are transmitted in the order of their occurrence in table C-XLVII, protocol field first. Bits of the Ack Bit-Mask field are transmitted in the order in which the corresponding data packets were transmitted in the HDL_DATA PDU that is being acknowledged; for instance, the first ack bit acknowledges the first data packet.

HDL_ACK PDUs are transmitted using the BW1 burst waveform described by section C.5.1.4.

C.5.4.4.3 HDL_EOM.

The HDL_EOM PDU is sent from the sending station to the receiving station in a HDL transfer, to indicate that the sending station has received acknowledgements from the receiving station of every data packet in the datagram being transferred, and hence will send no more HDL_DATA PDUs for the current datagram. The HDL_EOM PDU is sent as many times as possible within the time interval defined for a forward transmission (based on the value of numPkts), to maximize the probability of its being received without errors. Table C-XLVIII specifies the format and contents of the HDL_EOM PDU.

TABLE C-XLVIII. HDL_EOM PDU.

Field name	size (bits)	Values	Description
Protocol	3	000_2 = HDL	identifies this PDU as an HDL PDU
Type	1	1_2 = EOM	Identifies this PDU as an HDL_EOM PDU.
Check	44	0xA5A5A5A5A5A	Unused, but should be inspected by the recipient to verify that it contains the correct bit-pattern specified here.

The fields of the HDL_EOM PDU are transmitted in the order of their occurrence in table C-XLVIII, protocol field first. The bits of the Check field are transmitted following the single bit of the Type field in order of significance, most significant bit first, so that the first four bits transmitted from the Check field are 1, 0, 1, 0.

HDL_EOM PDUs are transmitted using the BW1 burst waveform described by section C.5.1.4.

On traffic links established for packet traffic delivered using the HDL protocol, the user process can terminate the data link transfer and use the next data link transmission time slot in either direction — i.e., the time slot for the HDL_DATA or the HDL_ACK PDU — to instead send one or more TM PDUs (described in C.5.3.4) within the data link PDU time-slot. This means that while a HDL transfer is in progress, the receiving station must be simultaneously attempting to demodulate TM PDUs conveyed by the BW1 waveform as it is attempting to demodulate and receive HDL_DATA PDUs conveyed by BW2. The sending station may receive a TM PDU conveyed by BW1 in place of an HDL_ACK PDU (also BW1), and must ensure that this PDU is received and processed by its TM sublayer.

C.5.4.5 Protocol behavior.

This section provides an informal overview of the behavior of the HDL protocol. The following sections define this behavior precisely:

- C.5.4.5.1 identifies and defines the events to which HDL responds;
- C.5.4.5.2 identifies and defines the actions taken by HDL in response to these events;
- C.5.4.5.3 describes the data items used and maintained by HDL;
- C.5.4.5.4 provides a state diagram and a state transition table specifying the behavior of HDL in terms of these events, actions, and data items; and
- C.5.4.5.5 provides additional information on the timing characteristics of HDL behavior.

Data transfer by HDL begins after the TM sublayer has already established the data link connection, in so doing negotiating the fact that HDL will be used (as opposed to LDL or some other mechanism), the number of data packets to be sent in each HDL_DATA PDU, and the precise time synchronization of data link transmissions.

In an HDL data transfer, the sending station and the receiving station alternate transmissions in the manner depicted in figure C-33, the sending station transmitting HDL_DATA PDUs containing payload data packets, and the receiving station transmitting HDL_ACK PDUs containing acknowledgements of the data packets received without errors in the preceding HDL_DATA PDU. If either station fails to receive a PDU at the expected time, it sends its own next outgoing PDU at the same time as if the incoming PDU had been received successfully. The times at which the burst waveforms conveying HDL_DATA, HDL_ACK, and HDL_EOM PDUs may be emitted are precisely stipulated; see C.5.4.5.5 for details.

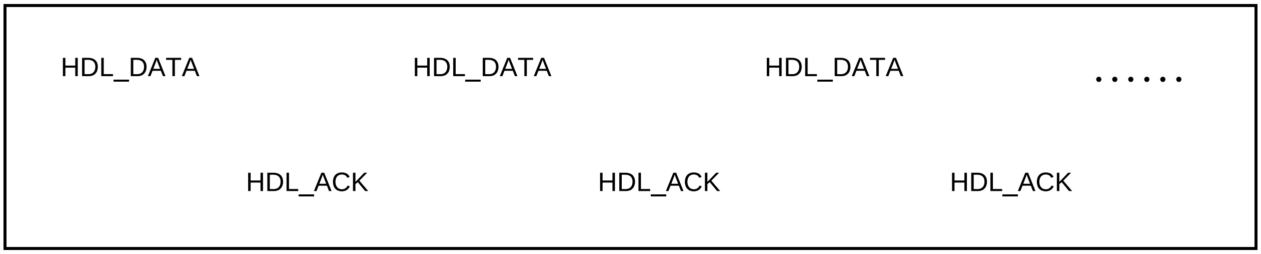


FIGURE C-33. HDL data transfer overview.

The end of a data transfer is reached when the sending station has transmitted HDL_DATA PDUs containing all of the payload data in the delivered datagram, and the receiving station has received these data without errors and has acknowledged their successful delivery. When the sending station receives an HDL_ACK PDU indicating that the entire contents of the datagram have been delivered successfully, it sends an HDL_EOM PDU repeated as many times as possible within the duration of an HDL_DATA PDU, starting at the time at which it would have otherwise transmitted the next HDL_DATA PDU, to indicate to the receiving station that the data transfer will be terminated. This link termination scenario is depicted in figure C-34.

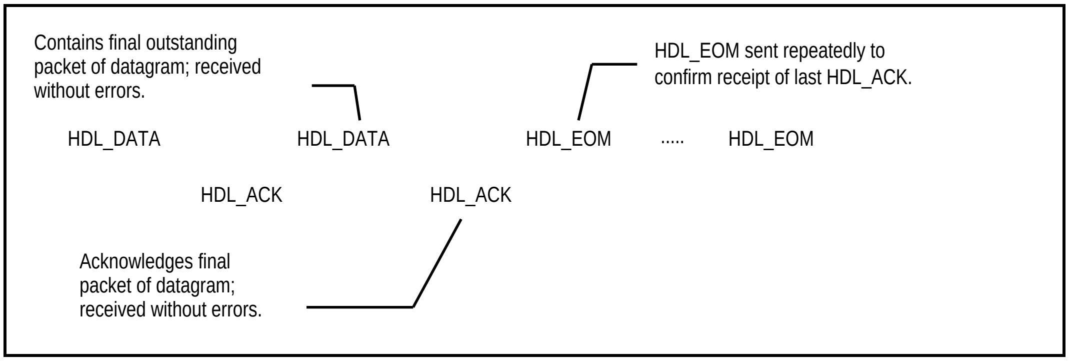

Contains final outstanding packet of datagram; received without errors.
HDL_EOM sent repeatedly to confirm receipt of last HDL_ACK.
HDL_DATA
HDL_DATA
HDL_EOM
.....
HDL_EOM
HDL_ACK
HDL_ACK
Acknowledges final packet of datagram; received without errors.

FIGURE C-34. HDL link termination scenario overview.

The definition of HDL behavior presented in the following sections includes mechanisms for dealing appropriately with the following occurrences:

- excessive number of consecutive failures to receive an expected HDL_DATA PDU;
- excessive number of consecutive failures to receive an expected HDL_ACK PDU;
- immediate termination of an ongoing data link transfer requested by the TM sublayer.

C.5.4.5.1 Events.

Table C-XLIX defines the events to which the HDL entity responds. The event names are used in the state diagram and the state transition table in C.5.4.5.4, which define the behavior of the HDL protocol. Some event names refer to the receipt of PDUs from the HDL entity at a remote station; in these cases, the ‘description’ field of the table entry describes the manner in which the arrival of a PDU is accomplished through HDL's accepting one or more service primitives from lower-layer entities at the local station. The prefix 'R:' in the name of an event indicates that the event is the receipt of a PDU from the remote station. 'D:' indicates that the event is an HDL service primitive passed down to HDL from a higher-layer entity; 'U:' indicates a lower-layer service primitive passed *up* to HDL from a lower-layer entity.

TABLE C-XLIX. HDL events.

Event name	Description
R:HDL_DATA PDU	A BW2_Receive primitive containing an HDL_DATA PDU was accepted.
R:HDL_ACK PDU	A BW1_Receive primitive containing an HDL_ACK PDU was accepted, and the HDL_ACK PDU was found to be free of errors by checking its CRC.
R:HDL_EOM PDU	A BW1_Receive primitive containing a HDL_EOM PDU was accepted, and the HDL_EOM PDU was found to be free of errors by comparing the received PDU against the known HDL_EOM bit pattern specified in table C-XLVIII.
D:HDL_Send_Req	An HDL_Send_Req primitive was accepted from the user process.
D:HDL_Abort_Req	An HDL_Abort_Req primitive was accepted from the user process.
AckTimeout	A valid HDL_ACK PDU was not received within the time period in which it was expected.
DataTimeout	An HDL_DATA PDU was not received within the time period in which it was expected.

C.5.4.5.2 Actions.

Table C-L defines the actions which the HDL entity can perform. The action name is used in the state diagrams and/or state transition tables used below to define the behavior of the HDL protocol. Some action names refer to sending PDUs to the HDL entity at a remote station; in these cases, the 'description' field of the table entry describes the manner in which sending of the PDU is accomplished by issuing one or more service primitives to lower-layer entities at the local station.

C.5.4.5.3 Data.

Table C-LI defines the data items used and maintained by HDL, including buffers, counters, timers, configuration parameters, and so forth. These data items are referred to by the names assigned to them here, in the definitions of HDL events and actions presented in the preceding sections. These data items are used in this specification only as expository devices, and are not required to be present in any compliant implementation of the protocol.

TABLE C-L. HDL actions.

Action name	Description
TxInit	Set NumPktsTx to the value of the numPkts parameter of the HDL_Send_Req primitive. Insert the outgoing datagram into TxDatagramBuf. Clear TxFrameBuf. Reset MissedAckCount to zero.
S:HDL_DATA	For each of the first NumPktsTx data packet positions in TxFrameBuf, if the data packet position is empty, construct a new outgoing data packet (as described by table C-XLV) in this position: 1. Get the next data segment (the next 233 consecutive data bytes to be transmitted) from TxDatagramBuf; place these in the Payload field of the data packet. If fewer than 233 data bytes remain to be transmitted, place these bytes at the beginning of the Payload field; fill the remainder of the field with zero bytes. 2. Construct a sequence number value as specified in table C-XLVI; write this value into the packet's Sequence Number field. If TxDatagramBuf is completely emptied of payload data before all packet positions in TxFrameBuf have been filled with new packets, fill the remaining packet positions with repetitions of packets already residing in other positions of TxFrameBuf. **Note:** The HDL transmitter is at liberty to select packets from the current datagram to repeat as it pleases; the HDL receiver must inspect the sequence number of each packet received without errors, and use this information to discard duplicate packets. **Note:** Whenever a packet is retransmitted, it is always placed in the same packet position within the Tx frame that it occupied in the previous transmission. Send an HDL_DATA PDU containing the tx frame in TxFrameBuf, using a BW2_Send primitive. Set the primitive's reset parameter to TRUE if this is the first tx frame transmitted for the current datagram, and to FALSE otherwise. Reset the AckTimeout timer. If an AckTimeout has occurred, increment MissedAckCount.
ProcessAck	Check the HDL_ACK PDU's CRC. If the CRC is valid: 1. Copy the Ack Bit-Mask field value to RxAck. 2. For each i, $0 \leq i \leq$ (NumPktsTx - 1), if RxAck[i] is 1, clear the i^{th} position of TxFrameBuf. (Each unacknowledged packet is retained in its current position in TxFrameBuf, and will be retransmitted in the same position that it occupied in the previous transmission.) 3. If TxDatagramBuf is not empty, set the condition MoreToSend to TRUE; otherwise set it to FALSE.
S:HDL_EOM	Send an HDL_EOM PDU to the remote station using BW1_Send, as many times as possible within the duration of an HDL_DATA PDU. The number of times the HDL_EOM PDU is sent depends on the value of NumPktsTx as follows: · if NumPktsTx = 3, the HDL_EOM PDU is transmitted *once* · if NumPktsTx = 6, the HDL_EOM PDU is transmitted *once* · if NumPktsTx = 12, the HDL_EOM PDU is transmitted *three* times · if NumPktsTx = 24, the HDL_EOM PDU is transmitted *seven* times. Disable AckTimeout timer.
U:HDL_Send_Conf	Issue an HDL_Send_Conf primitive to the user process that requested the outgoing data transfer.
AbortTransmit	Disable AckTimeout timer; reset MissedAckCount to zero.
RxInit	Set NumPktsRx to the value of the numPkts parameter of the HDL_Rcv_Req primitive. Clear RxDatagramBuf, and RxFrameBuf. Reset MissedTxFrameCount to zero. Set CompleteDatagramRcvd to FALSE.

TABLE C-L. HDL actions (continued).

Action name	Description
S:HDL_ACK(ack/nak)	Clear TxAck to all zeroes. Reset MissedTxFrameCount to zero. if CompleteDatagramRcvd is FALSE then For each indexed packet received from the BW2 receiver, 1. Insert a data packet containing the Payload and Sequence Number field values from the indexed packet into RxFrameBuf[i], where i = the Index value from the indexed packet (the packet's position in the transmitted HDL_DATA PDU). 2. Set TxAck[i] to 1. 3. Move the Payload field contents of RxFrameBuf[i] to the position in RxDatagramBuf indicated by the Sequence Number field-value. 4. If all packets for the current datagram have been received without errors, set CompleteDatagramRcvd to TRUE. else (CompleteDatagramRcvd is TRUE) Set the first NumPktsRx bits of TxAck to 1. end if Send an HDL_ACK PDU containing TxAck, using BW1_Send. Reset the DataTimeout timer. **Note:** An implementation of HDL can, without impairing compliance to this standard, provide segments of a partially-received datagram to the user process, in order of their occurrence in the original datagram at the sending station, before the entire datagram has been received. Doing so would allow a higher-layer protocol to abort an ongoing data transfer, then resume it at a later time, without having to retransmit the entire portion of the current datagram that was already delivered successfully. **Note:** The HDL transmitter can send duplicate packets either as a result of missing an HDL_ACK PDU, or at the end of a datagram, in order to fill the (otherwise unused) packet positions of an HDL_DATA PDU. The HDL receiver is required to inspect the sequence number of each data packet received without errors, and to use the sequence numbers to identify and discard duplicate packets.
S:HDL_ACK(naks)	Reset the DataTimeout timer. Clear TxAck to all zeroes. Send an HDL_ACK PDU containing TxAck using BW1_Send. If a DataTimeout has occurred, increment MissedTxFrameCount.
U:HDL_Rcv_Ind	Issue an HDL_Rcv_Ind primitive containing the received datagram to the user process.
AbortReceive	Disable DataTimeout timer; reset MissedTxFrameCount to zero.

TABLE C-LI. HDL data items.

Data item	Description
NumPktsTx	the number of data packets in each tx frame, as negotiated during the Traffic Set-Up (TSU) phase.
TxDatagramBuf	buffer storing the data contents of an outgoing datagram which have not yet been inserted into a tx frame for transmission.
TxFrameBuf	buffer storing the sequence of NumPkts data packets contained in an outgoing tx frame (an HDL_DATA PDU).
NumPktsRx	the maximum number of data packets in each incoming rx frame, as negotiated during the Traffic Set-Up (TSU) phase.
RxDatagramBuf	buffer storing the data contents of an incoming datagram which have been received thus far, in which the complete incoming datagram is re-assembled in correct order.
RxFrameBuf	buffer storing those incoming data packets of an rx frame which were received without errors (as determined by the CRC check performed by the BW2 receiver).
TxAck	ack flag sequence to be transmitted to the remote station. Contains one bit-flag for each data packet in a maximum-length tx frame; TxAck[i] = 1 indicates that the i^{th} data packet in the most recently received rx frame was received without errors.
RxAck	ack flag sequence received in an HDL_ACK PDU from the remote station. Contains one bit-flag for each data packet in a maximum-length tx frame; RxAck[i] = 1 indicates that the remote station received the i^{th} data packet in the previously-transmitted rx frame without errors.
MissedAckCount	count of consecutive failures to receive an HDL_ACK PDU in the time period in which one was expected.
MissedTxFrameCount	count of consecutive failures to receive an HDL_DATA PDU (a tx frame) in the time period in which one was expected.
AckTimeout	timer used to time the duration of the interval in which receipt of an HDL_ACK PDU is expected; fires when the interval expires.
DataTimeout	timer used to time the duration of the interval in which receipt of an HDL_DATA PDU is expected; fires when the interval expires.
MAX_MISSED_ACKS	HDL configuration parameter specifying the maximum number of consecutive missed HDL_ACK PDUs that can occur without causing the HDL transmitter to terminate the data link transfer. The value of this parameter is not stipulated by this specification, since it is not required for interoperability that this parameter have identical values in both the sending and receiving stations.
MAX_MISSED_TX_FRAMES	HDL configuration parameter specifying the maximum number of consecutive missed HDL_DATA PDUs that can occur without causing the HDL receiver to terminate the data link transfer. The value of this parameter is not stipulated by this specification, since it is not required for interoperability that this parameter have identical values in both the sending and receiving stations.
MoreToSend	Boolean condition variable: is TRUE if and only if an outgoing datagram transfer is in progress, and there are one or more data packets in the datagram for which the local station has not yet received an acknowledgement of their receipt without errors by the remote station.
CompleteDatagramRcvd	Boolean condition variable: is TRUE if and only if an incoming datagram transfer has been successfully completed (all contents of the datagram received without errors), but an HDL_Rcv_Ind primitive has not yet been issued to the user process.

C.5.4.5.4 Behavior definition.

For the reader's convenience, two equivalent representations of the behavior of the HDL protocol are provided in this section: the state transition table in table C-LII, and the state diagram in figure C-35.

Both table C-LII and figure C-35 specify the behavior of HDL in terms of the events defined in C.5.4.5.1 and the actions defined in C.5.4.5.2. The conditions gating certain transitions are specified in terms of the data items defined in C.5.4.5.3.

In the state diagram, each state transition is labeled with an event, an optional condition, and zero or more actions. This indicates that the state transition occurs whenever the event occurs and the condition obtains (is TRUE), causing the associated actions to be performed. In the diagram,

- the name of each event is shown in brackets preceded by the letter 'E';
- the description of each condition is shown in brackets preceded by the letter 'C'; and
- the names of the actions associated with a transition are shown in brackets preceded by the letter 'A'.

Where a transition is labeled with two or more events, this indicates that the transition occurs whenever any of the events occurs.

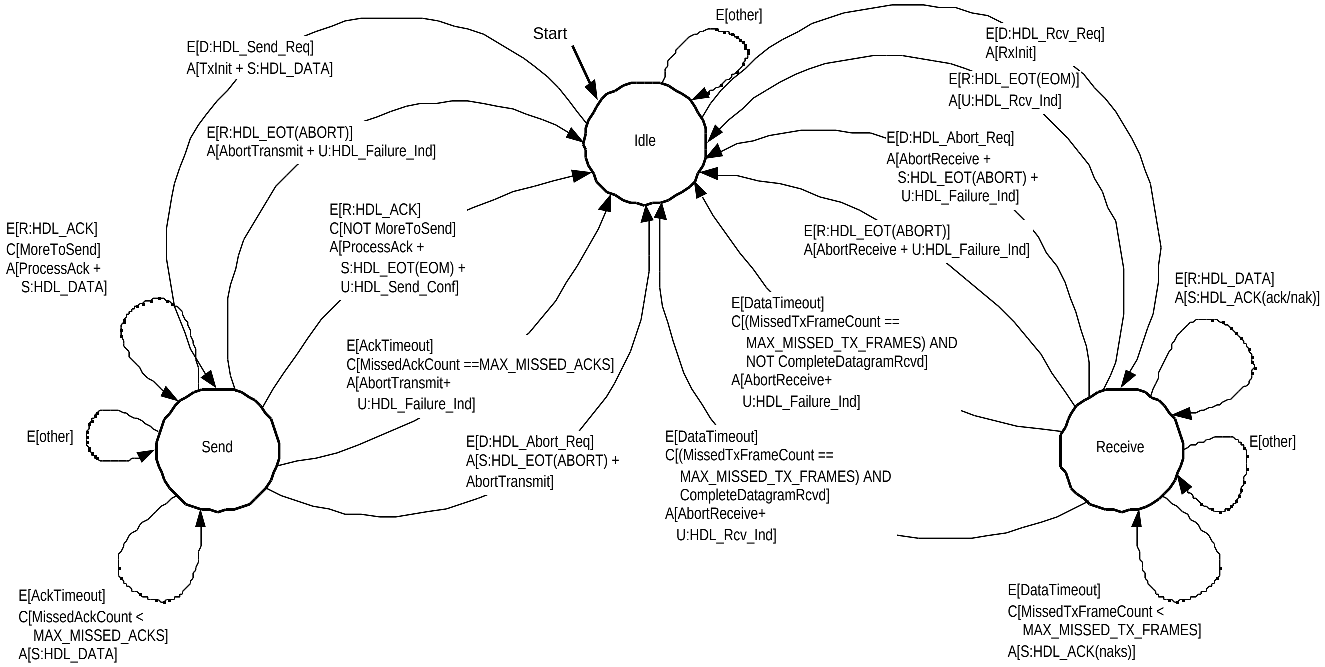


FIGURE C-35. HDL state diagram.

TABLE C-LII. HDL state transition table.

State	Event	Condition	Action	Next State
Idle	D:HDL_Send_Req		TxInit + S:HDL_DATA	Send
	D:HDL_Rcv_Req		RxInit	Receive
	other		none	Idle
Send	R:HDL_ACK	MoreToSend	ProcessAck + S:HDL_DATA	Send
	R:HDL_ACK	NOT MoreToSend	ProcessAck + S:HDL_EOM + U:HDL_Send_Conf	Idle
	R:HDL_EOT(ABORT)		AbortTransmit + U:HDL_Failure_Ind	Idle
	AckTimeout	MissedAckCount < MAX_MISSED_ACKS	S:HDL_DATA	Send
	AckTimeout	MissedAckCount == MAX_MISSED_ACKS	AbortTransmit + U:HDL_Failure_Ind	Idle
	D:HDL_Abort_Req		AbortTransmit	Idle
	other		none	Send
Receive	R:HDL_DATA		S:HDL_ACK(ack/nak)	Receive
	R:HDL_EOT(EOM)		U:HDL_Rcv_Ind	Idle
	R:HDL_EOT(ABORT)		AbortReceive + U:HDL_Failure_Ind	Idle
	DataTimeout	MissedTxFrameCount < MAX _MISSED_TX_FRAMES	S:HDL_ACK(naks)	Receive
	DataTimeout	(MissedTxFrameCount = MAX _MISSED_TX_FRAMES) AND NOT CompleteDatagramRcvd	AbortReceive + U:HDL_Failure_Ind	Idle
	DataTimeout	(MissedTxFrameCount == MAX_MISSED_TX_FRAMES) AND CompleteDatagramRcvd	AbortReceive + U:HDL_Rcv_Ind	Idle
	D:HDL_Abort_Req		AbortReceive + S:HDL_EOT(ABORT) + U:HDL_Failure_Ind	Idle
	other		none	Receive

C.5.4.5.5 Timing characteristics.

See C.5.3.5.5, which includes an analysis of the timing of the HDL protocol in conjunction with TM protocol timing.

C.5.5 LDL protocol .

C.5.5.1 Overview.

The LDL protocol is used to provide reliable acknowledged point-to-point delivery of datagrams from a transmitting station to a receiving station across an already-established HF link. The datagram passed to the LDL protocol entity for delivery is a finite-length ordered sequence of 8-bit data bytes (octets). The LDL protocol provides improved performance for all datagram lengths under fair to very poor HF channel conditions, and under all channel conditions for short datagram lengths.

C.5.5.2 Data object types.

The terms defined in table C-LIII are used to refer to specific types of data objects in defining the LDL protocol.

TABLE C-LIII. LDL data object types.

Data object type	Definition
datagram	an arbitrary sequence of 8-bit data bytes (octets) of length *dl*, where 1 $\le$ *dl* $\le$ 16,777,216 (equal to (512 * 32,768): i.e., 512 payload data bytes per data packet times a maximum of 32,768 data packets per datagram).
data segment	a sequence of 8-bit data bytes (octets) that occur consecutively within a datagram, of length *sl* where 1 $\le$ *sl* $\le$ 512.
filled segment	a data segment of length *sl* $\le$ *pl* bytes, followed by a sequence of *pl* - *sl* fill bytes having value 0, where *pl* is the packet length established for the current LDL transfer (32, 64 … 512).
sequence number	a 17-bit data object having the format defined in table C-LVI, which indicates the position occupied by a data segment within a datagram, and, when the data segment includes the last data byte of the datagram, the number of bytes of payload data from the datagram in the data segment.
control field	an 8-bit data object reserved for future use.
data packet	the combination of a filled segment with a corresponding sequence number and control field. If the data segment contained in the filled segment is of length less than *pl* bytes (because it includes the last data byte of the datagram*)*, the value of the sequence number indicates how many of the *pl* bytes in the extended data segment contain payload data from the datagram – i.e., the value of *sl* for the data segment contained in the filled segment.
tx frame	a single data packet. Same as an LDL_DATA PDU as defined in table C-LV.
rx frame	a single data packet that was received without errors, as determined by the CRC check performed by the BW3 receiver.

C.5.5.3 Service primitives.

Table C-LIV describes the service primitives exchanged between the LDL protocol entity and one or more user processes at LDL's upper interface. Note that there is no requirement that implementations of the waveforms and protocols defined in this Appendix contain precisely these service primitives; nor are the services primitives defined below necessarily all of the service primitives that would be required in an implementation of these waveforms and protocols.

TABLE C-LIV. LDL service primitives.

Name	Attribute	Values	Description
LDL_Send_Req	Overview	LDL Send Request: generated by the user process when it has a datagram to send on an already-established HF link using the LDL protocol.	
	Parameters	datagram	datagram to be delivered, as described in table C-LIII.
		pktLength	the number *pl* of payload data bytes and fill bytes to be transmitted in each data packet transmitted using BW3, where *pl* = 32, 64 … 512. The value of pktLength should correspond to the TrafficType field-value of the TM_REQUEST PDU sent in establishing the packet traffic link: e.g., pktLength should be 64 if and only if the TrafficType field of the TM_REQUEST PDU had the value LDL_64. (See Table C-XXXVII.)
	Originator	User process	
	Preconditions	TM has just completed a successful point-to-point HF link establishment with the intended datagram recipient.	
LDL_Rcv_Req	Overview	LDL Receive Request: generated by the user process to request that LDL perform the processing required to receive an expected incoming datagram.	
	Parameters	pktLength	the number *pl* of payload data bytes or fill bytes expected to be present in each incoming data packet received by the BW3 receiver, where *pl* = 32, 64… 512. The value of pktLength should correspond to the TrafficType field-value of the TM_REQUEST PDU received when the packet traffic link was established: e.g., pktLength should be 512 if and only if the TrafficType field of the TM_REQUEST PDU had the value LDL_512. (See Table C-XXXVII.)
	Originator	User process	
	Preconditions	TM has just completed a successful point-to-point HF link establishment with the expected datagram sender.	
LDL_Rcv_Ind	Overview	LDL Receive Indication: issued by LDL when LDL has a successfully-received datagram to give to the local user process.	
	Parameters	datagram	datagram just received, as described in table C-LIII; identical to the datagram parameter-value of the original LDL_Send_Req primitive at the remote station.
	Originator	LDL	
	Preconditions	LDL has accepted an LDL_Rcv_Req service primitive from a higher-layer entity since the last outgoing or incoming datagram transfer.	
LDL_Send_Conf	Overview	LDL Send Confirm: Issued by LDL when LDL has completed successful delivery of a datagram to the remote station.	
	Parameters	(none)	
	Originator	LDL	
	Preconditions	LDL was requested to deliver the datagram by the user process by means of an LDL_Send_Req service primitive.	

TABLE C-LIV. LDL service primitives (continued).

Name	Attribute	Values	Description
LDL_Abort_Req	Overview	LDL Abort Request: used by the user process to terminate an LDL protocol data transfer that is currently in progress. The purpose of this service primitive is only to cause the LDL entity to cease attempting to send or receive the current datagram; coordinating the data transfer termination with the remote station is the responsibility of TM.	
	Parameters	(none)	
	Originator	User process	
	Preconditions	Either an outgoing or an incoming data transfer is in progress, using the LDL protocol.	
LDL_Failure_Ind	Overview	LDL Failure Indication: Issued by LDL when LDL is unable to complete delivery of an outgoing or incoming datagram.	
	Parameters	(none)	
	Originator	LDL	
	Preconditions	Either an outgoing or an incoming data transfer is in progress, using the LDL protocol.	

C.5.5.4 PDUs.

The sub-sections of this section describe the PDUs exchanged between an LDL protocol entity and its remote peer entities.

The LDL_ACK and LDL_EOM PDUs are conveyed using BW4 and thus require no special distinction from TM PDUs which are conveyed using BW1. The LDL_ACK and LDL_EOM PDUs are distinguished from one another by context: any PDU sent using BW4 in the forward direction is an LDL_EOM PDU, while any PDU sent using BW4 in the reverse direction is an LDL_ACK PDU.

C.5.5.4.1 LDL_DATA.

An LDL_DATA PDU is a tx frame as defined in table C-LIII, in which the format and contents of each data packet are as shown in table C-LV. Table C-LVI specifies the format and contents of the Sequence Number field of each data packet.

TABLE C-LV. LDL Data packet format.

Field name	Size (bits)	Values	Description
Payload	256, 512, … 4096 (fixed for each datagram transfer)	any	Contains a filled segment as defined in table C-LIII: i.e., a data segment followed by however many zero bytes are needed to fill the Payload field to the length given by PacketLength as defined in table C-LXI. Whenever the field contains fewer than PacketLength bytes of payload data from the datagram being delivered (the remainder being fill bytes with value 0), the value of the Sequence Number field indicates how many bytes of payload data are present.
Sequence Number	17 (fixed)	As defined by table C-LVI.	
Control	8 (fixed)	Reserved (set to zero); must be ignored by the receiving station.	

The fields of the LDL_DATA PDU are transmitted in order of their occurrence in table C-LV, Payload field first. The bytes of the Payload field are transmitted in the order of their occurrence in the datagram; the bits of each byte are transmitted in order of significance, starting with the most significant bit. Following the last bit of the Payload field-value, the bits of the Sequence Number field are transmitted in order of significance within the 17-bit field-value, starting with the most significant bit (bit 16). Finally, the bits of the Control field are transmitted in order of significance, most significant bit first.

LDL_DATA PDUs are transmitted using the BW3 burst waveform described by section C.5.1.6.

TABLE C-LVI. LDL Sequence number field definition.

Case	Bit 16 (EOM)	Bit 15 (SOM)	Bits 14 - 10	Bits 9 - 0
only packet in datagram	1	1	0	Payload field byte count: the number of bytes (octets) of datagram data present in the Payload field of the packet.
last packet in datagram	1	0	0	Payload field byte count (see above)
first packet in datagram	0	1	number of packets required to convey the data contents of the datagram, minus one: equal to (the least integer greater than or equal to (datagram length in bytes / PacketLength (as defined in table C-LXI)) - 1.	
packet in interior of datagram	0	0	down-counting packet sequence number: the number of packets in the current datagram, following the current packet.	

C.5.5.4.2 LDL_ACK.

The LDL_ACK PDU is used to transfer acknowledgement, in the form of an ack bit for the data packet in the immediately preceding LDL_DATA PDU, from the receiving station to the sending station in an LDL transfer. Table C-LVII specifies the format and contents of the LDL_ACK PDU.

TABLE C-LVII. LDL_ACK PDU format.

Field name	Size (bits)	Values	Description
Ack Bit	1	any	Contains one ack bit for the data packet received in an LDL_DATA PDU. The bit indicates whether the corresponding data packet was received without errors; 1 = ACK (was received successfully); 0 = NAK (not received successfully).
Complete DatagramRcvd	1	any	Contains one bit to indicate that the data packet received in the immediately previous LDL_DATA PDU is understood by the receiving station to be the last data packet of the datagram; 1 = complete datagram received; 0 = complete datagram not received.

Of the two bits of the LDL_ACK PDU, the Ack Bit is transmitted first.

LDL_ACK PDUs are transmitted using the BW4 burst waveform described by section C.5.1.7.

C.5.5.4.3 LDL_EOM.

The LDL_EOM PDU is sent from the sending station to the receiving station in an LDL transfer, to indicate that the sending station has received acknowledgements from the receiving station of every data packet in the datagram being transferred, and hence will send no more LDL_DATA PDUs for the current datagram. The LDL_EOM PDU is sent as many times as possible within the time interval defined for a forward transmission, to maximize the probability of its being received without errors. Table C-LVIII specifies the format and contents of the LDL_EOM PDU.

TABLE C-LVIII. LDL_EOM PDU format.

Field name	Size (bits)	Values	Description
Unused	1	1 (fixed)	Must be set to 1.
EOM	1	1 (fixed)	Must be set to 1.

Of the two bits of the LDL_EOM PDU, the Unused bit is transmitted first.

LDL_EOM PDUs are transmitted using the BW4 burst waveform described by section C.5.1.7.

On traffic links established for packet traffic delivered using the LDL protocol, the user process can terminate the data link transfer and use the next data link transmission time slot in either direction — i.e., the time slot for the LDL_DATA or the LDL_ACK PDU — to instead send one or more TM PDUs (described in 0.4) within the data link PDU time-slot. This means that while an LDL transfer is in progress, the receiving station must be simultaneously attempting to demodulate TM PDUs conveyed by the BW1 waveform as it is attempting to demodulate and receive LDL_DATA PDUs conveyed by BW3. The sending station must likewise simultaneously attempt to demodulate and receive TM PDUs conveyed by the BW1 waveform as it is attempting to demodulate and receive LDL_ACK PDUs conveyed by BW4. Since the duration of the BW1 burst is longer than that of the BW4 burst, if the sending station detects a

BW1 preamble during the LDL_ACK time-slot, it must skip the transmission of the next LDL_DATA PDU in order to be able to receive the remainder of the BW1 burst. If the received BW1 burst is a TM PDU, then the sending station must ensure that this PDU is received and processed by its TM sublayer.

C.5.5.5 Protocol behavior.

This section provides an informal overview of the behavior of the LDL protocol. The following paragraphs define this behavior precisely:

- C.5.5.5.1 identifies and defines the events to which LDL responds;
- C.5.5.5.2 identifies and defines the actions taken by LDL in response to these events;
- C.5.5.5.3 describes the data items used and maintained by LDL;
- C.5.5.5.4 provides a state diagram and an equivalent state transition table specifying the behavior of LDL in terms of these events, actions, and data items; and
- C.5.5.5.5 provides additional information on the timing characteristics of LDL behavior.

Data transfer by LDL begins after the TM sublayer has already established the data link connection, in so doing negotiating the fact that LDL will be used (as opposed to HDL or some other mechanism), and the precise time synchronization of data link transmissions.

In an LDL data transfer, the sending station and the receiving station alternate transmissions in the manner depicted in figure C-36, the sending station transmitting LDL_DATA PDUs containing payload data packets, and the receiving station transmitting LDL_ACK PDUs containing acknowledgement of whether or not the data packet in the preceding LDL_DATA PDU was received without error. If either station fails to receive a PDU at the expected time, it sends its own next outgoing PDU at the same time as if the incoming PDU had been received successfully. The times at which the burst waveforms conveying LDL_DATA, LDL_ACK, and LDL_EOM PDUs may be emitted are precisely stipulated. See C.5.5.5.5 for details.

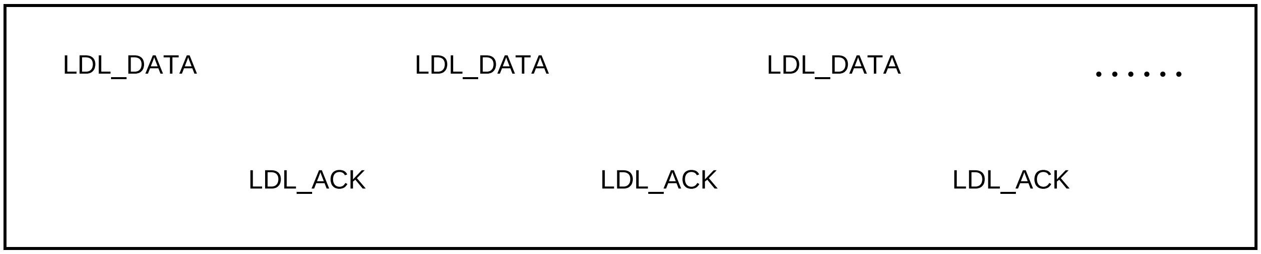


FIGURE C-36. LDL data transfer overview.

The end of a data transfer is reached when the sending station has transmitted LDL_DATA PDUs containing all of the payload data in the delivered datagram, and the receiving station has received these data without errors and has acknowledged their successful delivery. When the sending station receives an LDL_ACK PDU indicating that the entire contents of the datagram have been delivered successfully, it sends an LDL_EOM PDU repeated as many times as possible within the duration of an LDL_DATA PDU, starting at the time at which it would have otherwise transmitted the next LDL_DATA PDU, to indicate to the receiving station that the data transfer will be terminated. This link termination scenario is depicted in figure C-37. See C.5.5.5.5 for timing details.

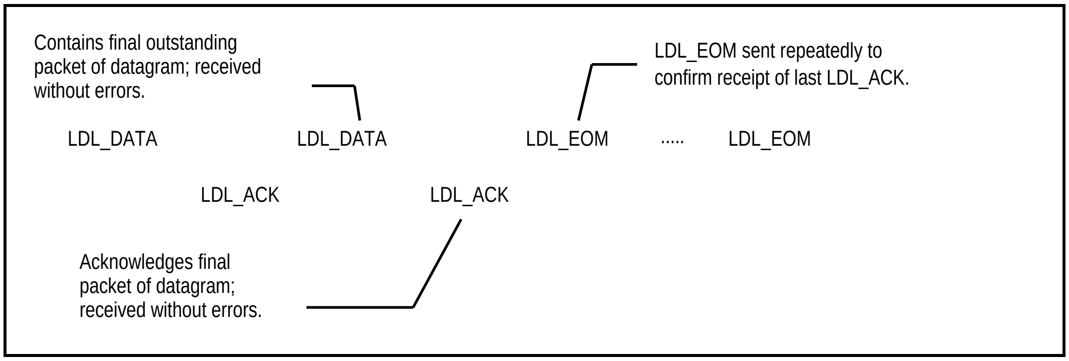

Contains final outstanding packet of datagram; received without errors.
LDL_EOM sent repeatedly to confirm receipt of last LDL_ACK.
LDL_DATA
LDL_DATA
LDL_EOM
.....
LDL_EOM
LDL_ACK
LDL_ACK
Acknowledges final packet of datagram; received without errors.

FIGURE C-37. LDL link termination scenario overview.

The definition of LDL behavior presented in the following sections includes mechanisms for dealing appropriately with the following occurrences:

- excessive number of consecutive failures to receive an expected LDL_DATA PDU;
- excessive number of consecutive failures to receive an expected LDL_ACK PDU;
- immediate termination of an ongoing data link transfer requested by the TM sublayer.

C.5.5.5.1 Events.

Table C-LIX defines the events to which the LDL entity responds. The event names are used in the state diagram and the state transition table in C.5.5.5.4, which define the behavior of the LDL protocol. Some event names refer to the receipt of PDUs from the LDL entity at a remote station; in these cases, the ‘description’ field of the table entry describes the manner in which the arrival of a PDU is accomplished through LDL’s accepting one or more service primitives from lower-layer entities at the local station. The prefix ‘R:’ in the name of an event indicates that the event is the receipt of a PDU from the remote station. ‘D:’ indicates that the event is an LDL service primitive passed down to LDL from a higher-layer entity; ‘U:’ indicates a lower-layer service primitive passed up to LDL from a lower-layer entity.

C.5.5.5.2 Actions.

Table C-LX defines the actions which the LDL entity can perform. The action name is used in the state diagrams and/or state transition tables used below to define the behavior of the LDL protocol. Some action names refer to sending PDUs to the LDL entity at a remote station; in these cases, the ‘description’ field of the table entry describes the manner in which sending of the PDU is accomplished by issuing one or more service primitives to lower-layer entities at the local station.

TABLE C-LIX. LDL events.

Event name	Description
R:LDL_DATA PDU	A BW3_Receive primitive containing an LDL_DATA PDU was accepted.
R:LDL_ACK PDU	A BW4_Receive primitive containing an LDL_ACK PDU was accepted.
R:LDL_EOM PDU	A BW4_Receive primitive containing an LDL_EOM PDU was accepted, and the LDL_EOM PDU was found to be free of errors by comparing the received PDU against the known LDL_EOM bit pattern specified in C.5.5.4.3.
D:LDL_Send_Req	An LDL_Send_Req primitive was accepted from the user process.
D:LDL_Abort_Req	An LDL_Abort_Req primitive was accepted from the user process.
U:BW1_Pre_Detect	A BW1_Pre_Detect primitive was received, indicating that the BW1 Receiver has detected the BW1 acquisition preamble, with the likely implications that the remote station has sent a TM PDU in the current LDL_ACK time slot.
AckTimeout	A valid LDL_ACK PDU was not received within the time period in which it was expected.
DataTimeout	An LDL_DATA PDU was not received within the time period in which it was expected.

TABLE C-LX. LDL actions.

Action name	Description
TxInit	Insert the outgoing datagram into TxDatagramBuf. Clear TxFrameBuf. Reset MissedAckCount to zero. Set PacketLength to the value of the pktLength parameter of the LDL_Send_Req service primitive just accepted by LDL.
S:LDL_DATA	If the TxFrameBuf is clear, construct a new outgoing data packet (as described by table C-LV) in the following manner: 1. Get the next data segment (the next PacketLength consecutive data bytes to be transmitted) from TxDatagramBuf; place these in the Payload field of the data packet. If fewer than PacketLength data bytes remain to be transmitted, place these bytes at the beginning of the Payload field; fill the remainder of the field with zero-valued bytes so that the Payload field contains a filled segment. 2. Construct a sequence number value as specified in table C-LVI; write this value into the packet's Sequence Number field. 3. Construct a control field with all 8 bits set to zero. 4. Place the data packet generated in steps 1-3 into the TxFrameBuf. Send an LDL_DATA PDU containing the tx frame in TxFrameBuf, using a BW3_Send primitive. Set the primitive's reset parameter to TRUE if this is the first transmission of a tx frame for the current datagram, and to FALSE otherwise. If an AckTimeout has occurred, increment MissedAckCount. Reset AckTimeout timer.
ProcessAck	Copy the Ack Bit-Mask field value to RxAck. If RxAck is 1, clear the TxFrameBuf. If TxDatagramBuf is not empty, set the condition MoreToSend to TRUE; otherwise set it to FALSE.

TABLE C-LX. LDL actions (continued).

Action name	Description
S:LDL_EOM	Send an LDL_EOM PDU to the remote station using BW4_Send, as many times as possible within the duration of an LDL_DATA PDU. The number of times the LDL_EOM PDU is sent depends on the value of PacketLength; for example: · if PacketLength = 64, the LDL_EOM PDU is transmitted *once* · if PacketLength = 128, the LDL_EOM PDU is transmitted *three* times · if PacketLength = 256, the LDL_EOM PDU is transmitted *five* times · if PacketLength = 512, the LDL_EOM PDU is transmitted *eleven* times. Disable AckTimeout timer.
Skip LDL_DATA slot	Do not send an LDL_DATA PDU in the next LDL_DATA time slot, so that an incoming BW1 burst can be received. However, continue the LDL slot timing, and be prepared to send an LDL_DATA PDU in the slot after the next one.
U:LDL_Send_Conf	Issue an LDL_Send_Conf primitive to the user process that requested the outgoing data transfer.
AbortTransmit	Disable AckTimeout timer; reset MissedAckCount to zero.
RxInit	Clear RxDatagramBuf, and RxFrameBuf. Reset MissedTxFrameCount to zero. Reset CompleteDatagramRcvd. Reset DataTimeout timer. Set PacketLength to the value of the pktLength parameter of the LDL_Rcv_Req service primitive just accepted by LDL.
S:LDL_ACK(ack)	Reset MissedTxFrameCount to zero. Insert the received data packet containing the Payload and Sequence Number field into RxFrameBuf. Set TxAck to 1. Move the Payload field contents of RxFrameBuf to the position in RxDatagramBuf indicated by the Sequence Number field-value. In doing this, move only payload data bytes into RxDatagramBuf; discard any fill bytes. Use the Sequence Number field-value to determine which bytes contain payload data and which are fill bytes. If the entire datagram has been received, set CompleteDatagramRcvd. Reset DataTimeout timer. Send an LDL_ACK PDU containing the TxAck and CompleteDatagramRcvd values, using BW4_Send. Note: An implementation of LDL can, without impairing compliance to this standard, provide segments of a partially-received datagram to the user process, in order of their occurrence in the original datagram at the sending station, before the entire datagram has been received. Doing so would allow a higher-layer protocol to abort an ongoing data transfer, then resume it at a later time, without having to retransmit the entire portion of the current datagram that was already delivered successfully. Note: The LDL transmitter can send duplicate packets either as a result of missing an LDL_ACK PDU, or at the end of a datagram, in order to fill the (otherwise unused) packet positions of an LDL_DATA PDU. The LDL receiver is required to inspect the sequence number of each data packet received without errors, and to use the sequence numbers to identify and discard duplicate packets.
S:LDL_ACK(nak)	Clear TxAck. Send an LDL_ACK PDU containing the TxAck and CompleteDatagramRcvd values, using BW4_Send. If a DataTimeout has occurred, increment MissedTxFrameCount. Reset the DataTimeout timer.
U:LDL_Rcv_Ind	Send an LDL_Rcv_Ind service primitive to the user process.
AbortReceive	Disable DataTimeout timer; reset MissedTxFrameCount to zero.

C.5.5.5.3 LDL Data.

Table C-LXI defines the data items used and maintained by LDL, including buffers, counters, timers, configuration parameters, and so forth. These data items are referred to by the names assigned to them here, in the definitions of LDL events and actions presented in the preceding sections.

TABLE C-LXI. LDL data items.

Data item	Description
TxDatagramBuf	buffer storing the data contents of an outgoing datagram which have not yet been inserted into a tx frame for transmission.
TxFrameBuf	buffer storing the outgoing tx frame (an LDL_DATA PDU).
RxDatagramBuf	buffer storing the data contents of an incoming datagram which have been received thus far, in which the complete incoming datagram is re-assembled in correct order.
RxFrameBuf	buffer storing the most recent rx frame that was received without errors (as determined by the CRC check performed by the BW3 receiver).
TxAck	ack flag to be transmitted to the remote station. TxAck = 1 indicates that the data packet in the most recently received rx frame was received without errors.
RxAck	ack flag received in an LDL_ACK PDU from the remote station. RxAck = 1 indicates that the remote station received the data packet in the previously-transmitted frame without errors.
MissedAckCount	count of consecutive failures to receive an LDL_ACK PDU in the time period in which one was expected.
MissedTxFrameCount	count of consecutive failures to receive an LDL_DATA PDU (a tx frame) in the time period in which one was expected.
AckTimeout	timer used to time the duration of the interval in which receipt of an LDL_ACK PDU is expected; fires when the interval expires.
DataTimeout	timer used to time the duration of the interval in which receipt of an LDL_DATA PDU is expected; fires when the interval expires.
MAX_MISSED_ACKS	LDL configuration parameter specifying the maximum number of consecutive missed LDL_ACK PDUs that can occur without causing the LDL transmitter to terminate the data link transfer. The value of this parameter is not stipulated by this specification, since it is not required for interoperability that this parameter have identical values in both the sending and receiving stations.
MAX_MISSED_TX_FRAMES	LDL configuration parameter specifying the maximum number of consecutive missed LDL_DATA PDUs that can occur without causing the LDL receiver to terminate the data link transfer. The value of this parameter is not stipulated by this specification, since it is not required for interoperability that this parameter have identical values in both the sending and receiving stations.
CompleteDatagramRcvd	flag maintained by the receiving station indicating whether or not the entire datagram has been successfully received.
MoreToSend	Boolean condition variable: is TRUE if and only if an outgoing datagram transfer is in progress, and there are one or more data packets in the datagram for which the local station has not yet received an acknowledgement of their receipt without errors by the remote station.
PacketLength	number of payload data bytes and fill bytes (if any) carried within each LDL forward transmission in the current datagram transfer; possible values are 64, 128, 256, and 512. The value of PacketLength is determined by the pktLength parameter of the LDL_Send_Req or LDL_Rcv_Req service primitive that was accepted by LDL just prior to the start of the datagram transfer.

SUPERSEDES PAGE 424 OF MIL-STD-188-141B.

C.5.5.5.4 LDL Behavior definition.

For the reader's convenience, two equivalent representations of the behavior of the LDL protocol are provided in this section: the state transition table in table C-LXII, and the state diagram in figure C-38.

Both table C-LXII and figure C-38 specify the behavior of LDL in terms of the events defined in C.5.5.5.1 and the actions defined in C.5.5.5.2. The conditions gating certain transitions are specified in terms of the data items defined in C.5.5.5.3.

In the state diagram, each state transition is labeled with an event, an optional condition, and zero or more actions. This indicates that the state transition occurs whenever the event occurs and the condition obtains (is TRUE), causing the associated actions to be performed. On figure C-38,

- the name of each event is shown in brackets preceded by the letter 'E';
- the description of each condition is shown in brackets preceded by the letter 'C'; and
- the names of the actions associated with a transition are shown in brackets preceded by the letter 'A'.

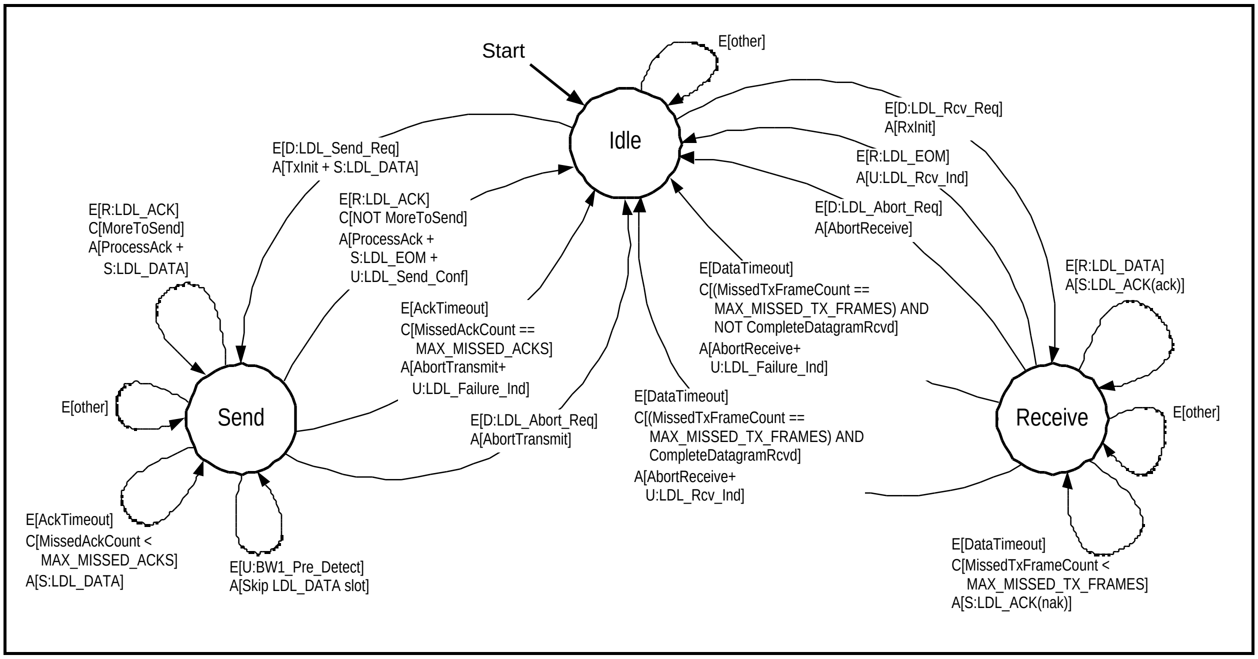


FIGURE C-38. LDL state diagram.

Where a transition is labeled with two or more events, this indicates that the transition occurs whenever any of the events occurs.

TABLE C-LXII. LDL state transition table.

State	Event	Condition	Action	Next State
Idle	D:LDL_Send_Req		TxInit + S:LDL_DATA	Send
	D:LDL_Rcv_Req		RxInit	Receive
	other		none	Idle
Send	R:LDL_ACK	MoreToSend	ProcessAck + S:LDL_DATA	Send
	R:LDL_ACK	NOT MoreToSend	ProcessAck + S:LDL_EOM + U:LDL_Send_Conf	Idle
	U:BW1_Pre_Detect		Skip LDL_Data slot	Send
	AckTimeout	MissedAckCount < MAX_MISSED_ACKS	S:LDL_DATA	Send
	AckTimeout	MissedAckCount == MAX_MISSED_ACKS	AbortTransmit + U:LDL_Failure_Ind	Idle
	D:LDL_Abort_Req		AbortTransmit	Idle
	other		none	Send
Receive	R:LDL_DATA		S:LDL_ACK(ack)	Receive
	R:LDL_EOM		U:LDL_Rcv_Ind	Idle
	DataTimeout	MissedTxFrameCount < MAX_MISSED_TX_FRAMES	S:LDL_ACK(nak)	Receive
	DataTimeout	(MissedTxFrameCount == MAX_MISSED_TX_FRAMES) AND NOT CompleteDatagramRcvd	AbortReceive + U:LDL_Failure_Ind	Idle
	DataTimeout	(MissedTxFrameCount == MAX_MISSED_TX_FRAMES) AND CompleteDatagramRcvd	AbortReceive + U:LDL_Rcv_Ind	Idle
	D:LDL_Abort_Req		AbortReceive	Idle
	other		none	Receive

C.5.5.5.5 Timing characteristics.

See C.5.3.5.5, which includes an analysis of the timing of the LDL protocol in conjunction with TM protocol timing.

C.5.6 CLC.

C.5.6.1 Overview.

The CLC monitors and coordinates traffic on an established circuit link. It provides a simple listen-before-transmit access control mechanism:

- Transmission of new outgoing traffic is inhibited whenever the CLC detects that the circuit link is busy, due to either traffic being received from another station, or traffic currently being transmitted by the local station.
- At the end of each outgoing or incoming traffic transmission, the CLC continues to inhibit transmission of new outgoing traffic for the duration of a backoff interval.

In addition, the CLC provides a traffic timeout indication when an interval of a specified duration elapses in which no outgoing or incoming traffic is detected on the circuit link, allowing the traffic link to be terminated when no longer required.

The CLC is employed only on circuit links.

C.5.6.2 Service primitives.

Table C-LXIII describes the service primitives exchanged between the CLC and one or more user processes at the CLC's upper interface. Note that there is no requirement that implementations of the waveforms and protocols defined in this Appendix contain precisely these service primitives; nor are the services primitives defined below necessarily all of the service primitives that would be required in an implementation of these waveforms and protocols.

TABLE C-LXIII. CLC service primitives.

Name	Attribute	Values	Description
CLC_Active_Req	Overview	CLC Active Request: issued to CLC by the user process to request that CLC begin monitoring and arbitration of access to the currently-established circuit link, using the indicated priority level in its backoff mechanism. CLC sets its data item TrafficPriority to the value of the *prio* parameter.	
	Parameters	prio	priority level of waiting outgoing traffic (if any); value is one of · P2P: a point-to-point circuit link is being established, which is treated as a special case by CLC: the backoff delay depends on which station has just transmitted, rather than on traffic priority · TM: TM is waiting to send a TM-PDU · HIGHEST: highest priority level for user traffic · HIGH · ROUTINE · LOW: lowest priority level for user traffic; also serves as default value when no outgoing traffic is pending.
	Originator	user process	
	Preconditions	CLC is Idle; a circuit link was just established by the Traffic Manager.	

TABLE C-LXIII. CLC service primitives (continued).

Name	Attribute	Values	Description
CLC_Idle_Req	Overview	CLC Idle Request: issued to CLC by the user process to request that CLC cease to monitor and arbitrate access to the current circuit link.	
	Parameters	none	
	Originator	user process	
	Preconditions	CLC is presently active: i.e., not presently residing in its Idle state.	
CLC_Set_Priority	Overview	issued to CLC by the user process to request that CLC use the indicated outgoing traffic priority level in its backoff mechanism. CLC sets its data item TrafficPriority to the value of the *prio* parameter.	
	Parameters	prio	priority level of waiting outgoing traffic (if any); value is one of · P2P: a point-to-point circuit link is being established, which is treated as a special case by CLC: the backoff delay depends on which station has just transmitted, rather than on traffic priority · TM: TM is waiting to send a TM-PDU · HIGHEST: highest priority level for user traffic · HIGH · ROUTINE · LOW: lowest priority level for user traffic; also serves as default value when no outgoing traffic is pending.
	Originator	user process	
	Preconditions	none: can be accepted by CLC in any state.	
CLC_Idle_Ind	Overview	CLC Idle Indication: issued to the user process by CLC, to indicate that CLC is ceasing to monitor and arbitrate access to the current circuit link due to occurrence of a traffic timeout (no link traffic detected over a time interval of a specific duration).	
	Parameters	none	
	Originator	CLC	
	Preconditions	CLC is presently active: i.e., not presently residing in its Idle state.	
CLC_Busy_Ind	Overview	CLC Busy Indication: issued to the user process by CLC, to indicate that CLC considers the circuit link to be busy — i.e., unavailable for new traffic because of a traffic exchange currently in progress, or because a backoff period following a traffic exchange has not yet expired.	
	Parameters	none	
	Originator	CLC	
	Preconditions	CLC is either · newly-activated: i.e., the most recent service primitive passed between CLC and the user process was a CLC_Active_Req primitive; or · indicating that the traffic link is available: i.e., the most recent service primitive passed between CLC and the user process was a CLC_Avail_Ind primitive.	
CLC_Avail_Ind	Overview	CLC Available Indication: issued to the user process by CLC, to indicate that CLC considers the circuit link to be available for new traffic.	
	Parameters	none	
	Originator	CLC	
	Preconditions	CLC is indicating that the traffic link is busy: i.e., the most recent service primitive passed between CLC and the user process was a CLC_Busy_Ind primitive.	

C.5.6.3 PDUs.

The CLC does not exchange PDUs with a remote peer entity.

C.5.6.4 Protocol behavior.

The following paragraphs define the behavior of the CLC:

- C.5.6.4.1 identifies and defines the events to which the CLC responds;
- C.5.6.4.2 identifies and defines the actions taken by the CLC in response to these events;
- C.5.6.4.3 describes the data items used and maintained by the CLC; and
- C.5.6.4.4 provides a state diagram specifying the behavior of the CLC in terms of these events, actions, and data.

C.5.6.4.1 Events.

Table C-LXIV defines the events to which the CLC responds. The event names are used in the state diagram in C.5.6.4.4, which defines the behavior of the CLC.

C.5.6.4.2 Actions.

Table C-LXV defines the actions which the CLC can perform. The action name is used in the state diagram used below to define the behavior of the CLC.

TABLE C-LXIV. CLC events.

event name	description
D:CLC_Active_Req (prio)	CLC_Active_Req primitive issued by user process, with the indicated value for its *prio* parameter.
D:CLC_Idle_Req	CLC_Idle_Req primitive issued by user process.
D:CLC_Set_Priority (prio)	CLC_Set_Priority primitive issued by user process, with the indicated value for its *prio* parameter. CLC sets its data item TrafficPriority to the value of the *prio* parameter.
ModemRTS	signal indicating that the local station's modem is starting to modulate data to be transmitted the current circuit link.
AudioRTS	signal indicating presence of an outgoing audio signal to be transmitted on the current circuit link, such as a handset keyline assertion.
ModemEOTx	signal indicating that a transmission of modem data by the local station has been completed.
AudioEOTx	signal indicating that transmission of an outgoing audio signal has ended due to, for instance, de-assertion of the handset keyline.

TABLE C-LXV. CLC actions.

Action name	Description
ModemDetect	signal indicating that the local station's modem has detected incoming data signalling; equivalent to a signal presence indication. As a minimum requirement for compliance with this standard, the CLC shall employ a signal detector capable of detecting MIL-STD-188-110 serial tone modem signalling, including both the preamble and data portions of the modem waveform. As a design objective, it is also desirable that the signal detector be able to detect as many as possible of the signalling types corresponding to the traffic types enumerated in table C-XXII, as well as the following: · CW signalling · frequency shift keying (FSK) signalling.
AudioDetect	signal indicating that the local station has detected incoming audio (typically voice) signalling on the circuit link. The capability to detect SSB analog voice is a requirement for compliance with this standard.
ModemEORx	signal indicating that the local station's modem has detected the MIL-STD-188-110 serial tone End-Of-Message (EOM) sequence.
ModemSigLoss	signal indicating that the local station's modem has declared signal absence after previously having acquired an incoming modem transmission, without having detected the modem's EOM sequence.
AudioSigLoss	signal indicating that the local station has determined that an incoming audio signal on the circuit link has ceased to be present.
TrafficTimeout	timeout event generated by TrafficTimer when the local station has not detected incoming or outgoing traffic on the current circuit link within an interval of duration ≥ TRAFFIC_TIMEOUT_INTVL
BackoffTimeout	timeout event generated by BackoffTimer when the backoff interval following the most recent detected incoming or outgoing traffic transmission has expired.
ReacqTimeout	timeout event generated by ReacqTimer when the local station has not detected resumption of reception of an interrupted incoming modem or audio signal within an interval of duration ≥ REACQ_TIMEOUT_INTVL.
SetPriority	Set TrafficPriority to the value of the *prio* parameter of a CLC_Active_Req or CLC_Set_Priority service primitive.
StartTrafficTimer	Set TrafficTimer to TRAFFIC_TIMEOUT_INTVL.
StopTrafficTimer	Disable TrafficTimer.
StartBackoffTimer	Set BackoffTimer to BACKOFF_TIMEOUT_INTVL.
StopBackoffTimer	Disable BackoffTimer.
StartReacqTimer	Set ReacqTimer to REACQ_TIMEOUT_INTVL.
StopReacqTimer	Disable ReacqTimer.
U:CLC_Idle_Ind	Issue a CLC_Idle_Ind service primitive to the user process.
U:CLC_Avail_Ind	Issue a CLC_Avail_Ind service primitive to the user process.
U:CLC_Busy_Ind	Issue a CLC_Busy_Ind service primitive to the user process.

C.5.6.4.3 Data.

Table C-LXVI defines the data items used and maintained by CLC, including buffers, counters, timers, configuration parameters, and so forth. These data items are referred to by the names assigned to them here, in the definitions of CLC events and actions presented in the preceding sections. These data items are used in this specification only as expository devices; it is not required for compliance that an implementation contain these data items in the form described here.

TABLE C-LXVI. CLC data items.

Data item	Description
BackoffTimer	down-counting timer used to time BackoffTimeouts. Is set with a duration value and, unless reset before the timeout interval expires, counts down to zero, then generates a BackoffTimeout event.
ReacqTimer	down-counting timer used to time ReacqTimeouts. Is set with a duration value and, unless reset before the timeout interval expires, counts down to zero, then generates a ReacqTimeout event.
TrafficTimer	down-counting timer used to time TrafficTimeouts. Is set with a duration value and, unless reset before the timeout interval expires, counts down to zero, then generates a TrafficTimeout event.
TrafficPriority	the priority level of the pending outgoing traffic, if any.
TRAFFIC_TIMEOUT_INTVL	constant configuration parameter: duration of the time interval after which a traffic timeout will occur if no incoming or outgoing traffic is detected on the current circuit link. The value of this parameter is not stipulated as a requirement for interoperability.
BACKOFF_TIMEOUT_INTVL	duration of the time interval after which a backoff timeout will occur if no incoming traffic is detected on the current circuit link. Determines the interval following each outgoing or incoming transmission on the circuit link during which the local station will listen for traffic on the circuit before transmitting. The interval duration is always zero for pending analog or digital voice traffic (preventing annoying delays in voice answer-back operation). For data traffic, the interval duration is selected randomly from one of five values; the relative probabilities of the possible duration values are determined by the priority level of the pending outgoing data traffic (if any), as specified in table C-LXVII. If no traffic is pending, the interval duration is set to zero.
REACQ_TIMEOUT_INTVL	constant configuration parameter: duration of the time interval after which a reacquisition timeout will occur if an incoming modem or audio traffic signal is not detected on the traffic circuit. The value of this parameter is not stipulated as a requirement for interoperability; a suggested default value is 800 milliseconds.

TABLE C-LXVII. Backoff interval duration probabilities.

Priority	0 ms	250 ms	450 ms	650 ms	850 ms
P2P (after transmit or on start of link if not initiator)					100%
P2P (after receive or on start of link if initiator)	100%				
TM	50%	50%			
HIGHEST	50%	50%			
HIGH		50%	50%		
ROUTINE			50%	50%	
LOW				50%	50%

The backoff scheme using these interval durations is intended to accomplish the following:

- on a broadcast or multicast link, at the end of each transmission, stations having traffic of higher priority get earlier opportunities to seize the link. Mapping each priority level to two backoff interval durations serves to reduce congestion when multiple stations have pending traffic at the same priority level.

- TM PDUs (expected to be TM_TERM ABORT or SIGN_OFF PDUs) are treated as being of priority equal to the HIGHEST priority level for user traffic.
- Point-to-point traffic links are used fairly: at the end of each transmission, the receiving station gets the first opportunity to seize the link, based on the expectation that point-to-point link users will tend to want to send traffic in an alternating fashion.
- When a circuit traffic link is initially established, the initiator of the link gets the first opportunity to transmit on it.

C.5.6.4.4 Behavior definition.

The state diagram in figure C-39 specifies the behavior of the CLC in terms of the events defined in C.5.6.4.1 and the actions defined in C.5.6.4.2.

In the state diagram, each state transition is labeled with an event, an optional condition, and zero or more actions. This indicates that the state transition occurs whenever the event occurs and the condition obtains (is TRUE), causing the associated actions to be performed. In the diagram,

- the name of each event is shown in square brackets preceded by the letter ‘E’;
- the description of each condition is shown in square brackets preceded by the letter ‘C’; and
- the names of the actions associated with a transition are shown in square brackets preceded by the letter ‘A’.

Where a transition is labeled with two or more events, this indicates that the transition occurs whenever any of the events occurs.

In its Idle state, the CLC does not monitor link traffic or control access to the link. When a circuit link is established, the CLC of the station that initiated the link is placed in its Ready state and begins to monitor traffic on the circuit link. From Ready it proceeds to its Transmit or its Receive state, respectively, when outgoing or incoming traffic is detected. When the traffic ends, the CLC proceeds into its Backoff state where it waits for the duration of a backoff interval before returning to its ready state. If incoming signal presence is lost during reception of incoming modem signalling, the CLC enters its Signal Reacq state, where it remains until either incoming signal presence is reacquired, or a ReacqTimeout event occurs causing the CLC to decide that the incoming traffic has ended and proceed to its Backoff state.

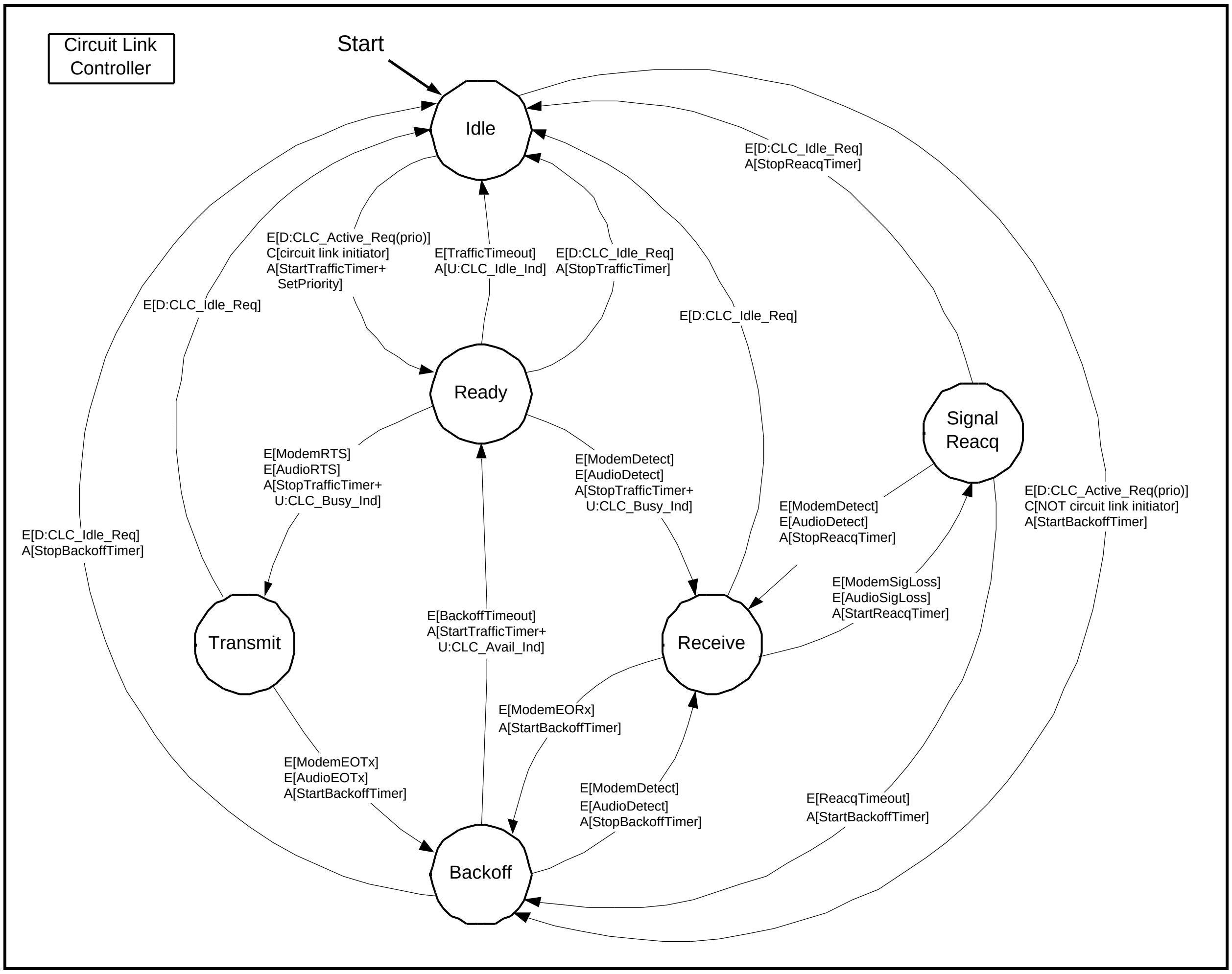

Circuit Link Controller
Start
Idle
Ready
Transmit
Receive
Backoff
Signal Reacq
E[D:CLC_Idle_Req] A[StopReacqTimer]
E[D:CLC_Active_Req(prio)] C[circuit link initiator] A[StartTrafficTimer+ SetPriority]
E[TrafficTimeout] A[U:CLC_Idle_Ind]
E[D:CLC_Idle_Req] A[StopTrafficTimer]
E[D:CLC_Idle_Req]
E[D:CLC_Idle_Req]
E[ModemRTS] E[AudioRTS] A[StopTrafficTimer+ U:CLC_Busy_Ind]
E[ModemDetect] E[AudioDetect] A[StopTrafficTimer+ U:CLC_Busy_Ind]
E[ModemDetect] E[AudioDetect] A[StopReacqTimer]
E[D:CLC_Active_Req(prio)] C[NOT circuit link initiator] A[StartBackoffTimer]
E[D:CLC_Idle_Req] A[StopBackoffTimer]
E[ModemSigLoss] E[AudioSigLoss] A[StartReacqTimer]
E[BackoffTimeout] A[StartTrafficTimer+ U:CLC_Avail_Ind]
E[ModemEORx] A[StartBackoffTimer]
E[ModemEOTx] E[AudioEOTx] A[StartBackoffTimer]
E[ModemDetect] E[AudioDetect] A[StopBackoffTimer]
E[ReacqTimeout] A[StartBackoffTimer]

FIGURE C-39. CLC state diagram.

When a circuit link is established, the CLC of each station that did not initiate the link enters its Backoff state, giving the link initiator the first opportunity to transmit on the circuit link.

Note that on the occurrence of any event not shown here, the CLC will take no action and remain in its current state.

C.5.7 Examples.

Figure C-40 through figure C-45 illustrate the operation of the protocols described in this appendix.

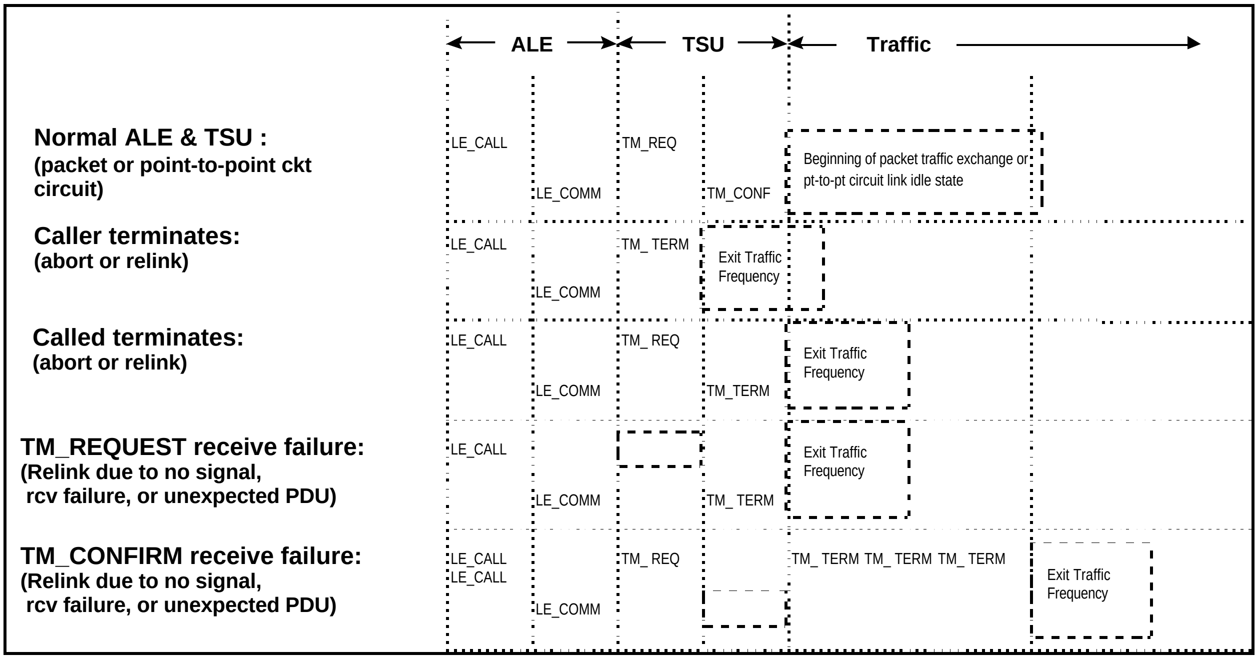
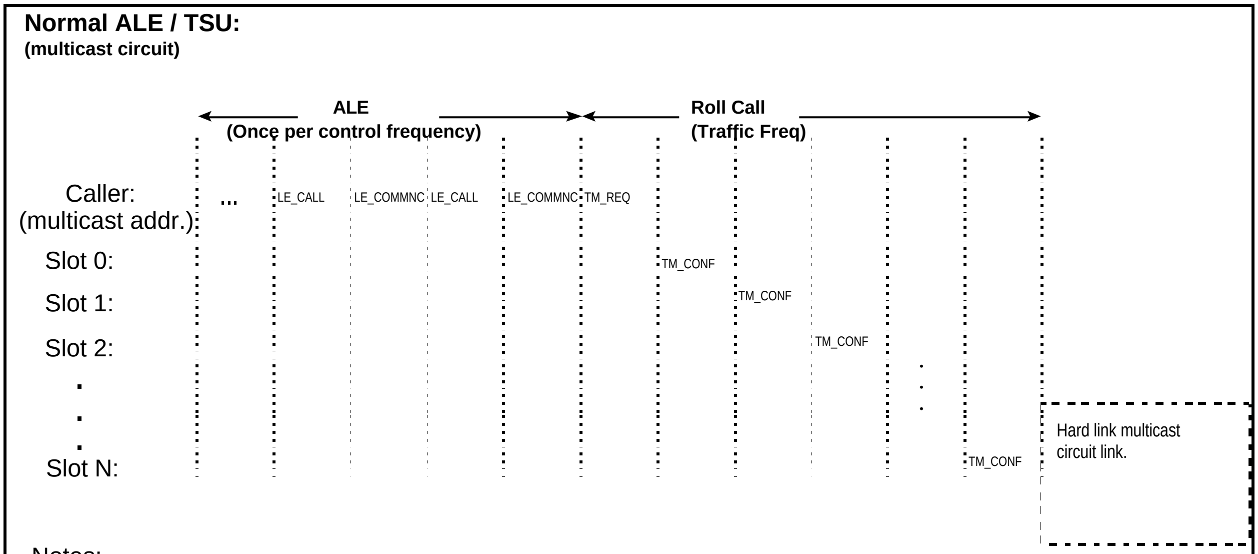


FIGURE C-40. ALE/TSU scenarios: packet and point-to-point circuit links.



Notes:
- ALE shown is only the last 2 attempts before caller diverts to the traffic frequency. The ALE is repeated on all c frequencies.
- Caller responds in his own slot, based on his address.
- If Slot k is not allocated, then no station shall respond in slot k.
- N shall be known to all stations and, if possible, N shall include spare stations for late entry.
- No other TM PDUs shall be transmitted until the TSU process has completed as shown.

FIGURE C-41. ALE/TSU scenario: multicast circuit links.

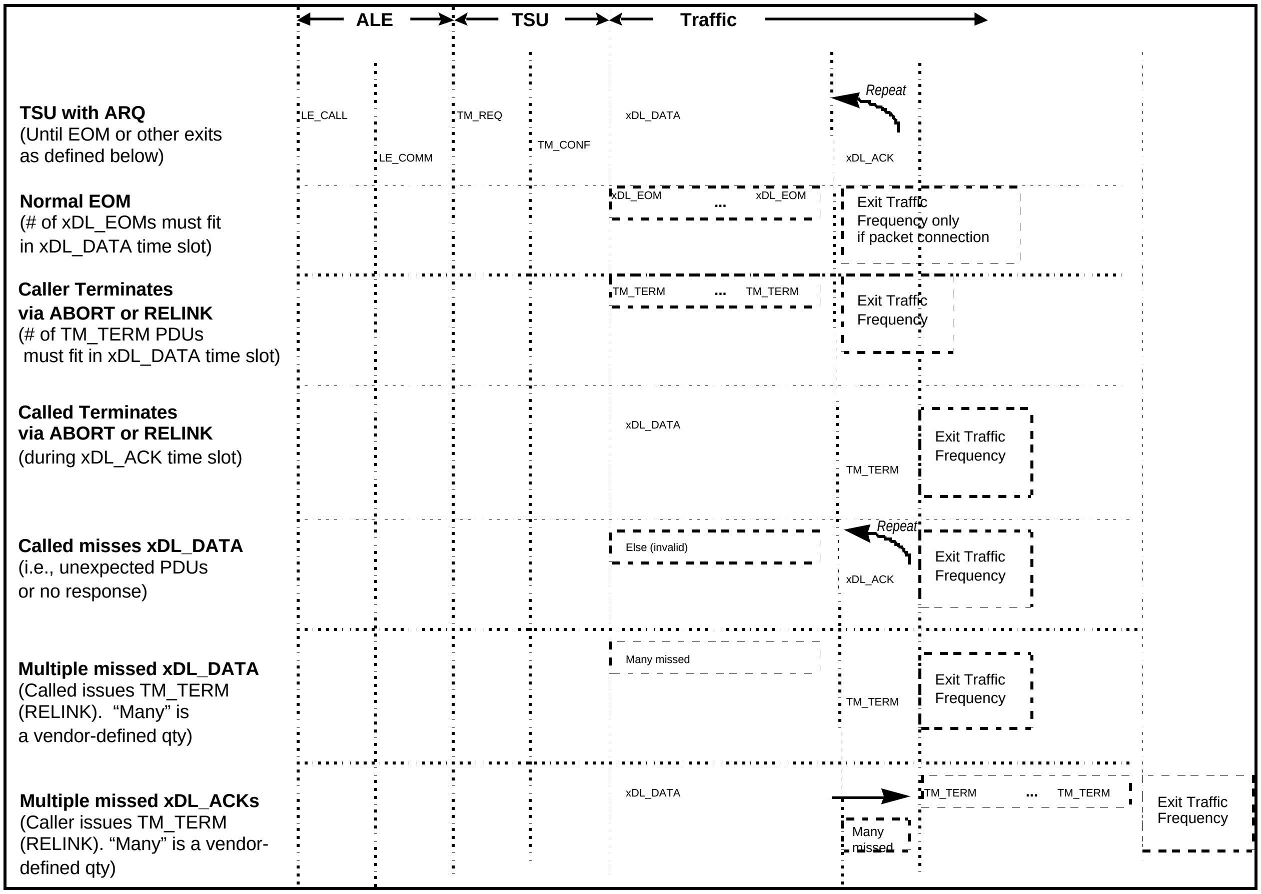
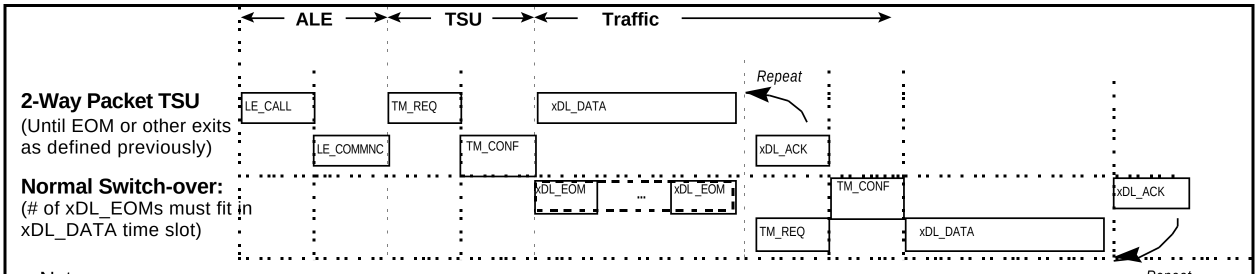


FIGURE C-42. Packet traffic link termination scenarios.



Notes:

- Anomaly scenarios and link exit criteria are the same as defined for one-way Packet TSU

FIGURE C-43. Two-way packet link scenarios.

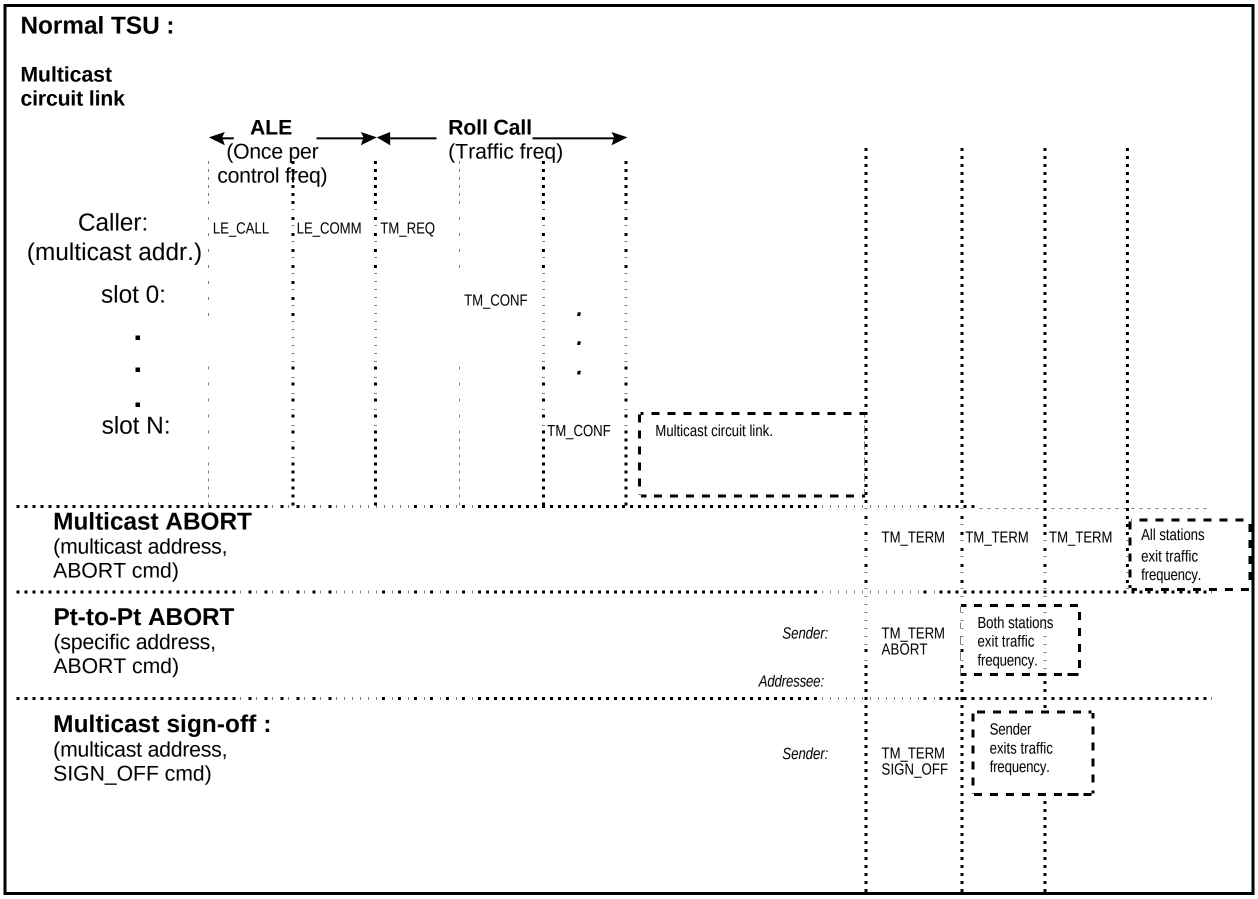


FIGURE C-44. Link termination scenarios: multicast circuit links.

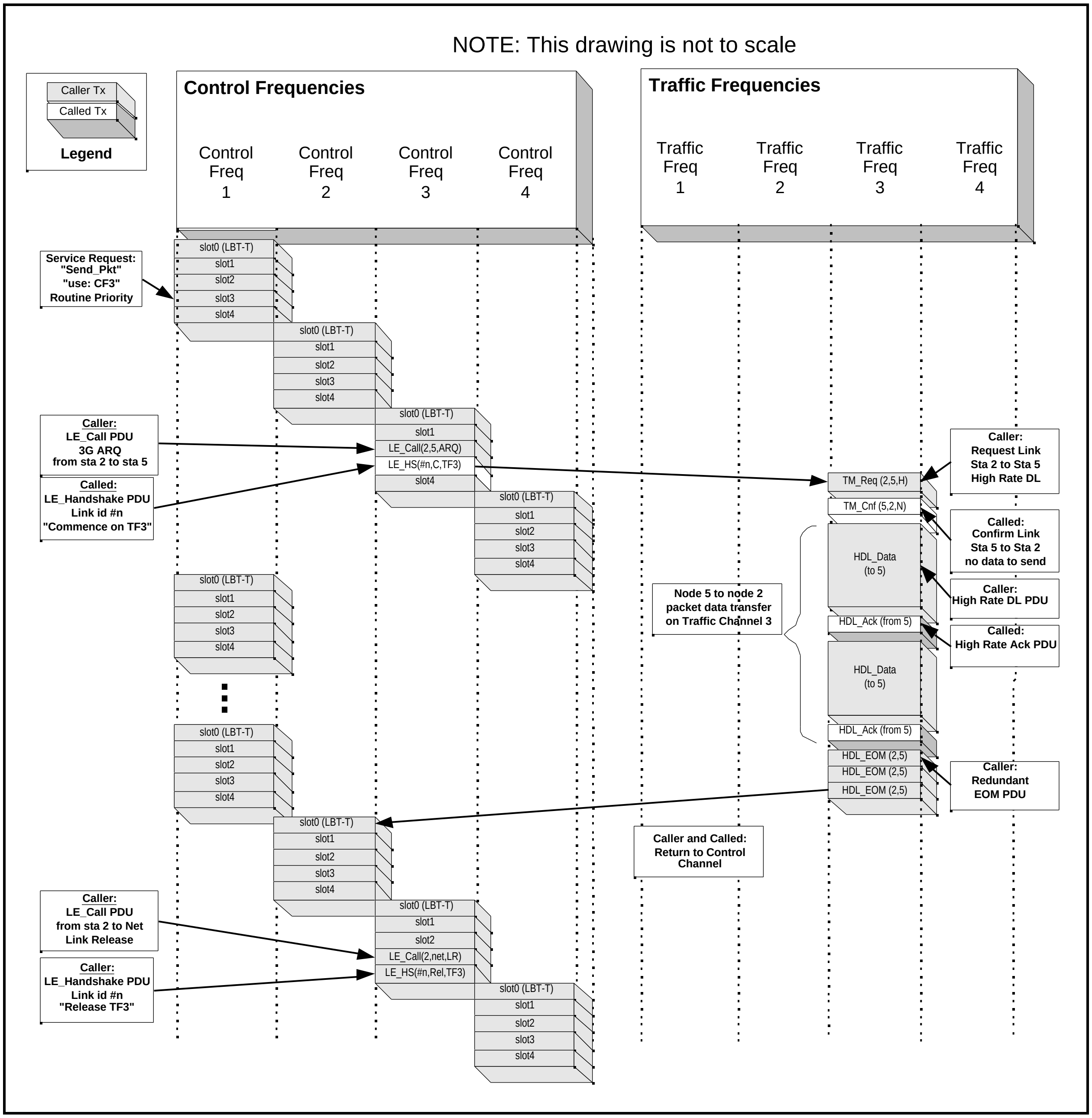

NOTE: This drawing is not to scale
Caller Tx
Called Tx
Legend
Control Frequencies
Control Freq 1
Control Freq 2
Control Freq 3
Control Freq 4
Traffic Frequencies
Traffic Freq 1
Traffic Freq 2
Traffic Freq 3
Traffic Freq 4
slot0 (LBT-T)
slot1
slot2
slot3
slot4
Service Request: "Send_Pkt" "use: CF3" Routine Priority
Caller: LE_Call PDU 3G ARQ from sta 2 to sta 5
Called: LE_Handshake PDU Link id #n "Commence on TF3"
LE_Call(2,5,ARQ)
LE_HS(#n,C,TF3)
TM_Req (2,5,H)
TM_Cnf (5,2,N)
HDL_Data (to 5)
HDL_Ack (from 5)
HDL_EOM (2,5)
Caller: Request Link Sta 2 to Sta 5 High Rate DL
Called: Confirm Link Sta 5 to Sta 2 no data to send
Caller: High Rate DL PDU
Called: High Rate Ack PDU
Node 5 to node 2 packet data transfer on Traffic Channel 3
Caller: Redundant EOM PDU
Caller and Called: Return to Control Channel
Caller: LE_Call PDU from sta 2 to Net Link Release
Caller: LE_Handshake PDU Link id #n "Release TF3"
LE_Call(2,net,LR)
LE_HS(#n,Rel,TF3)

FIGURE C-45. Packet linking and traffic exchange: on-air signalling overview.

C.6 NOTES

(This section contains information of a general or explanatory nature which may be helpful, but is not mandatory.)

C.6.1 Changes from previous issue.

The margins of this standard are marked with vertical lines to indicate where changes from the previous issue were made. This was done as a convenience only and the Government assumes no liability whatsoever for any inaccuracies in these notations. Bidders and contractors are cautioned to evaluate the requirements of this document based on the entire content irrespective of the marginal notations and relationship to the last previous issue.

TABLE OF CONTENTS

TABLE OF CONTENTS
(continued)

FIGURES

APPLICATION PROTOCOLS FOR HF RADIO NETWORKS

E.1 GENERAL.

E.1.1 Scope.

This appendix contains the requirements for the prescribed protocols and directions for the implementation and use of communications applications such as file transfer and electronic mail in HF radio networks.

E.1.2 Applicability.

Application protocols provide advanced technical capabilities via automated HF radio systems. None of the features and functions described in this appendix are mandatory requirements for the user in the acquisition of an HF radio system. However, if the user requires the features and functions described herein, they shall be provided in accordance with the technical parameters specified in this appendix.

E.2 APPLICABLE DOCUMENTS.

E.2.1 General.

The documents listed in this section are specified in sections E.3, E.4, and E.5 of this standard. This section does not include documents cited in other sections of this standard or recommended for additional information or as examples. While every effort has been made to ensure the completeness of this list, document users are cautioned that they must meet all specified requirements documents cited in sections E.3, E.4, and E.5 of this standard, whether or not they are listed.

E.2.2 Government documents.

The following specifications, standards, and handbooks form a part of this document to the extent specified herein. Unless otherwise specified, the issues of these documents are those listed in the issue of the Department of Defense Index of Specifications and Standards (DoDISS) and supplement thereto, cited in the solicitation.

STANDARDS

FEDERAL

FED-STD-1037	Telecommunications: Glossary of Telecommunication Terms

Unless otherwise indicated, copies of federal and military specifications, standards, and handbooks are available from the Naval Publications and Forms Center, ATTN: NPODS, 5801 Tabor Avenue, Philadelphia, PA 19120-5099.

INTERNATIONAL STANDARDIZATION DOCUMENTS

NORTH ATLANTIC TREATY ORGANIZATION (NATO)
STANDARDIZATION AGREEMENTS (STANAG)

STANAG 5066	Profile for High Frequency (HF) Radio Data Communications

E.2.3 Non-Government publications.

The following documents form a part of this document to the extent specified herein. Unless otherwise specified, the issues of the documents which are DoD adopted are those listed in the issues of the DODISS cited in the solicitation. Unless otherwise specified, issues of documents not listed in the DODISS are the issues of the documents cited in the solicitation (see 6.3).

INTERNET DOCUMENTS

RFC-854	Telnet Protocol specification
RFC-821	Simple Mail Transfer Protocol
RFC-822	Standard for the format of ARPA Internet text messages
RFC-959	File Transfer Protocol
RFC-1651	SMTP Service Extensions
RFC-1730	Internet Message Access Protocol - Version 4
RFC-1939	Post Office Protocol - Version 3
RFC-1950	ZLIB Compressed Data Format Specification version 3.3
RFC-1951	DEFLATE Compressed Data Format Specification version 1.3
RFC-1952	GZIP file format specification version 4.3
RFC-2068	Hypertext Transfer Protocol - HTTP/1.1
RFC-2197	SMTP Service Extension for Command Pipelining
RFC-2045	Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies
RFC-2046	Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types
RFC-2047	Multipurpose Internet Mail Extensions (MIME) Part Three: Message Header Extensions for Non-ASCII Text
RFC-2049	Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples
RFC-2246	The TLS Protocol Version 1.0
RFC-2401	Security Architecture for IP

(Internet documents may be obtained from http://www.rfc-editor.org/rfc.html.)

E.2.4 Order of precedence.
In the event of a conflict between the text of this document and the references cited herein, the text of this document takes precedence. Nothing in this document, however, supersedes applicable laws and regulations unless a specific exemption has been obtained.

E.3 DEFINITIONS.

E.3.1 Standard definitions and acronyms.
None.

E.3.2 Abbreviations and acronyms.
The abbreviations and acronyms used in this document are defined below. Those listed in the current edition of FED-STD-1037 have been included for the convenience of the reader.

ALE	automatic link establishment
ALM	automatic link maintenance
ARQ	automatic repeat request
CFTP	compressed file transfer protocol
COMSEC	communications security
e-mail	electronic mail
FTP	file transfer protocol
HF	high frequency
HMTP	HF mail transfer protocol
HTTP	hypertext transfer protocol
IMAP4	internet mail access protocol – version 4
IP	internet protocol
Ipsec	IP security
LAN	local area network
MTA	mail transfer agent
PDU	protocol data unit
POP3	post office protocol – version 3
SAP	service access point
SMTP	simple mail transfer protocol
SSL	secure sockets layer
TCP	transmission control protocol
TLS	transport layer security
UDP	user datagram protocol
WAN	wide area network

E.4 GENERAL REQUIREMENTS

E.4.1 Introduction.
Data applications such as electronic mail (e-mail), file transfer, remote login, and limited web browsing can employ HF links either for communication among hosts directly connected to HF stations, or for wireless access to other data networks.

E.4.1.1 Required HF subnetwork service.

Interoperation among applications in use at different stations requires that the applications *and all supporting protocols* at the stations interoperate. Performance will then be determined by how well the protocol stacks work with each other and with the HF medium. Systems that implement any application from this appendix shall implement the HF subnetwork service described in E.4.1.1.1 and E.4.1.1.2 to convey the corresponding application protocol data units (PDUs) over the HF medium.

E.4.1.1.1 Required HF subnetwork protocols.

To simplify the task of ensuring interoperability among applications using the HF medium, a small number of lower-layer protocols is approved for use with the application protocols specified in this appendix:

- HF radio in accordance with MIL-STD-188-141.
- EITHER

  the second-generation (2G) HF data link suite: Automatic Link Establishment (ALE) in accordance with Appendix A and the Automatic Repeat Request (ARQ) data traffic protocol in accordance with Appendix G,
  — OR —
  the third-generation (3G) HF data link suite: ALE, ARQ, and Automatic Link Maintenance (ALM) in accordance with Appendix C.

E.4.1.1.2 Required HF subnetwork interface.

Application clients of the HF subnetwork shall interact with the HF subnetwork using the Service Data Units (SDUs) specified in STANAG 5066 Annex A: Subnetwork Interface Sublayer. As a design objective, an Ethernet interface should be provided for exchange of SDUs with clients external to the subnetwork interface device.

Subnetwork interface PDUs (S_PDUs) shall be conveyed over the air by the subnetwork protocols specified in E.4.1.1.1.

E.4.1.1.3 Indirect routing support.

The optional indirect routing capability described in Appendix D can improve connectivity within a network by routing traffic through relay stations. However, the overhead traffic required to manage and support indirect routing can be substantial. Use of the Appendix D protocols should be restricted to those cases when indirect routing is required for acceptable network performance.

E.4.1.2 Support for HF-native applications.

In many cases, an application requires communication solely between host computers that communicate using only local connections and HF links. In such cases, an application protocol that is optimized for the characteristics of HF networks may be used to improve performance over protocols not designed for HF networks. The protocol stack in Figure E-1 illustrates the relationship of such HF-native application-layer protocols to the protocols defined elsewhere in this standard.

NOTE: Encryption of traffic for Communications Security (COMSEC) is not shown in Figures E-1 through E-3. Security is discussed in E.4.1.4.

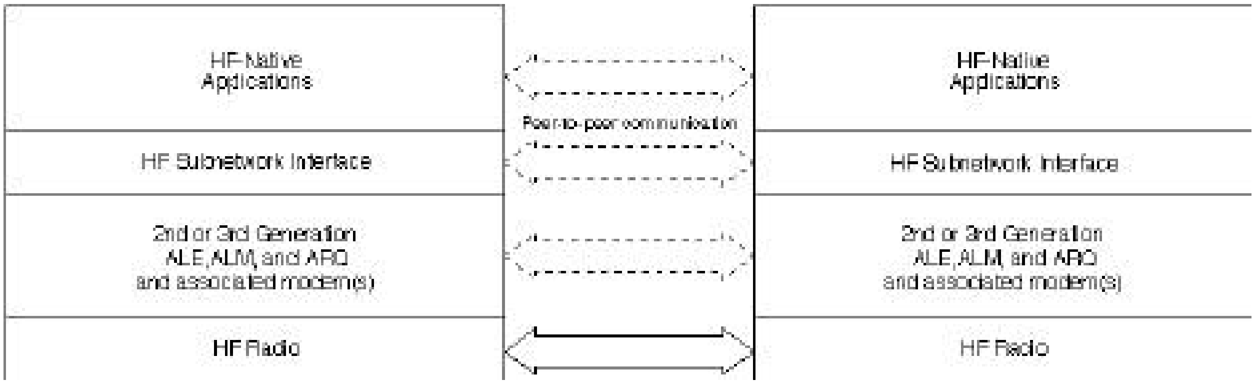

HF-Native Applications
Peer-to-peer communication
HF-Native Applications
HF Subnetwork Interface
HF Subnetwork Interface
2nd or 3rd Generation ALE,ALM, and ARQ and associated modem(s)
2nd or 3rd Generation ALE,ALM, and ARQ and associated modem(s)
HF Radio
HF Radio

FIGURE E-1. <u>HF-native application interoperation</u>.

E.4.1.3 <u>Support for Internet applications</u>.

For access via HF radio to distant local area networks (LANs) or wide-area networks (WANs) such as the Internet, the application protocols already in use within those networks must either be used within the HF network as well, or terminated at HF gateways which employ alternate HF-oriented protocols over the HF medium.

E.4.1.3.1 <u>Gateway support for Internet applications</u>.

An application-layer gateway at the boundary between an HF network and non-HF networks allows the use of dissimilar protocols at every layer within each subnetwork (Figure E-2).

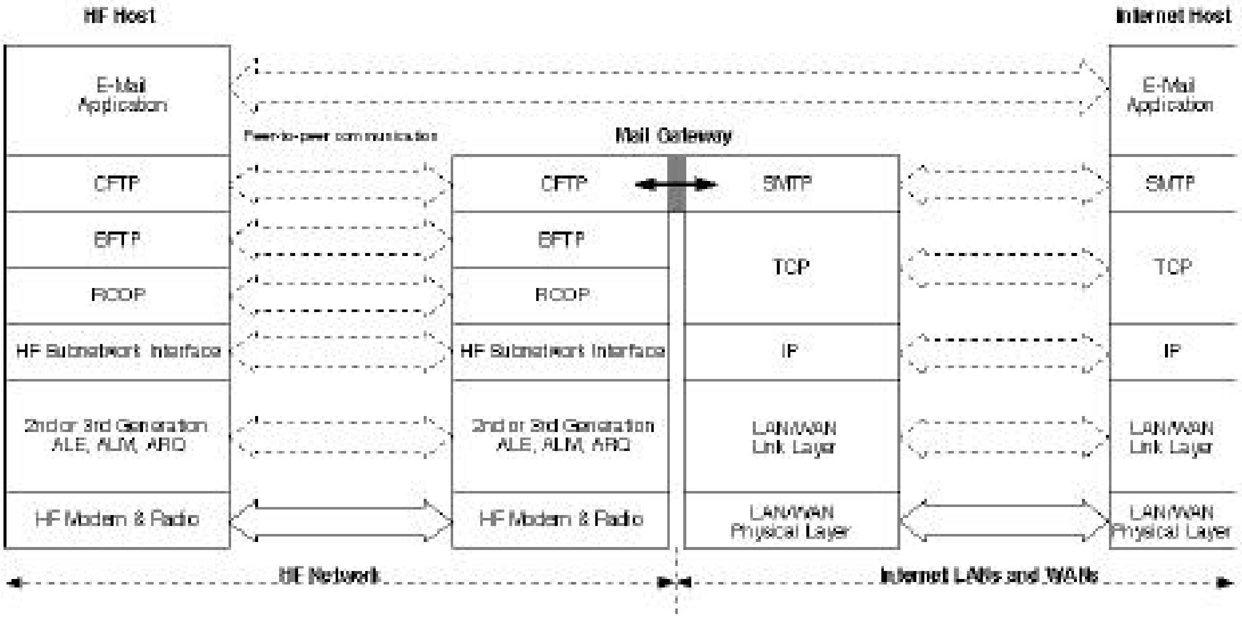


FIGURE E-2. <u>Application-layer mail gateway</u>.

This permits the use of protocols optimized for operation over HF links (e.g., the HF mail transfer protocol HMTP), and the exclusion from the HF network of protocols that can work poorly under some propagation conditions (e.g., the transmission control protocol TCP).

Gateways that operate in store-and-forward fashion introduce delays that may be undesirable in interactive applications. However, e-mail transfer is designed to operate in store-and-forward fashion, and naturally accommodates the gateway approach.

E.4.1.3.2 Transparent support for Internet applications.
For interactive applications such as remote login, file transfer, and web browsing, a router (IP gateway) at the boundary of the HF network serves to interface the distinct media-dependent protocols and hardware while allowing application and transport protocols to flow transparently through the HF subnetwork (Figure E-3). The Router function shown here may be more easily implemented in an automated HF radio or its external controller than in a commercial router because it includes components not usually found in commercial routers:

- driver software that executes HF-specific protocols
- hardware interfaces for the HF radio and modem.

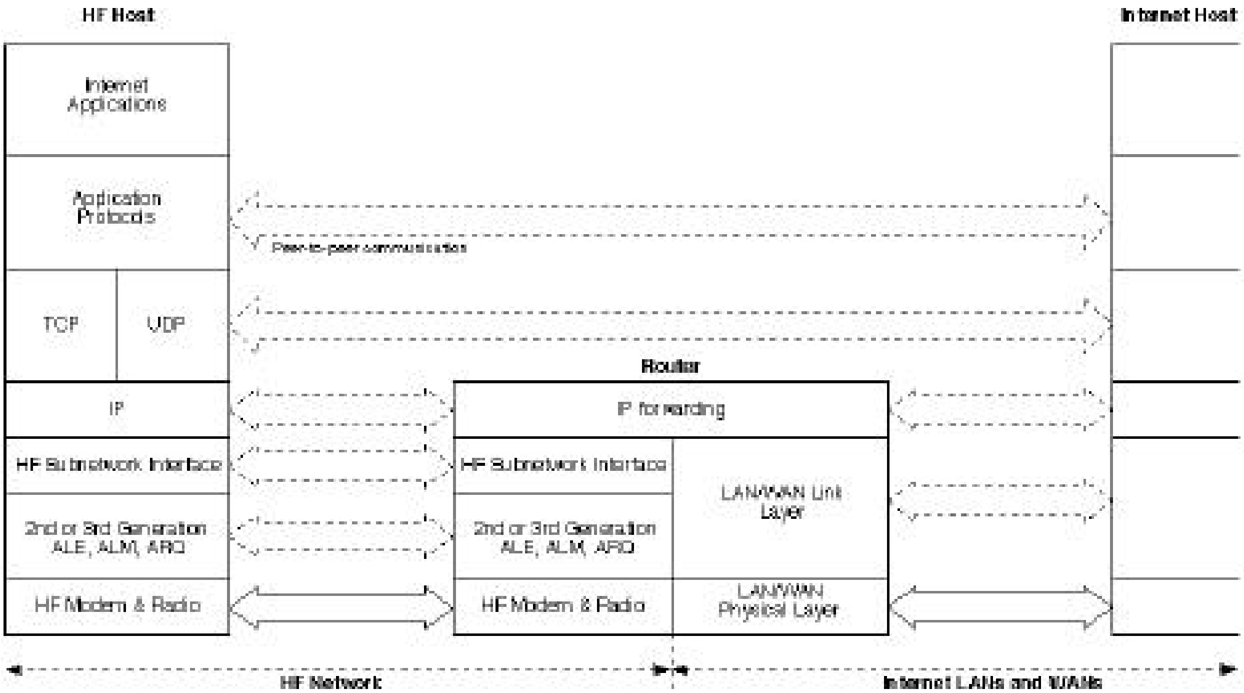


FIGURE E-3. Transparent support of Internet-native applications.

When a host computer is connected to the Internet via an HF network (e.g., HF Host in figure E-3), most Internet applications will call upon the Transmission Control Protocol (TCP) or the User Datagram Protocol (UDP) for end-to-end transport service to the distant Internet Host. These two protocols, in turn, require the services of the Internet Protocol (IP) for routing packets through the Internet. HF network designers should be aware of several potential performance problems that arise when TCP and IP are used in an HF network:

a. The two protocols together add 40 bytes of overhead to each application PDU sent.

b. TCP connection setup requires an additional three-way handshake after the link establishment handshake and data link protocol startup. Each link turnaround consumes at least three interleaver times. For example, when using a MIL-STD-188-110 serial-tone modem with a 4.8 s interleaver, this three-way handshake will add at least 43 s to the time to establish a link (at least 58 s if the data rate is 75 bps).

c. The TCP congestion avoidance mechanisms can significantly reduce throughput each time the HF data link throughput changes abruptly.

E.4.1.4 Security (information only).

Many military applications require encryption of application data. Figures E-4a through d show four alternatives for implementing Communications Security (COMSEC) for applications (including e-mail) that communicate over HF networks:

a. Application-layer security encrypts application data within each application before it is delivered to the application-layer protocol for delivery (see Figure E-4a). This offers end-to-end protection in application-specific fashion, but requires secured applications. Compression of application data will be useful only if applied before encryption.

b. Transport-layer security (e.g., the Secure Socket Layer, SSL) provides end-to-end application-independent security (see Figure E-4b). The TLS protocol [RFC-2246] was derived from SSL, and also offers optional in-line compression.

c. IPsec [RFC-2401] provides secure "tunnels" through IP networks for any higher-layer protocol, including TCP, UDP, ICMP, BGP, etc. (see Figure E-4c).

d. Link encryption (see Figure E-4d) individually secures each link in the end-to-end path. For illustration, Figure E-4d depicts a local area network (LAN) that operates in a secure area and does not require COMSEC, and an encrypted HF link.

Note that COMSEC key management is evolving towards use of the Public Key Infrastructure (PKI). Further COMSEC considerations are beyond the scope of this appendix.

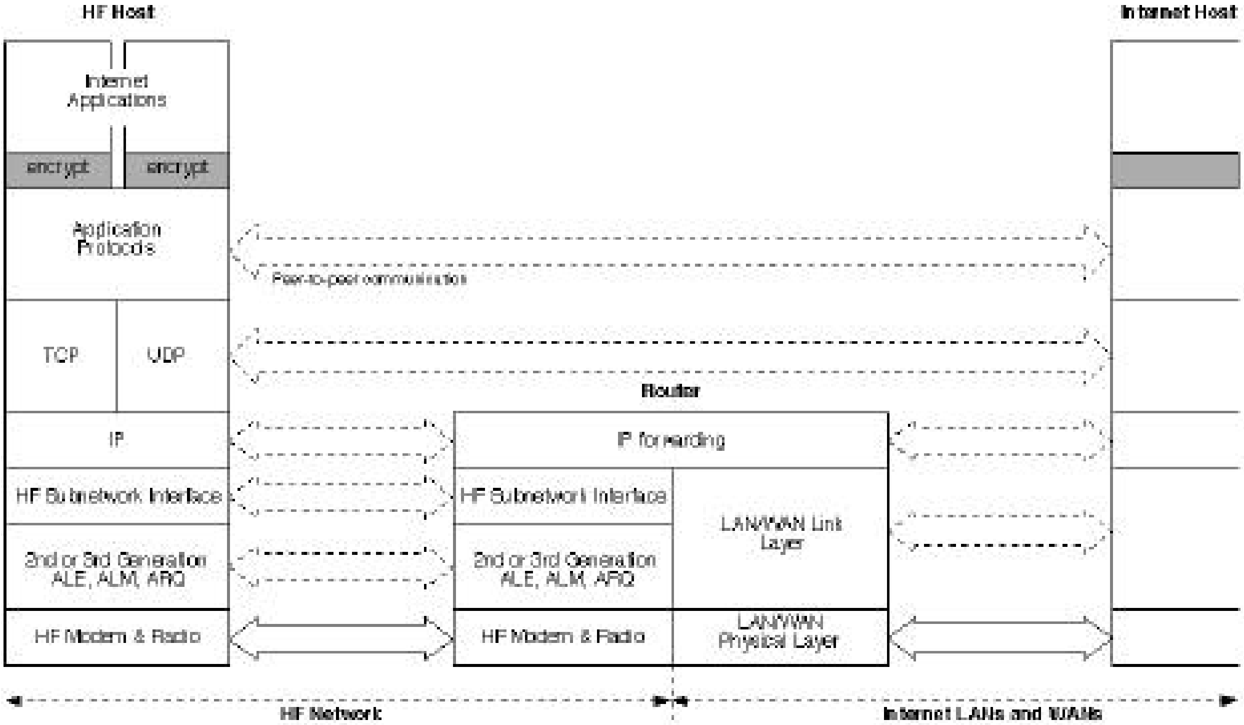


FIGURE E-4a. Application-layer security.

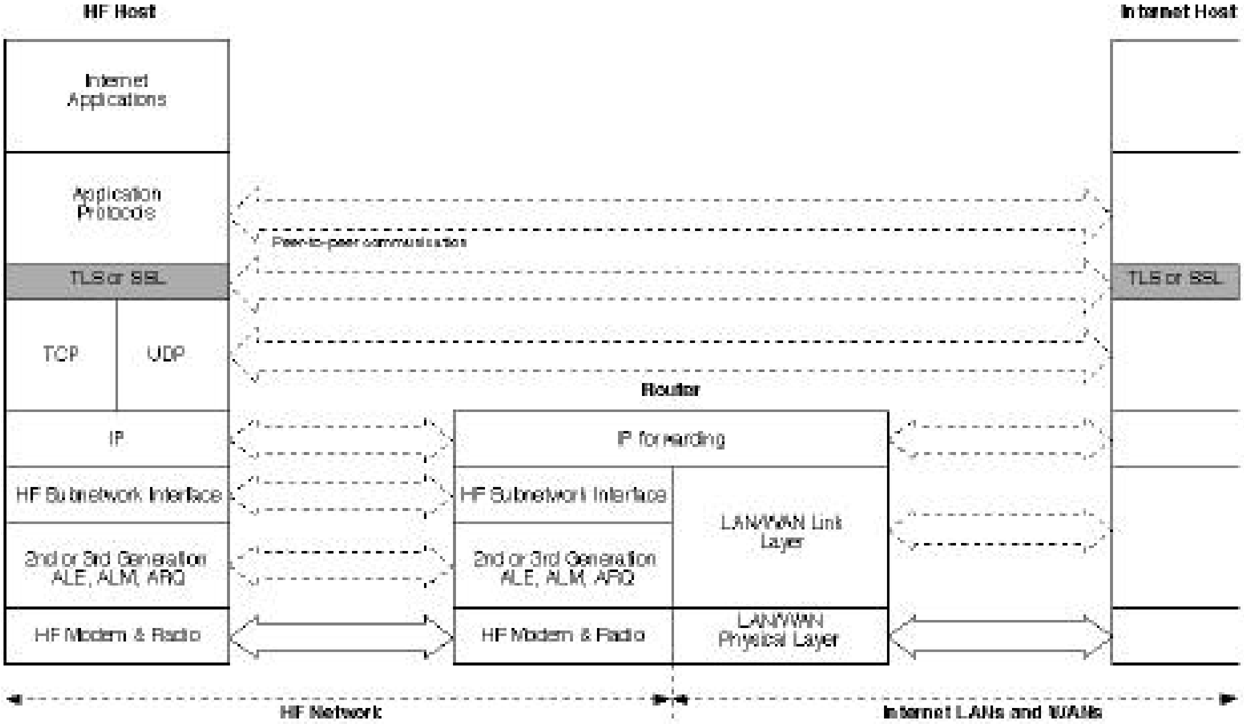


FIGURE E-4b. Transport-layer security.

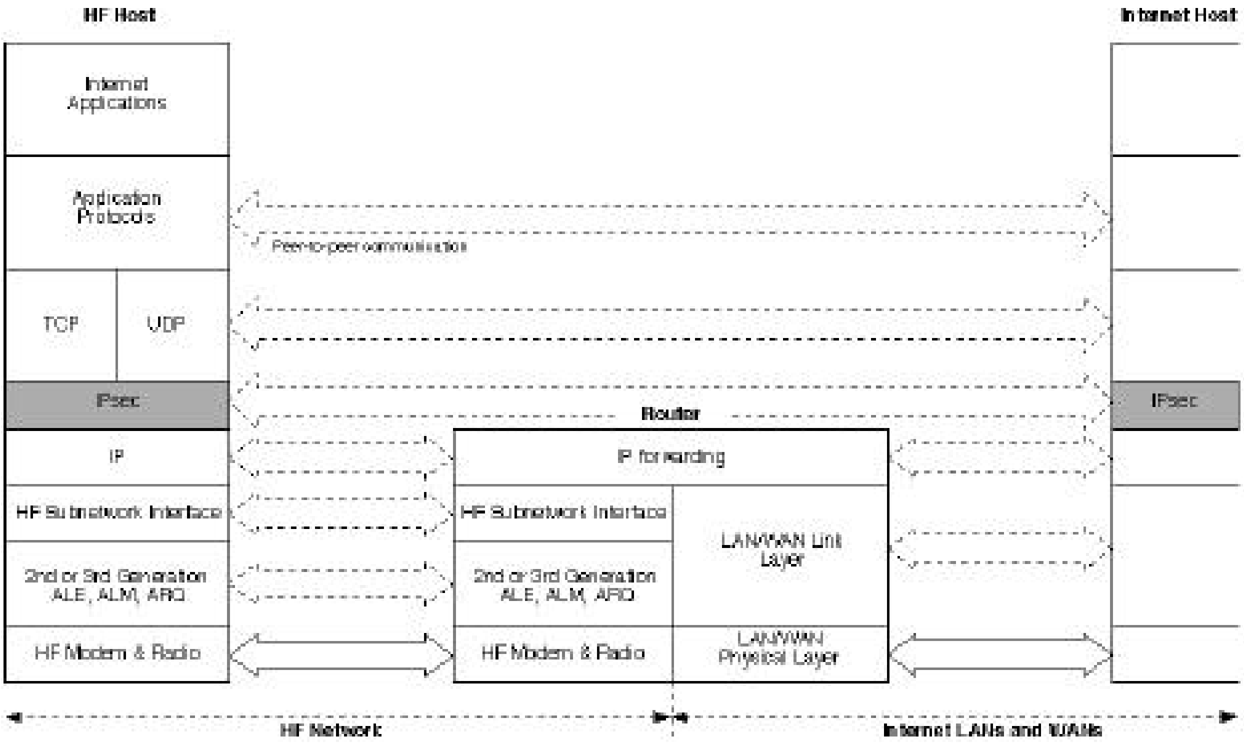


FIGURE E-4c. IPsec security.

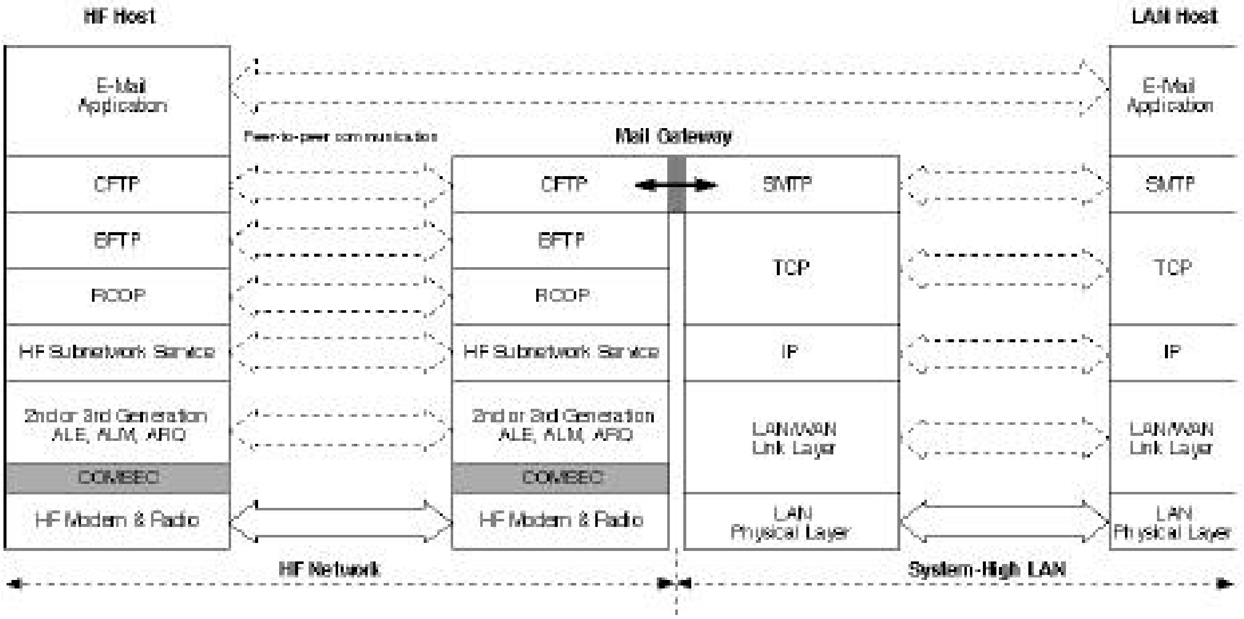


FIGURE E-4d. Link encryption.

E.4.2 Electronic mail transfer.
An HF e-mail system will be found to comply with this appendix if it conveys e-mail through HF networks using the required HF subnetwork protocols (see E.4.1.1 Required HF subnetwork service) and the Compressed File Transfer Protocol (CFTP) described in E.5.2.1 (Compressed FileTransfer Protocol). E-mail transfer on the non-HF side of mail gateways may use any suitable protocol.

E.4.2.1 Mail transfer within HF networks.
Mail shall be transferred within HF networks using CFTP, except as provided in the following two paragraphs.

E.4.2.2 Mail retrieval by call-in users.
When connectivity to a user is too infrequent to use CFTP to push messages to that user's host computer, a mail drop should be created at a host that can usually be reached by that user over a single HF link. One of the mail retrieval protocols from E.5.2.2 (HF mail retrieval protocols) shall be used to pull mail from the mail drop host to the user's host.

E.4.2.3 Mail transfer to and from NATO HF networks.
When interoperation with NATO networks employing the HF Mail Transfer Protocol (HMTP) is required, HMTP in accordance with E.5.2.3 shall be employed.

E.4.3 Digital imagery transfer. (not yet standardized)

E.4.4 Digital voice operation. (not yet standardized)

E.4.5 Other applications.
Interactive applications such as file transfer and hypertext transfer (in support of the worldwide web) shall employ the usual Internet application protocols for those applications:

Application	Protocol	Reference
Remote terminal	telnet	RFC-854
File transfer	File Transfer Protocol (FTP)	RFC-959
Hypertext transfer	Hypertext Transfer Protocol (HTTP)	RFC-2068

TCP shall be implemented at the client and server hosts that support these applications. IP and related protocols shall be implemented at client and server hosts and at routers that interconnect HF subnetworks with other subnetworks (see Figure E-3). IP shall bind as a client to the HF Subnetwork Interface at SAP ID 9.

E.5 DETAILED REQUIREMENTS.

E.5.1 Introduction.

The functions supported by the protocols specified in this section are optional. However, when the functionality provided by one of these protocols is required, that protocol shall be implemented as specified herein to provide such functionality.

E.5.2 Electronic mail protocols.

All implementations of HF e-mail shall implement client and server CFTP; this is the interoperability mode for HF e-mail. CFTP shall be used to “push” e-mail messages from one mail transfer agent (MTA) to the next. The Post Office Protocol version 3 (POP3) or the Internet Mail Access Protocol version 4 (IMAP4) shall be used when retrieving (“pulling”) e-mail messages from servers. For e-mail delivery within HF networks, the HF mail transfer protocol (HMTP) may be used as an alternative to CFTP. An e-mail server that implements more than the default protocol shall simultaneously listen for, and correctly process, incoming e-mail requests in any of its supported protocols.

E.5.2.1 Compressed file transfer protocol.

The Compressed File Transfer Protocol (CFTP) sends compressed e-mail over an HF link using a file transfer protocol, rather than a mail transfer protocol, as depicted earlier in Figure E-2. Messages produced by an email application are processed by a MTA, compressed in CFTP, segmented in the STANAG 5066 Basic File Transfer Protocol (BFTP), and passed to the subnet interface by the STANAG 5066 Reliable Connection Oriented Protocol (RCOP). At the receiving node, this process is reversed, and the uncompressed e-mail message is delivered to the receiving MTA for delivery or forwarding.

E.5.2.1.1 CFTP subnetwork service requirements

CFTP shall use the BFTP described in STANAG 5066 Annex F, paragraph F.7.3 “Example Application and Extended Client Definition: A Basic File Transfer Protocol built on top of RCOP.” BFTP uses the RCOP defined in STANAG 5066 Annex F, paragraph F.7 “Reliable Connection-Oriented Protocol.” The RCOP client that supports CFTP shall bind to the HF Subnetwork at SAP ID 12. (Note that either the second- or third-generation ARQ protocol may be used “beneath” the HF subnetwork interface to support CFTP/BFTP/RCOP and other clients.)

E.5.2.1.2 CFTP operation

Each e-mail message to be delivered using CFTP shall be formatted in accordance with RFC-822. The sending CFTP entity (CFTP “client”) shall process each such message as follows:

a. A CFTP file shall be created in accordance with E.5.2.1.3.

b. The CFTP file (including header) shall be compressed in accordance with RFCs 1950, 1951 and 1952 using an application such as gzip.

c. A BFTP header shall be prepended to the compressed CFTP message to form a BFTP_PDU.

d. The BFTP_PDU shall be segmented if necessary (see STANAG 5066 Appendix F.7.3).

e. An RCOP header shall be prepended to each BFTP segment.

f. Each RCOP packet shall be encapsulated in an S_UNIDATA_REQUEST sent to the subnetwork interface.

At the receiving host, the CFTP “server” shall process arriving packets as follows:

a. reassemble the BFTP_PDU from one or more error-free RCOP packets

b. extract the compressed CFTP file from the BFTP data field

c. decompress the CFTP file using a method compliant with RFC 1952

d. reconstruct the CFTP message

e. extract the Message field and deliver it to an SMTP server using the SMTP [RFC-821] protocol, using information extracted from the CFTP header.

E.5.2.1.3 CFTP compressed file format

A CFTP file shall be formatted as a header containing three fields, followed by the e-mail message to be sent which is the fourth field in the file. Fields shall separated by the linefeed <LF> character 0x0A. The fields and the order in which they are compressed are described in the following table.

Order of Compression	Field Name	Description
1	MessageID	The MessageID field contains a free-text string that serves as the ID for the message. The MessageID is unique to an e-mail message. It is not the same as the ID in the email message that follows "Message-ID:" in the RFC 822 header. When the compressed file is decompressed, the MessageID is used as the root filename for the decompressed components. The MessageID must be less than 256 characters and is composed of upper/lowercase alphanumeric characters.
2	RecipientList	The RecipientList is a string containing e-mail addresses extracted from the e-mail message separated by the "," character (0x2c). The first address in the recipients list is the "Return-Path". There can be cases where there is no return path, e.g. the mail is being bounced by a Mail Transfer Agent. In these cases, the first address will be a blank string (the RecipientList will begin with at least one space (0x20) character). The RecipientList must be less than 10240 characters.
3	MessageSize	The MessageSize is encoded as a decimal number in string format. It represents the size (in bytes) of the Message field that follows the MessageSize field.
4	Message	Actual message including RFC 822 header, as produced by SMTP processing in accordance with RFC 821 (including the transparency procedure specified in RFC 821 paragraph 4.5.2).

All characters in the CFTP file are 8 bits.

E.5.2.2 HF mail retrieval protocols.
When a user is usually not reachable (i.e., the user connects sporadically to pick up e-mail), CFTP will not be appropriate for delivery of mail *to* that user. In such cases, POP3 in accordance with RFC 1939 or IMAP4 in accordance with RFC 1730 shall be used to retrieve mail from a mail drop server (see E.4.2.2). Messages *sent* by such users shall be conveyed to the server in accordance with E.4.2.1.

E.5.2.3 HF mail transfer protocol.
The HF Mail Transfer Protocol (HMTP) is an extended version of the Simple Mail Transfer Protocol (SMTP). HMTP clients and servers shall implement SMTP in accordance with RFC 821, the SMTP service extension (“EHLO”) protocol in accordance with RFC 1651, command pipelining in accordance with RFC 2197 as modified in the following paragraphs, and MIME Extensions for SMTP, as defined in RFCs 2045, 2046, 2047, and 2049.

E.5.2.3.1 HMTP server requirements.
An HMTP server shall comply with RFC 2197 to ensure a reliable implementation of the basic SMTP protocol.

a. The server shall not lose buffered incoming commands in its transport layer (or equivalent) queue. This rule ensures that servers will correctly process arbitrarily long batches of commands.

b. The server shall send all buffered responses whenever its queue of incoming commands is emptied. This rule ensures that responses to a batch of commands will always be sent after the end of the batch, no matter how short the batch, providing backward interoperability with SMTP clients.

For improved performance over HF subnetworks, HMTP servers may depart from the rules in RFC 2197 in the following regard.

c. The server may buffer responses to all incoming commands until the queue of incoming commands is empty.

The HMTP server shall provide a relay capability for the client in accordance with RFC 821. This relay capability may be used to provide an application-level gateway capability between the HF subnetwork and other networks, for example, those with lower latency and higher throughput for which the other protocols are more suited. With respect to relay and routing, the argument to the SMTP MAIL command is in the form “user@hostname”, which specifies who the mail is from. The argument to the RCPT command is in the same form and specifies the ultimate destination of the mail. The HMTP server shall forward mail in accordance with RFC 821. A destination may be rejected only if the server can not understand it. Source routing shall not be used, as the HMTP model requires the server to have mail-routing information.

E.5.2.3.2 HMTP client requirements.
When connected to a server that supports command pipelining, HMTP clients shall group commands to the maximum extent permitted in RFC 2197:

a. All setup commands, including RSET (if required), MAIL, RCPT, and DATA, for each message shall be sent as a single group.

b. Multiple messages sent to a single server shall be chained by appending the setup commands for each subsequent message to the message body of the preceding message.

For improved performance over HF subnetworks, HMTP clients connected to an HMTP server may depart from the rules in RFC 2197 in the following regard.

c. Any number of the commands sent to a single server, including the initial EHLO, may be grouped for transmission.

Unlike ESMTP with command pipelining, which first checks for a valid response that confirms a peers capability to use the pipelined commands, HMTP proceeds under the assumption that the peer process is fully compliant with its pipelined SMTP commands. This streamlines the process by using the minimum number of transactions between the client and server. The disadvantage is that if the peer-level mail process is not compliant with HMTP, then the transactions are lengthy to no purpose, since the mail will not be transferred correctly but the transmissions could take significant time on the channel before this is determined.

When connected to a server that does not support command pipelining, HMTP clients shall execute SMTP in its basic interlocked mode in accordance with RFC 821.

E.5.2.3.3 HMTP over TCP.
When HMTP uses TCP transport services, it shall listen on TCP port 25 (the well-known SMTP port), and, in general, use TCP in the same manner as does SMTP.

E.5.2.3.4 HMTP without TCP.
When TCP is not used to transport HMTP data, the HMTP server shall bind to SAP ID 3 of the HF subnetwork service.

E.6 NOTES.

(This section contains information of a general or explanatory nature which may be helpful, but is not mandatory.)

E.6.1 Changes from previous issue.

The margins of this standard are marked with vertical lines to indicate where changes from the previous issue were made. This was done as a convenience only and the Government assumes no liability whatsoever for any inaccuracies in these notations. Bidders and contractors are cautioned to evaluate the requirements of this document based on the entire content irrespective of the marginal notations and relationship to the last previous issue.

F.4.3.3 Occupancy detection: link while hopping.
3G ALE systems operating in link-while-hopping mode (Mode 2) shall correctly recognize that a traffic hop set is occupied at least as reliably as indicated in table C-II during the Listen portion of Slot 0 (see F.5.2.2, Hopping synchronous dwell structure). The test procedure of A.4.2.2 shall be used. Systems shall also meet or exceed the requirements of Table F-III for detecting calling hop sets in use while listening before calling during Slots 1 through 4. The probability of declaring a hop set occupied when each hop dwell contains only additive white Gaussian noise (AWGN) shall be less than 1percent.

TABLE F-III. Occupancy detection requirements for linking while hopping (3 kHz SNR dB).

Waveform	AWGN 3 kHz SNR (dB)	Minimum Required Detection Probability
2G-ALE	1	50%
3G-ALE (BW0)	-8	50%
3G-HDL (BW2)	1	30%
SSB Voice	7	50%
MIL-STD-188-110 or FED-STD-1052 PSK modem	1 7	30% 70%
STANAG 4285 PSK modem	1	30%

F.5 DETAILED REQUIREMENTS

F.5.1 Linking before hopping.
When 3G ALE systems link before hopping, the ALE protocols of Appendix C shall be employed with the following modification: hopping synchronization and startup shall occur at the point in the respective protocol at which traffic startup would occur in non-hopping operation. Traffic startup shall commence upon completion of hopping startup.

F.5.2 Linking while hopping.
When linking while hopping, a 3G ALE system shall operate in the modified synchronous mode specified below. The following timing parameters are used:

- The hopping dwell period, denoted T_{hop}, is the reciprocal of the hopping rate.
- Each dwell comprises a guard time T_{guard}, during which the radio is changing frequency, and a user data time T_{data}, during which user data may be sent. $T_{hop} = T_{guard} + T_{data}$.
- Stations are synchronized while hopping. The maximum discrepancy among network member time bases is T_{disc}.
- The maximum propagation delay among network member stations is T_{prop}.
- The duration of the transmit level control, preamble, and data portions of the 3G-ALE PDU are T_{tlc}, $T_{BW0\ pre}$, and $T_{BW0\ data}$, respectively

F.5.2.1 Hopping synchronous slot duration.
When the network hopping rate is such that an entire 3G-ALE PDU can be sent during T_{data}, the slot time T_{slot} shall be equal to T_{hop}. Otherwise, the 3G-ALE T_{slot} shall be extended to the smallest integral multiple of T_{hop} greater than or equal to T_{prop} plus the time to send an entire 3G-ALE PDU using the procedure specified in F.5.2.3 Hopping PDU transmission.

F.5.2.2 Hopping synchronous dwell structure.
The dwell structure for 3G ALE while hopping shall comprise six slots, each of duration T_{slot}:

- Slot 0, during which stations shall synchronously check a traffic hop set for occupancy;
- Slots 1-5, during which stations initiate calls and other protocols in accordance with Appendix C.

F.5.2.3 Hopping PDU transmission.
When any 3G PDU cannot be sent during a single hop, it shall be spread over multiple hops as described below.

F.5.2.3.1 Hopping PDU transmit level control sequence.
The transmit level control portion of a PDU shall be sent by the controller to the transmitter during T_{guard} and the first portion of T_{data}, with symbols dropped from the beginning of the sequence if necessary so that the end of the T_{tlc} sequence shall occur at time T_{disc} after the end of the guard time (see figure F-1). Some of the symbols may be dropped by the transmitter while it is changing frequency.

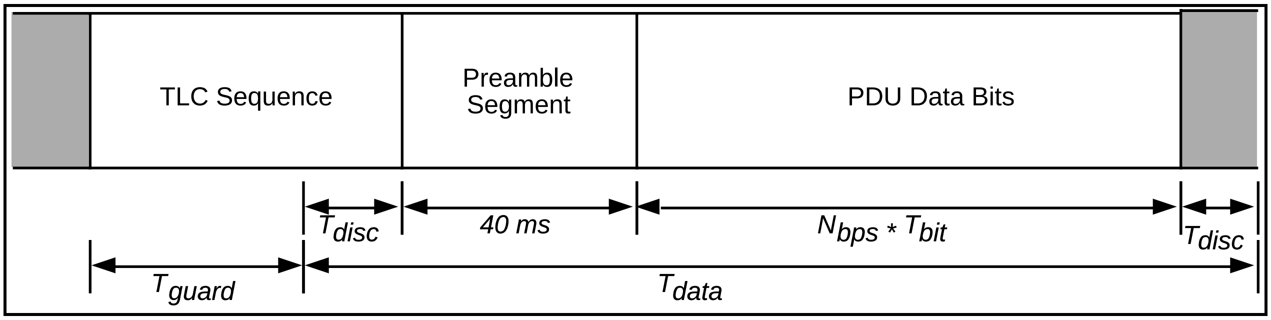


FIGURE F-1. Hopping PDU transmission.

F.5.2.3.2 Hopping 3G-ALE PDU transmission during data time.
When $T_{data} \geq T_{BW0\ pre} + T_{BW0\ data} + 2\ T_{disc}$, the 3G-ALE PDU shall be sent during a single hop, with the preamble beginning immediately after the end of the T_{tlc} sequence.

Otherwise, the preamble and data portions of the 3G ALE PDU shall be split over multiple hops. The transmissions during each T_{hop} shall be (a portion of) the T_{tlc} sequence as described above, followed immediately by a 40 ms (96 symbol) segment of the preamble, followed immediately by up to N_{bph} PDU data bits:

$$N_{bph} = \left\lfloor \frac{T_{data} - 40ms - 2T_{disc}}{T_{bit}} \right\rfloor$$

Transmissions should cease during the period from the final symbol of the PDU data bits until the first symbol of the next T_{tlc} sequence. This will reduce interference levels. The number of hops required to send the 26 data bits that compose a 3G ALE PDU is H_{BW0}:

$$H_{BW0} = \left\lceil \frac{26}{N_{bph}} \right\rceil$$

Preamble segments shall be sent in order in successive hops, with the specified symbol sequence cycling back to the beginning if necessary in the later hops. If the unused PDU data bit positions in the final hop do not exceed T_{prop}, an extra hop must be included in T_{slot} to allow for propagation to all network stations.

F.5.2.3.3 Hopping transmission of other PDUs during data time.
The other 3G PDUs shall be sent similarly to the 3G ALE PDUs.

F.6 NOTES

None.

APPENDIX G

SECOND GENERATION DATA LINK PROTOCOL

G.1 GENERAL

G.1.1 Scope.

The purpose of this appendix is to specify the requirements for an additional data link protocol to be used with implementations of second generation (2G) HF link automation.

G.1.2 Applicability.

2G ALE inherently includes data transfer protocols that use the frequency-shift keyed modem specified in Appendix A: Automatic Message Display (AMD), Data Text Message (DTM), and Data Block Message (DBM) protocols. The data link protocol specified in this appendix shall be used in 2G HF networks that employ data modems complying with MIL-STD-188-110.

G.2 APPLICABLE DOCUMENTS

G.2.1 General.

This section does not include documents cited in other sections of this standard or recommended for additional information or as examples.

G.2.2 Government documents.

The following standard forms a part of this document to the extent specified herein. Unless otherwise specified, the issues of these documents are those listed in the issue of the Department of Defense Index of Specifications and Standards (DODISS) and supplement thereto cited in the solicitation.

STANDARDS

DEPARTMENT OF DEFENSE

MIL-STD-188-110	Interoperability and Performance Standards for Data Modems

(Unless otherwise indicated, copies of the above standard are available from the Standardization Document Order Desk, 700 Robbins Ave. Building 4D, Philadelphia, PA 19111-5094.)

INTERNATIONAL STANDARDIZATION DOCUMENTS

NORTH ATLANTIC TREATY ORGANIZATION (NATO)
STANDARDIZATION AGREEMENTS (STANAG)

STANAG 5066	Profile for High Frequency (HF) Radio Data Communications

G.3 DEFINITIONS

None.

G.4 GENERAL REQUIREMENTS

The HF data modem waveforms specified in MIL-STD-188-110 can achieve better performance than the 2G ALE waveform in both throughput and low-SNR operation. Systems complying with this Appendix shall implement each of the following Annexes of STANAG 5066:

a. Annex A: Subnetwork Interface Sublayer

b. Annex B: Channel Access Sublayer

c. Annex C: Data Transfer Sublayer

The C_PDUs of the STANAG 5066 Data Transfer Sublayer shall be sent over the air using one or more of the following waveforms from MIL-STD-188-110:

Mode	Data Rates	Section of MIL-STD-188-110
SSB	75 – 2400	5.3.2: Serial (single-tone) mode
SSB	3200 – 9600	Appendix C: HF data modem waveforms for data rates above 2400 bps
ISB or multiple radios	Any	Appendix F: HF data modems for multiple channel systems

G.5 DETAILED REQUIREMENTS

None.

G.6 NOTES

None

H.1 GENERAL.

H.1.1 Scope.

The scope of this appendix is limited to the Management Information Base (MIB) for automatic high frequency (HF) radio networks.

H.1.1.2 Applicability.

This appendix defines the MIB for automated HF radio networks. Management of HF radio networks through the use of the Simple Network Management Protocol (SNMP) requires the formal definition of the data objects to be remotely read and written by a network management program.

H.2. APPLICABLE DOCUMENTS.

H.2.1 General.

The documents listed in this section are specified in H.4 and H.5 of this standard. This section does not include documents cited in other sections of this standard or recommended for additional information or as examples. While every effort has been made to ensure the completeness of this list, document users are cautioned that they must meet all specified requirements documents cited in H.4, and H.5 of this standard, whether or not they are listed.

H.2.2 Government documents.

INTERNATIONAL STANDARDIZATION DOCUMENTS

NORTH ATLANTIC TREATY ORGANIZATION (NATO)
STANDARDIZATION AGREEMENTS (STANAG)

STANAG 4538	Technical Standards for an Autommatic Radio Control System for HF Communication Links

H.2.3 Non-Government publications.

The following documents form a part of this document to the extent specified herein. Unless otherwise specified, the issues of the documents which are Department of Defense (DoD) adopted are those listed in the issues of the Department of Defense Index of Specifications and Standards (DODISS) cited in the solicitation. Unless otherwise specified, the issues of the documents not listed in the DODISS are the issues of the documents cited in the solicitation (see 6.3).

INTERNET DOCUMENTS

RFC-1155	Structure and Identification of Management Information
RFC-1157	A Simple Network Management Protocol (SNMP)
RFC-1213	Management Information Base for Network Management of TCP/IP-based Internets: MIB-II
RFC-1441	Introduction to Version 2 of the Internet-standard Network Management Framework
RFC-1442	Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)
RFC-1443	Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2)
RFC-1444	Conformance Statements for Version 2 of the Simple Network Management Protocol (SNMPv2)
RFC-1445	Administrative Model for Version 2 of the Simple Network Management Protocol (SNMPv2)
RFC-1446	Security Protocols for Version 2 of the Simple Network Management Protocol (SNMPv2)
RFC-1447	Party MIB for Version 2 of the Simple Network Management Protocol (SNMPv2)
RFC-1448	Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)
RFC-1449	Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)
RFC-1450	Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)
RFC-1451	Manager-to-Manager Management Information Base
RFC-1452	Coexistence between Version 1 and Version 2 of the Internet-standard Network Management Framework

(Internet documents may be obtained from http://www.rfc-editor.org/rfc.html.)

H.3 DEFINITIONS.

H.3.1 Standard definitions.

None.

H.3.2 Abbreviations and acronyms.

The abbreviations and acronyms used in this document are defined below. Those listed in the current edition of FED-STD-1037 have been included for the convenience of the reader.

ALE	automatic link establishment
AME	automatic message exchange
ARPANET	Advanced Research Projects Agency Network
ASN.1	abstract syntax notation 1
bps	bits per second
DoD	Department of Defense
HF	high frequency
HFDLP	HF Data Link Protocol
HNMP	HF Network Management Protocol
LP	linking protection
MIB	Management Information Base
PIN	personal identification number
PDU	protocol data unit
SNMP	Simple Network Management Protocol
SNMPv2	Version 2 of the Simple Network Management Protocol
TCP/IP	transmission control protocol/internet protocol
UDP	user datagram protocol

H.4 GENERAL REQUIREMENTS.

H.4.1 Introduction.

Automation of HF radio networks to date has simplified the tasks related to establishing links using HF radios. However, the automatic link establishment (ALE) technology that hides the complexities of linking has generated a new problem in radio network management: the automatic controllers use a number of intricate data structures that must be kept consistent throughout a network if operations are to proceed smoothly.

H.4.1.1 Network Connectivity and equipment status.

An aspect of network management that has not been addressed by previous HF standards is the need to observe network connectivity and equipment status from network control sites so that corrective action can be initiated promptly when malfunctions or other disruptions occur. Managers of packet networks have been at work on network management problems for some time, so it makes sense to adopt the procedures used in these more mature automated networks when that technology can be usefully applied to the management of HF networks.

H.4.1.2 Internet suite of protocols.

Perhaps the best-known of the packet network technologies is the Internet suite of protocols (including the transmission control protocol (TCP) / internet protocol (IP)), which grew out of the DoD-sponsored Advanced Research Projects Agency Network (ARPANET) research. The network management approach used in the Internet and associated sub-networks is based upon the Internet-standard Network Management Framework developed in the late 1980s. This technology is more often referred to by the protocol that it employs for managing network nodes, the SNMP [RFC-1157].

H.4.2 SNMP.

SNMP was designed so that it explicitly minimizes the number and complexity of management functions realized by the management agent itself. That is, the development costs of including SNMP in managed equipment are minimized at the expense of (perhaps) increasing the complexity of the software for network management stations. However, the ratio of managed nodes to management stations is large so the benefit of widespread implementation has greatly outweighed the cost of implementing the management software.

To briefly summarize the salient points of the SNMP approach:

a. Network management stations monitor and control network elements by communicating with agents in those elements.

b. This interaction uses SNMP [RFC-1157] to get and set the values of defined data objects. Agents may also send trap messages to management stations to announce important events asynchronously.

c. The defined data objects are described in the MIB [RFC-1213], which is currently strongly oriented toward the TCP/IP protocol suite, but is easily extensible. Object definitions are expressed formally in abstract syntax notation 1 (ASN.1) [ISO 8824].

d. Object names and values are encoded in accordance with a set of ASN.1 Basic Encoding Rules [ISO 8825].

e. When elements do not implement SNMP, they may still be managed by using proxy agents that translate the standard SNMP messages into proprietary messages understood by the non-SNMP elements.

f. Authentication is included in the standard, although current practice uses only trivial authentication. The mechanism is extensible using ideas similar to HF linking protection.

g. SNMP requires only a connectionless datagram transport service (e.g., the user datagram protocol (UDP) in the Internet).

SNMPv2, in accordance with RFC 1441-1452 shall be employed for HF network management with the following additional requirements:

(1) An agent receiving a SetRequest that selects a non-existent row in a table shall automatically create the requested row subject to resource availability, setting column objects in the new row to their default values unless other valid values are specified in the SetRequest message. However, if any value in the SetRequest message specifies an invalid value for any column object in the new row, the new row shall not be created, and a GetResponse message shall be returned indicating the erroneous variable binding.

(2) Table rows invalidated by a SetRequest shall not be reported in responses to GetNextRequests; that is, from the point of view of management stations, invalidated rows are deleted from the table.

Using HNMP, this transfer will require 2,831 octets. Using this as a baseline, the number of octets to load the remaining equipment is estimated on the following table:

TABLE H-I. Over the air transfers.

Radios	15,000
Antennas	500
HFDLP	200
Antenna matrix	500
BLACK patch	500
Total	20,000 (est)

H.5.3 Network management MIBs and HNMP definitions.
Table H-II contains MIBs for network management of automated HF radio networks. These MIBs are merely a subset of the management information that will be needed for a full implementation of automated HF radio network management. Additional objects may be defined in equipment-specific MIBs as described in the Structure of Management Information [RFC-1442]. The MIB for the third-generation technology in Appendix C is included in the ARCS MIB in section 12.2 “ARCS management information base” of STANAG 4538 Annex C.