NATO STANDARD

AComP-4691

MULTI-HOP IP NETWORKING WITH LEGACY UHF RADIOS: MOBILE AD HOC RELAY LINE OF SIGHT NETWORKING (MARLIN)

Edition A Version 1
JUNE 2016

NORTH ATLANTIC TREATY ORGANIZATION

ALLIED COMMUNICATIONS PUBLICATION

Published by the
NATO STANDARDIZATION OFFICE (NSO)

INTENTIONALLY BLANK

NORTH ATLANTIC TREATY ORGANIZATION (NATO)

NATO STANDARDIZATION OFFICE (NSO)

NATO LETTER OF PROMULGATION

30 June 2016

1. The enclosed Allied Communications Publication AComP-4691, Edition A, Version 1, MULTI-HOP IP NETWORKING WITH LEGACY UHF RADIOS: MOBILE AD HOC RELAY LINE OF SIGHT NETWORKING (MARLIN), which has been approved by the nations in the Consultation, Command, and Control Board (C3B), is promulgated herewith. The agreement of nations to use this publication is recorded in STANAG 4691.

2. AComP-4691, Edition A, Version 1 is effective upon receipt.

3. No part of this publication may be reproduced, stored in a retrieval system, used commercially, adapted, or transmitted in any form or by any means, electronic, mechanical, photo-copying, recording or otherwise, without the prior permission of the publisher. With the exception of commercial sales, this does not apply to member or partner nations, or NATO commands and bodies.

4. This publication shall be handled in accordance with C-M(2002)60.

for

Dieter Schmaglowski
Deputy Director NSO
Branch Head P&C

Edvardas MAŽEIKIS
Major General, LTUAF
Director, NATO Standardization Office

INTENTIONALLY BLANK

RESERVED FOR NATIONAL LETTER OF PROMULGATION

INTENTIONALLY BLANK

RECORD OF RESERVATIONS

CHAPTER	RECORD OF RESERVATION BY NATIONS
Note: The reservations listed on this page include only those that were recorded at time of promulgation and may not be complete. Refer to the NATO Standardization Document Database for the complete list of existing reservations.	

INTENTIONALLY BLANK

RECORD OF SPECIFIC RESERVATIONS

[nation]	[detail of reservation]
Note: The reservations listed on this page include only those that were recorded at time of promulgation and may not be complete. Refer to the NATO Standardization Document Database for the complete list of existing reservations.	

INTENTIONALLY BLANK

TABLE OF CONTENTS

INTENTIONALLY BLANK

Multi-hop IP Networking with Legacy UHF Radios: Mobile Ad hoc Relay Line of Sight Networking (MARLIN)

Annexes:

A. MOBILE AD HOC RELAY LINE OF SIGHT NETWORKING (MARLIN) PROFILE
B. TECHNICAL SPECIFICATIONS TO ENSURE INTEROPERABILITY OF SERIAL WAVEFORMS FOR 25 KHZ BANDWIDTH LINE OF SIGHT VHF AND UHF RADIO DATA COMMUNICATIONS
C. TECHNICAL SPECIFICATIONS TO ENSURE INTEROPERABILITY OF SERIAL WAVEFORMS FOR 100 KHZ BANDWIDTH LINE OF SIGHT VHF AND UHF RADIO DATA COMMUNICATIONS
D. TECHNICAL SPECIFICATIONS TO ENSURE INTEROPERABILITY OF SERIAL WAVEFORMS FOR 300 KHZ BANDWIDTH LINE OF SIGHT VHF AND UHF RADIO DATA COMMUNICATIONS
E. TECHNICAL SPECIFICATIONS TO ENSURE INTEROPERABILITY OF SERIAL WAVEFORMS FOR 500 KHZ BANDWIDTH LINE OF SIGHT VHF AND UHF RADIO DATA COMMUNICATIONS

Related Documents:

a. AC/322(SC/6-AHWG/2)N(2010)0020 - Mobile Ad Hoc Relay Line of Sight Networking Concept of Operations
b. AC/322(SC/6-AHWG/2)N(2010)0021 - TCP Performance Enhancement over Wireless Networks: Internet Protocol Traffic Manager (IPTM) proposal

AIM

1. The aim of this agreement (MARLIN – Mobile Ad Hoc Relay Line of Sight Networking) is to describe the system concept and set of protocols that provide Internet Protocol (IP) data transfer in multiple-node, multiple-hop dynamic networks employing line of sight (LOS) radio bearers. The protocols were originally developed to provide connectivity amongst ships at sea, but applications with other platforms and in other environments are possible.

AGREEMENT

2. Participating nations agree to use MARLIN as the interoperability standard for transport of IP data over legacy bearers at UHF, VHF, and HF LOS/Extended LOS (ELOS).

GENERAL

3. The MARLIN system makes use of these legacy bearers by interfacing between an IP data router and a tactical radio. The MARLIN protocol stack provides a number of services:

- Automatic and distributed neighbor discovery and network organization;
- Distributed coordination of transmission to avoid collisions on a shared channel;
- Ad-hoc operation;
- Reliable data delivery with repeat requests and automatic retransmission;
- Dynamic bandwidth allocation in response to reported requirements;

Traffic relay to extend connectivity in multi-hop topologies.

IMPLEMENTATION OF THE AGREEMENT

4. This standard is considered implemented by a nation when that nation has placed in service equipment complying with the standards defined in this document.

ANNEX A MOBILE AD HOC RELAY LINE OF SIGHT NETWORKING (MARLIN) PROFILE

A.1. Introduction

A.1.1 Overview

The Mobile Ad Hoc Relay Line of Sight Networking (MARLIN) profile describes the system concept and set of protocols that provide Internet Protocol (IP) data transfer in multiple-node, multiple-hop dynamic networks employing line of sight (LOS) radio bearers. The protocols were originally developed to provide connectivity amongst ships at sea, but applications with other platforms and in other environments are possible.

The MARLIN protocol stack provides a number of services:

- Automatic and distributed neighbor discovery and network organization.
- Distributed coordination of transmission to avoid collisions on a shared channel,
- Ad-hoc operation.
- Reliable data delivery with repeat requests and automatic retransmission.
- Dynamic bandwidth allocation in response to reported requirements.
- Traffic relay to extend connectivity in multi-hop topologies.

The protocols are highly parameterized and so are capable of operating over a variety of bearers.

A.1.2 Concept of Operations

The primary method of information transfer at sea is through satellite communications (SATCOM). However, SATCOM is not a feasible, all-encompassing solution for operational and tactical IP data networking amongst Coalition ships and other platforms at sea. Not all platforms are fitted for SATCOM, and access is expensive and sometimes limited. In addition, there are vulnerability and survivability issues that necessitate a backup or alternative to SATCOM for communications redundancy.

Several UHF, VHF, and HF LOS/Extended LOS (ELOS) bearers are available as alternatives to SATCOM. Furthermore, nearly all platforms engaged in Coalition operations have access to legacy tactical voice radios in these bands, especially UHF and HF. These legacy bearers could be made available for the transport of IP data.

The MARLIN system makes use of these legacy bearers by interfacing between an IP data router and a tactical radio. Because the radio channel is shared amongst platforms, coordinated channel access is required. Data relay amongst platforms might also be necessary to extend the network beyond line of sight or because radio connectivity is not always all-informed. Furthermore, the system must be highly adaptive to respond to highly dynamic topologies caused by platform mobility and other factors.

One possible LOS topology is represented in Figure A-1 below. Since the networks are formed in an ad hoc manner and the system is designed to be master-less, the system must self-organize in a distributed manner. This includes the admission of new members to an existing network, and the merging of two or more partitions as they come within range.

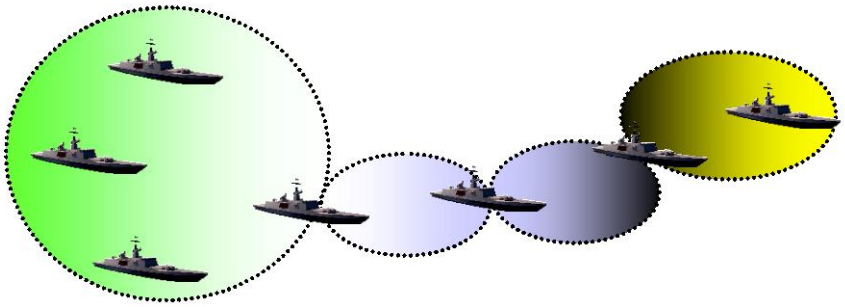


Figure A-1: Illustrative MARLIN topology

The expected operational requirements that were considered when designing MARLIN are listed in Appendix A.1. These were derived from current networking requirements; network and platform configurations; and typical battle group sizes, formations, and maneuvers.

A.1.3 Protocol Stack and Interfaces

An overview of the MARLIN protocol stack is shown in Figure A-2. MARLIN operates at Layer 2 of the Open System Interconnect (OSI) reference model, between an Internet Protocol (IP) router and a modem, cryptographic device, or radio which is operated in a half-duplex, single-channel mode. The protocol stack is virtual in the sense that any implementation of this standard must execute the functions performed by the sublayers shown. This standard does not mandate a specific framework for software implementation and internal software design within this stack is not subject to interoperability specification. To be compliant, an implementation must comply with all specifications defining message types and message handling on interfaces to bearer and routing functions.

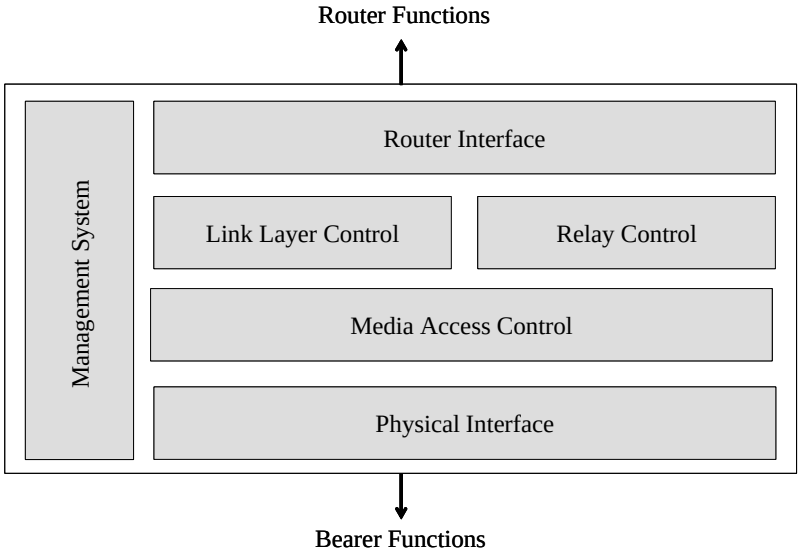


Figure A-2: MARLIN protocol stack

The following are brief descriptions of each component of the protocol stack. Each component is described more fully in subsequent sections:

- **Router Interface (RI)**: communicates with the attached IP router.
- **Link Layer Control (LLC)**: responsible for packet segmentation, reassembly, and duplicate detection. The LLC may also perform data compression and decompression using an open source algorithm.
- **Relay Control (RC)**: monitors the LOS network topology to determine network membership and shortest paths to desired destinations. This helper protocol provides Layer 2 routing when required.
- **Media Access Control (MAC)**: coordinates transmissions on the shared channel using a synchronous Time-Division Media Access (TDMA) scheme, allocating time slots in proportion to reported requirements. In addition, the MAC increases the reliability of data transfer by implementing automatic repeat request (ARQ).
- **Physical Interface (PI)**: connects to the on-air bearer.

A.1.4 Example Process Flow

Before describing the operation of the MARLIN protocol layers in detail, it is instructive to consider: the transfer of an IP data packet in a particular scenario. In the following example process flow, one node in a three-node network wishes to transfer data to another node. The nodes are located in a line with the transmitting and receiving nodes at either end of the network as illustrated in Figure A-3. Assuming that the two endpoint nodes are out of each other's LOS, this is the simplest example requiring relay.

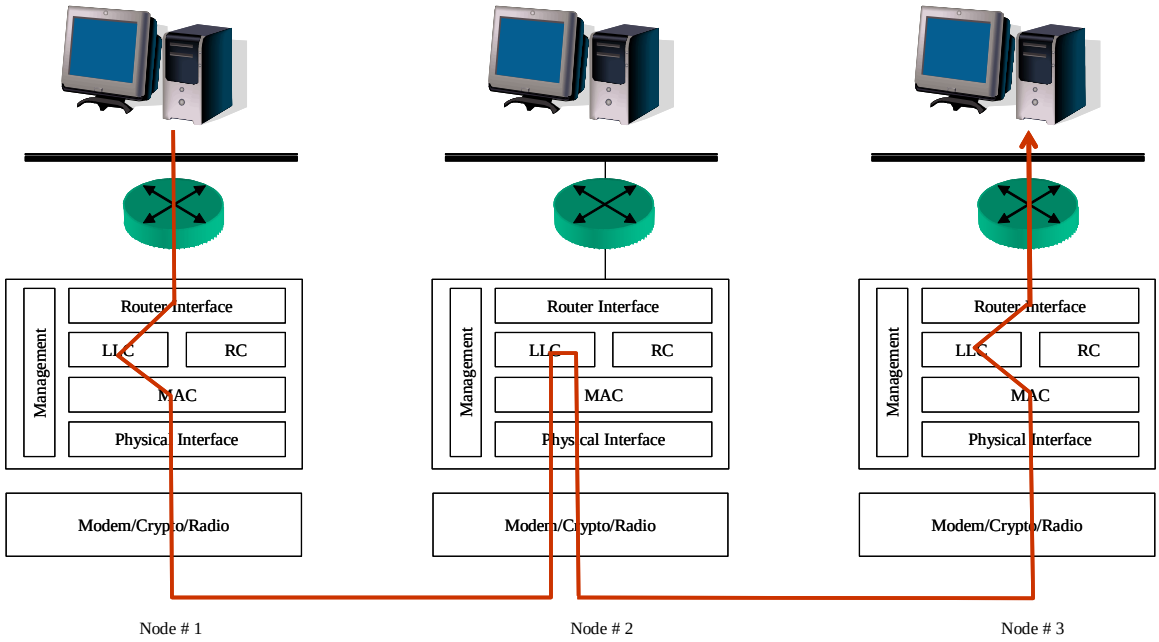


Figure A-3: Data transfer in a three-node linear network

An IP packet arriving at the router of the first node destined for the third will be sent as an Ethernet frame to the router interface of the attached MARLIN node. From the Ethernet header information, the router interface at node 1 will recognize the IP payload as a packet to be sent to node 3 and will send the packet to the LLC sublayer where it may be compressed, fragmented, and receive other processing, in preparation for its transmission on the air. The LLC sublayer at node 1, helped by the Relay Control, will know that packets sent to node 3 have to be relayed through node 2 and a MAC Protocol Data Unit (PDU) with next-hop destination #2 will be prepared for transmission with LLC PDU as payload. This MAC PDU will be passed via a synchronous serial physical interface to a modem, crypto, and/or radio, and other radio equipment. Finally, the packet, suitably encapsulated, will be transmitted on the air.

If received successfully at node 2, the de-encapsulated MAC PDU will be passed from the physical interface to the MAC sublayer. The MAC sublayer will pass the PDU payload to its LLC sublayer. The LLC sublayer will recognize from the reported header information that the destination is node 3 and, helped by the Relay Control, will know that node 3 is a direct neighbor. The LLC PDU will be sent to the MAC for relay transmission. A MAC Protocol Data Unit (PDU) with next-hop destination #3 will be prepared for transmission with LLC PDU as payload. From the MAC, the PDU is transmitted on the air through the physical interface.

If received correctly at the physical layer of node 3, the MAC PDU will be passed to the MAC sublayer and the MAC PDU payload, the LLC PDU, will be passed to the LLC sublayer. The LLC at node 3 may have to perform reassembly if the packet was fragmented and decompression if it was compressed, and eventually the original IP packet will be passed to

the router interface. Finally, the IP packet will be passed on to the router encapsulated in an Ethernet frame.

A.1.5 Purpose and Organization of this Document

This document specifies the operation, protocols, and algorithms of the MARLIN system in sufficient detail to ensure implementations of MARLIN complying with this STANAG will be interoperable.

The first Section of this document is this Introduction. The next Sections, in order, describe the Router Interface, Link Layer Control, Relay Control, Media Access Control, Automatic Repeat Request, and Physical Interface in detail. The final section of the main body provides the formats of headers and protocol data units employed in the system. Appendices provide details and supporting information.

A.2. Router Interface

A.2.1 Overview

IP data inbound to and outbound from the shipboard local area network (LAN) passes through one or more routers. MARLIN provides an alternative link to connect two or more ship networks and so must interface with these IP routers. The connection between each MARLIN node and its attached router is assumed to be a dedicated Ethernet. Furthermore, the Ethernet interface on a MARLIN node must be operated in promiscuous mode as defined in Section A.2.3 .

A.2.2 Router Configuration Options and Interaction with MARLIN

Routers connected to MARLIN systems must execute their own IP routing protocol in order to facilitate end-to-end connectivity. Specification of the routing protocol to be employed is beyond the scope of MARLIN specifications, but may for instance consist of Open Shortest Path First (OSPF) using a broadcast-type interface. The (hello) packets used by the router in this case must be presented to the MARLIN system with a multicast address. The system will then translate this to a broadcast address over the MARLIN network, thereby propagating to all reachable routers. The details of the LOS connectivity will be hidden from the router when using such a routing protocol. Alternatively, the IP router may run a mobile ad hoc network (MANET) routing protocol of its own, and only single-hop delivery will be required of the MARLIN nodes. Multi-hop operation is mandated by this standard; single-hop operation is optional.

With regards to OSPF and similar routing protocols, a “proxy” or modified version of the protocol may be helpful to enhance performance. The main reason for this is the need for long routing message intervals. For example, OSPF sends various messages, including OSPF “hello” messages, between routers at specified intervals. The advantage of a short hello interval is that it permits the network to respond quickly to connectivity changes at the router level. However, MARLIN is designed to deal with topology changes at the link-layer, as will be described in Section A.4 below, and for this reason it is beneficial to use a relatively long OSPF hello interval (typically on the order of one minute). A benefit of such an OSPF proxy is that with long hello intervals, the overhead associated with routing on the MARLIN network is quite small. No routing proxy is mandated in this document, but any proxy that is used must be transparent in its operation to allow interoperability with those units which do not have a proxy.

A.2.3 IP Addressing, Address Mapping, and Node Identifiers

All discussion in this section applies to both IPv4 and IPv6 addressing; either may be employed, although not simultaneously. The router interfaces attached to a MARLIN subnetwork must all be assigned IP addresses within a contiguous range (e.g. an IPv4 class C subnet). The least significant 8 bits of this IP address (e.g. the host address in an IPv4 class C subnet) is used as the MARLIN Node Identifier (ID). The specification of the MARLIN subnetwork, and the mapping between router interface IP address and MARLIN Node ID are static. These assignments must be configured on each MARLIN node.

The Ethernet MAC address on a MARLIN router interface is configured to be 00:*NodeID*:00:00:00:00, where the Node ID is described above. As such, an IP payload in arriving in Ethernet frames with destination field 00:*DestID*:00:00:00:00 will be forwarded to the MARLIN node with a Node ID of *DestID*. Likewise, when a MARLIN node forwards an Ethernet frame to its attached router, it uses 00:*SrcID*:00:00:00:00 in the source field, where *SrcID* is its own 8-bit Node ID.

The Ethernet Address Resolution Protocol (ARP), specified in Request for Comments (RFC) 826, provides a means for resolving IP addresses to Ethernet MAC addresses. Each MARLIN system must proxy responses to ARP requests for the addresses of all nodes within the MARLIN subnet. The proxy functions by intercepting all ARP requests (from the local router) for nodes within the subnetwork and replying with the MARLIN MAC address of the nodes in the network as the MAC address for reaching the node of interest.

All broadcast IP packets should be broadcast on the MARLIN network. IP multicast packets are also broadcast on the MARLIN network. The MARLIN Node ID of 255 is reserved for broadcast.

After stripping the Ethernet header, the IP payload of packets arriving from the router to the router interface are forwarded to the Link Layer Control (LLC) sublayer along with the ID of the egress node, obtained from the Ethernet header as just described. On the destination side, IP packets are forwarded from the LLC sublayer to the router interface, along with the ingress node ID for encapsulation into Ethernet frames and delivery to the router.

A.3. Link Layer Control

A.3.1 Overview

IP packets arriving from the MARLIN Router Interface have their Ethernet header stripped, are processed, and are queued until needed for transmission. Similarly, PDUs arriving from the MAC sublayer are queued until they are either relayed or reassembled and passed to the Router Interface for delivery to the attached router. The following are the responsibilities of the MARLIN LLC:

- Determine priority for each packet to be transmitted
- Determine need for retransmission/reliable delivery
- Data compression (optional) and decompression (mandatory)
- Fragmentation/reassembly
- Duplicate packet detection.
- Deliver logical protocol data units to MAC

The Link Layer protocol data unit (PDU) header is described in section A.8.6 . Along with the source, destination, and relay addresses already mentioned, the header contains several bit fields and flags. These indicate aspects of the LLC responsibilities listed above and described in detail below.

A.3.2 Data Priority

The MARLIN LLC determines the priority value of an incoming IP packet from amongst 16 priority values. The priority types and associated values are shown in Table A-1 below. MARLIN management packets are given highest priority, followed by IP routing packets, followed by user data packets, in order of priority. Priority and reliability of user data can be based on any information available in the IP packet and its payload, including protocol type, source and destination IP addresses, and TCP/UDP port numbers.

Table A-1: Link Layer Priority

Data Type	Priority	Notes
MARLIN Control Packets	15	Highest priority for network management
IP Routing Control Packets	13	IP routing information
Data PDUs	1-11	User data
Not Used	0,12,14	These values are reserved for future use.

When data packets of a single priority are considered, packets to be relayed should generally be sent before locally generated user data packets, except that each node may send one locally generated packet for every K relayed packets where K is the number of neighbors that have selected the node as a Multi-Point Relay (MPR), as described in section A.4.4 . Remotely

generated traffic is generally preferred to local traffic, as network resources have already been employed in moving the remote traffic at least one hop from source to destination. However, the 1 of K rule just described can be used to prevent local starvation of a relay node when network load exceeds capacity.

A.3.3 Compression/Decompression

Data compression is available in MARLIN and, if chosen, is implemented at the LLC sublayer of the transmitting node. Similarly, the LLC of the receiving node executes decompression of the PDU. Data compression is performed using the standard zlib specified in RFC 1950.

A.3.4 Packet Fragmentation and Reassembly

The MARLIN LLC also performs fragmentation and reassembly, if required. Fragmentation can occur at the source node as well as all intermediate nodes on the path to the destination node. Reassembly can occur at intermediate and destination nodes. The LLC must be able to reassemble a payload of up to 1500 bytes, the maximum size Ethernet payload.

Note that fragmentation and reassembly can also occur as part of the Automatic Repeat Request (ARQ) mechanism implemented in the Media Access Control (MAC) sublayer, described in section A.6.

A.3.5 Duplicate Detection and Time Slot Filling

Since all data packets pass through the LLC sublayer, it also performs duplicate detection. This is especially important for multi-hop relay and broadcast traffic. Duplicates can be detected by monitoring source ID, packet number, and fragment offset. Duplicate packets are dropped when detected.

Finally, the MARLIN LLC sublayer delivers data PDUs consisting of an IP packet payload (complete or fragmented) and a header to the MAC sublayer when needed. The MAC sublayer may request a PDU of a specific length in order to optimally fill its transmission time slot.

A.3.6 Queue Reporting

Before each MAC subsystem transmission, the LLC sublayer should calculate the current size of its transmit queues QSize, e.g. in bytes and an estimate of arrival rate, e.g. in bytes/second. QSize should include PDUs that have arrived from the attached router, and those that have been forwarded up from the MAC sublayer and require further relay. Since QSize is used as an indicator of load, the reported value should not include packet fragments awaiting reassembly for which the node is the egress node. The LLC maintains a separate queue for packet reassembly. If ASize is incremented for bytes queued in the transmit queue but is not decremented for bytes dequeued, then the arrival rate between times t_0 and t_1 is given by

$$\frac{\mathrm{ASize}(t_1) - \mathrm{ASize}(t_0)}{t_1 - t_0}.$$

An estimate ArrivalRate of the arrival rate can be updated according to

$$\mathrm{ArrivalRate}(t_1) \leftarrow \alpha^{t_1 - t_0}\mathrm{ArrivalRate}(t_0) + (1 - \alpha^{t_1 - t_0})\frac{\mathrm{ASize}(t_1) - \mathrm{ASize}(t_0)}{t_1 - t_0},$$

where the coefficient α is chose to satisfy $\alpha^{M \times T} = 0.5$ for the configured averaging window M $\times$ T. An averaging window of duration equal to one cycle time, defined in section A.5.3 , shall be used.

A.4. Relay Control

A.4.1 Overview

For data transfer to two-hop and further nodes, reaching beyond LOS, it is necessary to relay data. To limit the amount of traffic broadcast, and in particular the traffic exchanged to organize the network, the Relay Control sublayer of each MARLIN node selects a subset of its one-hop neighbors via which it can reach all two-hop neighbors (See section A.5.2). These neighbors are called a multi-point relay set, denoted MPR. The concept and nomenclature are borrowed from Optimized Link State Routing (OLSR). The Relay Control sublayer implements functionality similar to OLSR with link state extensions. As the information broadcast by the MPRs is disseminated through the network, nodes can determine the membership of the complete network, find shortest paths to each node, and determine their preferred next-hop relay to every node in the network.

If the link quality LQ denotes the estimate of the probability of successful packet reception for a transmission from one node to a neighbor and the inverse link quality ILQ is the probability of successful packet reception on the return, then the link cost used for the link between the nodes is the symmetric expected transmission count ETX, given by:

$$\mathrm{ETX} = \frac{1}{\mathrm{LQ} \times \mathrm{ILQ}}.$$

This is the expected number of transmissions required to transmit from one node to the other and then receive an acknowledgment in return. The link quality is known to a node based on information obtained by querying the MAC subsystem (see section A.5.9)

A.4.2 Relay Control Packet Transmission

The Relay Control (RC) sublayer is responsible for producing RC control packets. This control packet is analogous to the Topology Control packet used by OLSR. The RC PDU contains a list of all the good two-way neighbors of the reporting node, the respective link states of each neighbor, and the respective ETX values for the links to each neighbor.[1] The control packet also includes a sequence number which is incremented by +1 with each transmission. Each node will generate a RC PDU whenever a topology change leads to a change in its two-way neighbor list, and at least once every RC reporting interval T_{RI}, usually equal to one cycle time. The RC control packet is passed to the LLC for encapsulation in an LLC header and delivery to the MAC. The RC control PDU is transmitted as a broadcast packet with the reliability flag set to false.

A.4.3 Relay Control Packet Processing

RC PDUs generated by other nodes and transmitted by neighbors are received at the MAC and passed to the LLC. Duplicate packets are dropped by the LLC. If the receiving node is

[1] A node i is not required to report a neighbor j if ETX(i, j) + ETX(i, k) > ETX(j, k) for all other neighbors $k \neq j$. For in this case, node i is not even a candidate to relay for j, assuming symmetry of ETX. If j can not hear k, then ETX(j, k) = ∞, and j must be reported. If j can hear k, then ETX(j, k) can be estimated from the one-way link quality estimates reported by j and k.

identified as a MPR for the transmitting neighbor, then the LLC PDU will be queued for retransmission. The RC PDU payload is then passed to the RC sublayer. Each successive sequence number replaces the information with the same originator ID, and old packets, determined by monitoring the sequence number, are ignored.

The RC sublayer uses the information contained in the received RC control packets, together if necessary with the one-hop connectivity and link costs obtained from the MAC, to determine full membership in the network. A standard shortest path algorithm can be used to determine the best next-hop to each member of the network. Link costs ETX may be assumed to be symmetric for purposes of this computation. If the RC sublayer does not receive a control packet from an originating node within a suitable RC hold interval T_{RH}, nominally $3 \times T_{RI}$, the data reported by this node is considered out of date, is dropped, and is not used in subsequent shortest path computations. If after removing the information, there is no longer a path to a node, then that node is considered unreachable. When a RC control packet designates a node as a MPR, the originator of that packet is reported as a Multi Point Relay Selector (MPRS) in subsequent packets.

A.4.4 Multi-Point Relay Selection Algorithm

Information reported in the RC control packets originating at one-hop neighbors, namely the symmetric ETX link costs, can assist the Multi-Point Relay (MPR) selection executed at each node. Each node shall use a shortest path algorithm with ETX as link cost to determine a preferred next-hop towards each two-hop neighbor. In OLSR with link quality extensions, each good, two-way neighbor that is a preferred next-hop towards at least one two-hop neighbor is selected as an MPR for that node. The same selection criterion is used here. A node selected as an MPR is reported in subsequent RC Control packets accordingly.

A.4.5 Data Relay

When a node transmits a broadcast packet, it is heard by all of its one-hop neighbors. Only its RC sublayer MPR neighbors are required to retransmit it, in order to reach two-hop neighbors. This process is then repeated in order to reach nodes even further away. More generally, when a node transmits a broadcast packet, only its RC sublayer MPR neighbors that have not already transmitted it need to retransmit.

A.5. Media Access Control

A.5.1 Overview

The media access control (MAC) sublayer coordinates transmissions on the shared channel using a synchronous time division media access (TDMA) scheme called Distributed Slot Reservation Media Access (DSRMA). Each MARLIN node maintains a fixed allocation of time slots in each frame, although the specific assignment of slots may change if the topology changes. The remaining slots are allocated to the nodes in a distributed fashion in proportion to their reported demands. Nodes typically acquire additional slots when their demand increases relative to their neighbors and release slots when their demand declines. Certain slots can not be reserved and are set aside for new joiners to enter the network using random access. The MAC layer identifies a subset of its neighbors, called MAC Layer Arbiters (MLA) for mediating slot requests. These need not be the same as the MPR neighbors elected by the RC sublayer, but like MPS the MLAs are required to cover the set of two-hop neighbors. The MAC sublayer also implements an automatic repeat request (ARQ) scheme to facilitate reliable data delivery.

There are four principal states of operation in DSRMA. Existence in one of these states depends on a node's point-in-time in the network, its proximity to and type of neighboring nodes, and its success in reserving TDMA time slots. The state diagram for DSRMA is shown in Figure A-4, and the states are described below.

- **Listen:** in this state, the node listens to gain local information about the network. This state is entered immediately after start-up. A listening node has no transmission slots and no two-way neighbors, but it may have one-way neighbors.

- **Beacon:** a state in which a node has no good one-way or two-way neighbors, but does have a transmission slot, namely a beacon slot in which it will transmit once per frame to announce its presence.

- **Join:** in this state, a node has at least one good one-way or two-way neighbor, but no persistent time slots reserved for transmission. In this state, a node will periodically use a random access (RA) slot to transmit a join request in which it attempts to reserve a non-RA slot for subsequent transmissions.

- **Participate:** This is the desired end-state. A node in this state has at least one two-way neighbor and at least one non-RA slot to use for transmission. Once a node has obtained a non-RA slot, it can use it to reserve additional time slots for transmission. Additional use of the contention-based RA slots is not required unless the node loses connectivity with one of its neighbors.

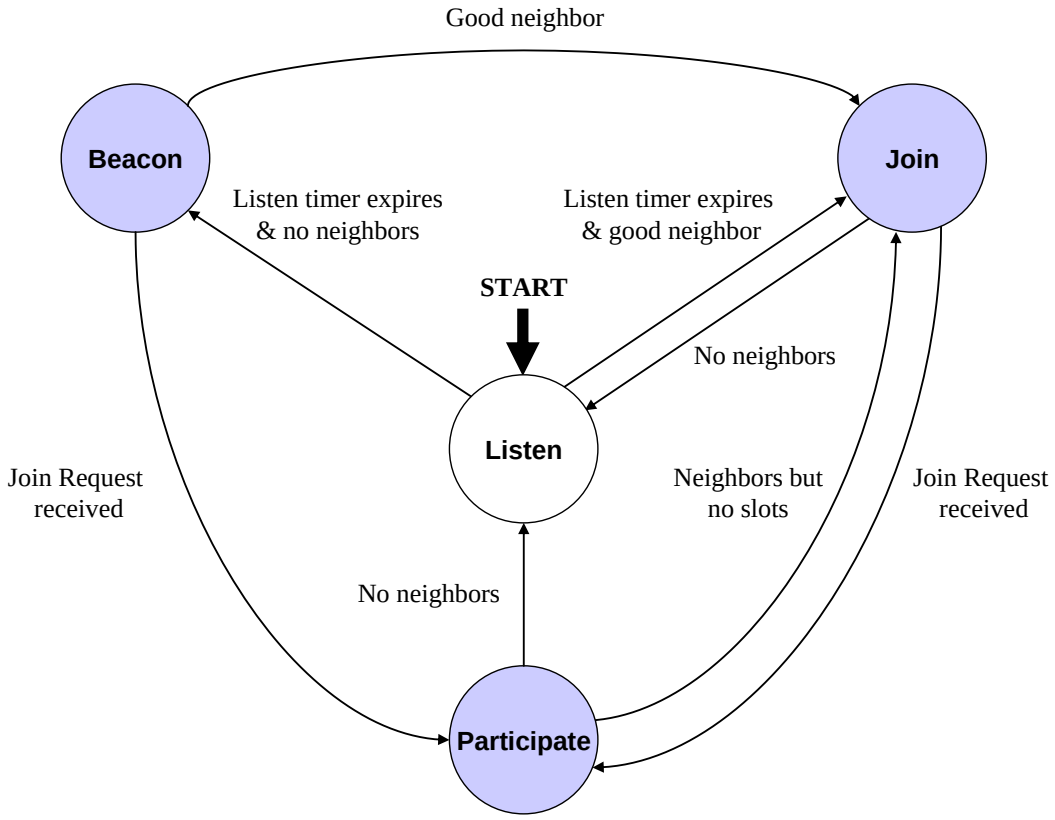


Figure A-4: DSRMA State Diagram

Details of the qualifiers and requirements mentioned in the four states are presented in the following subsections. In particular, the following topics of the MARLIN MAC sublayer will be described in detail:

- Types of neighbors, Multipoint Relays, and Multipoint Relay selectors.
- TDMA timing (cycle and slot structure).
- Slot types and management.
- Subnetwork formation and maintenance.
- PDU types and transmission ordering.

A.5.2 Types of Neighbors

DSRMA distinguishes between various types of neighbors. Each node monitors and reports the quality of receptions from all nodes it has recently heard. Based on the link quality assessment, the neighbor is declared a good neighbor, a bad neighbor, or a non-neighbor. The latter is logically equivalent to a node not being reported at all. Good neighbors are further classified as asymmetric (one-way) or symmetric (two-way). If a node assesses its link quality from a neighbor as good, but that node is not reporting the return link as good, the link is asymmetric. If both nodes report the link to the other as good, then the link is symmetric. User data is exchanged only with good, symmetric neighbors. Good, symmetric neighbors are also called one-hop neighbors.

Two-hop neighbors are nodes that can be reached by relay through a good, symmetric, one-hop neighbor (i.e., a one-hop neighbor of a one-hop neighbor) which are not themselves one-hop neighbors.

With MARLIN, a (possibly empty) subset of one-hop neighbors is selected to distribute MAC control information and mediate slot requests. These neighbors are called MAC Layer Arbiters (MLAs). In an all-informed network where all nodes are direct neighbors no mediation is required and the MLA sets will be empty. More generally, the MLA set must be selected to cover the node's two-hop neighbors. It is desirable that this set be minimal to expedite slot acquisition. One choice for MLA is to re-use the MPR set selected by the RC subsystem, but this is not required. Like MPR selection, this decision can use its own independent criteria to select this set and no specific selection algorithm is mandated. The set of nodes which have selected a given node as a MLA is called the MAC Layer Arbiter Selector set.

In summary, DSRMA distinguishes between the following neighbor types:

- Non-neighbors
- Bad neighbors (interferers)
- Good neighbors:
  - Asymmetric (one-way) neighbors
  - Symmetric (two-way) neighbors but not MLAs
  - MLAs

A.5.3 TDMA Timing: Cycle and Slot Structure

MARLIN uses a time division multiple access (TDMA) scheme for sharing channel bandwidth on a single frequency amongst the member nodes of a MARLIN network. To achieve synchronization on time slot, cycle, and frame boundaries, each MARLIN node must have access to a source of coordinated universal time (UTC), such as Global Positioning System (GPS).

Time on the shared channel is divided into cycles which are further divided into time slots. Each participating node reserves one or more time slots in a cycle for transmission to one or more of the nodes with which it has direct bidirectional connectivity. Slot 0 of cycle 0 begins at midnight January 1, 1970 (Unix time 0 or UTZ). If T is the slot duration in seconds, slot 1 of cycle 0 begins T seconds after UTZ, slot 2 of cycle 0 begins 2T seconds after UTZ, and so on. Slot numbers are recycled after M slots where M is the number of slots in a cycle. The cycle duration is $M \times T$ seconds. Thus, slot 0 of cycle 1 begins $T \times M$ seconds after UTZ, slot 1 of cycle 1 begins $T \times M + T$ seconds after UTZ, and so on. Cycle numbers are recycled after 16 cycles. This is illustrated in Figure A-5.

Setting t_z to be time in seconds since UTZ, the current slot number, *slot,* and cycle number, *cycle,* are then the given by the integers:

$$slot = \lfloor t_z / T \rfloor \bmod M, \text{ and}$$
$$cycle = \lfloor t_z / T \times M \rfloor \bmod 16.$$

A new slot begins whenever the division of t_z/T is an integer value.

Slot timing is defined in terms of the over-the-air transmission and the MARLIN Node Controller is responsible for ensuring that the signal from its transmission does not interfere with reception in another node's slot.

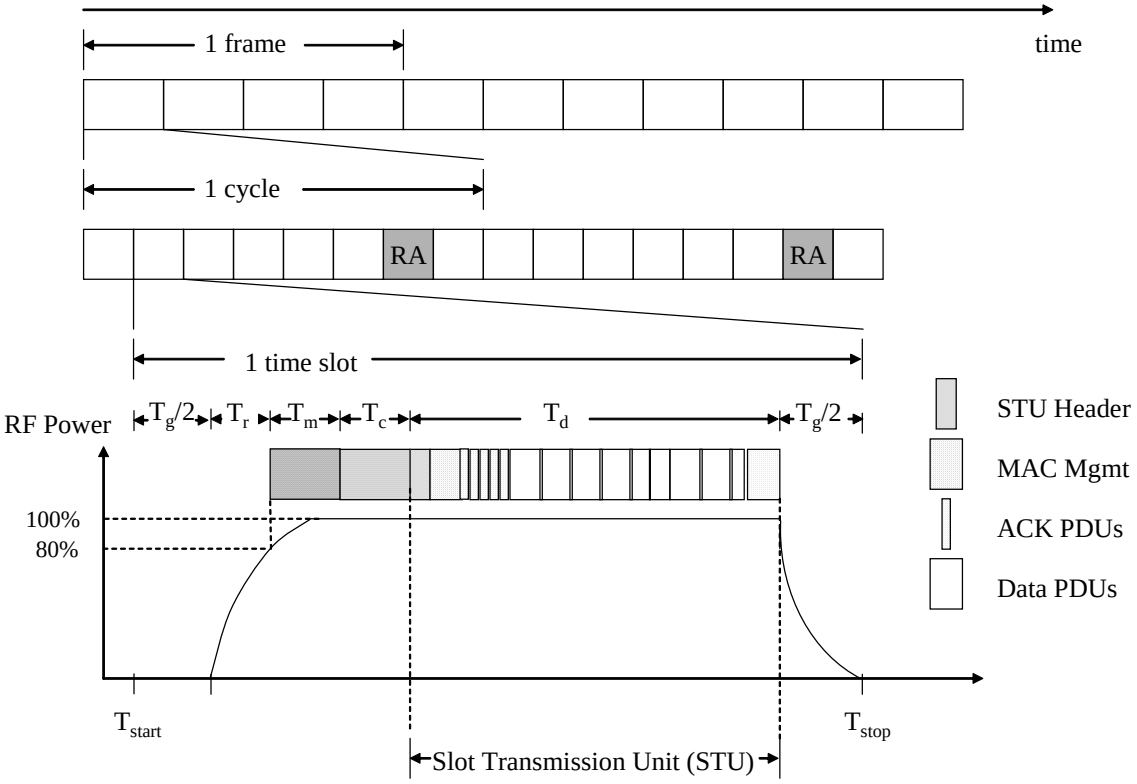


Figure A-5: MARLIN TDMA frame structure

Each time slot is divided into several intervals:

- T_r – time required for a transmitter to ramp up to 80% of its transmit power,
- T_m – time required for modem synchronization,
- T_c – time required for crypto synchronization,
- T_d – time during which data is transmitted, and
- T_g – a guard interval equally distributed between the beginning and end of the slot.

The on-air transmission occupies the slot from the time the radio is keyed and begins to ramp up, until the time the radio unkeys. Included in the on-air portion of the slot are modem preambles (used for synchronization), crypto preambles (if required), and portion devoted to data.

Fundamentally, the transmitter must ensure that its on-air transmission does not extend beyond the defined slot boundaries. In systems which support full duplex serial interfaces, it is permissible however to begin sending serial data from the node controller to the crypto or

modem prior to the beginning of the slot and to receive data from the crypto or modem after the end of the slot. The unoptimized delay is the sum of two terms. The first is the time from when the first bit is sent from the node controller to the time the signal appears on air. The second is the delay from the time the signal ends until the last bit is received by the node controller. These delay elements can be compensated to optimize transmissions and maximize the duration of the on-air signal while still fitting within the slot when the physical interface is full duplex.

In order to ensure interoperability with networks with heterogeneous equipment, nodes must agree on the maximum total unoptimized delay. This total delay value for the system is the sum of: the worst case (longest) unoptimized transmit delay from any node in the network; the worst case (longest) unoptimized receive delay from any node in the network (and not necessarily the same as the node with the longest transmit delay); and the longest receiver recovery time or radio switching time. Note that the delay values considered are not simply the measured delay values for transmit and receive delay, but rather their values after optimizations have been applied. For illustration, consider a node with a measured transmit delay, from onset of serial data to signal on-air of 50 ms. If that node optimizes its transmission by advancing the time at which it sends serial data by 30 ms, the unoptimized transmit delay is 20 ms. By sharing the worst case value for unoptimized delay, nodes can compute the longest slot that they can transmit and ensure that the most disadvantaged receiver will still be able to successfully receive successive transmissions from the most advantaged transmitter followed by the most disadvantaged transmitter.

During the data portion of the slot, a transmitting node will transmit a slot transmission unit (STU). The STU consists of a STU header followed by a number of MAC protocol data units (PDUs). These MAC PDUs include management PDU types and data PDUs. Protocol data units are added to the STU in priority order, starting with the highest priority 15. Data PDUs awaiting retransmission from the ARQ queue are added before those from the LLC. As described in section A.3.2 , remotely generated packets are generally added to the STU before locally generated packets, except that each node is guaranteed a certain fraction of locally generated packets to prevent starvation. PDUs are added until either there is no room left in the STU or there are no more PDUs awaiting transmission. More details on the PDUs being transmitted and their formats are provided in section A.8.

A.5.4 Slot Merging

In the event that a node owns two or more consecutive slots, the node may send a single transmission spanning the adjacent slots to minimize overheads. When slots are merged, a single STU is transmitted. Receiving nodes must be able to correctly recognize and decode these merged slots.

A.5.5 Types of Slots

There are three types of transmission slots used in MARLIN:

- Fixed Allocation

- Demand Assigned
- Random Access

A MARLIN node will request and release transmission slots in proportion to its own and reported traffic requirements. Thus, bandwidth is dynamically allocated as more or fewer slots are reserved. Each node, however, tries to maintain a minimal number of fixed allocation (FA) slots, typically 1 or 2 per cycle, but always at least 1, regardless of demand. Each cycle also includes 1 or 2 Random Access (RA) slots that can not be reserved and are available for nodes such as new joiners that do not have any reserved slots to request them.

A.5.6 Dynamic Bandwidth Allocation

As part of its operation, each node reports its slot demand and obtains the slot demands of its neighbors and its two-hop neighbors. The demand reported by a node is the sum of the LLC transmit queue arrival rate in units half-slot per cycle, the size of the LLC transmit queue in units of half-slots, and the size of the packets scheduled for retransmission by the ARQ mechanism in half-slots, rounded up to the nearest integer. That is, with each transmission node i will report:

Demand(i) = Transmit Arrival Rate + Transmit Queue + Retransmit Queue.

A half-slot is computed as one-half of the product of the slot duration and burst rate, $0.5 \times T \times R$.

Each node will obtain these values from its one-hop neighbors j and will also report the sum of demands in its one-hop neighborhood.

$$\text{Target(i)} = \max\left(\text{Fixed Allocation}, \frac{(M - M_{RA}) \times \text{Demand(i)}}{\max\limits_{j \in N1(i)} \sum\limits_{k \in N1(j)} \text{Demand(k)}} \right)$$

Here, M denotes the number of slots per cycle, M_{RA} denotes the number of RA slots not available for reservation, and N1(j) denotes the set of one-hop neighbors of j together with j itself.

A node whose slot allocation is less than its fixed allocation will always request additional slots if any are available. A node whose slot allocation is less than its fixed allocation may make a priority slot request for an additional slot, described in the next subsection. A node whose slot allocation is less than its target allocation may request additional slots if any are available. If a node has an allocation that is greater than its target allocation, and it can drop slots without reducing its allocation below the number of slots acquired in the last two cycles, then it must identify a slot for release and drop the slot within one cycle time. It is recommended that a node release a slot only as it transmits in it, to allow a full cycle time for another node to acquire

it. No algorithm for deciding which slots to release and which to request when more than one is available is mandated.

The intention of this scheme is that there should be many more slots per cycle than nodes in the network so that each node can acquire its allotment of FA slots and there still be additional DA slots to be exchanged in response to the mechanism just described. In an all-informed network it is a requirement that the number M of slots per cycle be greater than the sum of the number of FA slots for each node and the number of RA slots per cycle.

A.5.7 Slot Request and Approval

With each transmission, a MARLIN node transmits its slot ownership table (SOT), a bit-vector indicating the slots it is using for transmission. If a node is a MLA for a neighbor, it will report the SOT of that neighbor as well. Slot ownership tables for interfering nodes, including one-way neighbors, are summarized. In this way, slot ownership and availability is distributed locally. Requests for additional slots are made explicitly. A neighbor signals approval of a slot request made by advertising the slot in the SOT reported for the requesting node. Rejections are implicit when the same slot is advertised for another, competing node. Slot releases are indicated by changing the appropriate bit in the SOT by the transmitter. DSRMA supports two types of slot requests: priority and normal. Priority requests can preempt an existing allocation, whereas normal requests can not.

When a node is first joining a network and has no slots at all reserved, a priority slot request is made. A node with less than its full fixed allotment may also make priority slot requests. These preemptive slot requests must be honored by any other node owning or requesting that slot, unless conflicting priority slot requests are scheduled. In the event of a conflict, nodes approve the lowest node ID for a conflicting priority slot request. When a node makes a priority slot request, all of its neighbors that receive it must respond with an approval or rejection. The requesting node must wait for all its neighbors to have a chance to respond before it may transmit in the slot. Tacit approval is assumed for neighbors (e.g., one-way neighbor) that do not indicate they can hear the requesting node, and at least one explicit approval is required before the requesting node can treat the request as approved.

Ordinary slot requests can be made for any slot which is available when a node's allocation is below its target. The requesting node's MLA neighbors must respond to ordinary requests with approval or rejection, the latter only occurring in the case when a conflict is detected. Note that in an all-informed network, for example, a requesting node has no MLA neighbors and may request a slot and then transmit in that slot without waiting for an acknowledgement. More generally, a node making an ordinary slot request must wait for its MLA neighbors to approve the request before transmitting.

An additional use of the priority slot request is allowed, although not mandated, under this standard. A node which has a persistent one-way neighbor, particularly one in a partition different than its own, may attempt to establish a two-way relationship. To do so, it must transmit in a time slot in which the neighbor is available to receive. As explained more fully in

the next subsection, such a slot may be determined by transmitting in the RA slot position as determined by the partition id (PID) reported by the one-way neighbor if PID > 0 (a “help merge”) or the RA slot in position by the node’s own id if the reported PID = 0 (a “help join”). Obtaining such a slot may require the helper node to make a pre-emptive slot request, if it has no alternative. If exercised, this option should be employed sparingly and with suitable back-offs to avoid disrupting the normal operation of the network.

Nodes must monitor the slot ownership tables reported by their neighbors for possible conflicts with slots that have been successfully requested and are being used for transmission. In the event a conflict between the SOT reported by another node and their own is detected, a node must drop any slots reported as owned by other nodes.

A.5.8 Network Organization and Maintenance

The self-organization and maintenance of the MARLIN network are addressed in this section. This section describes the steps taken by a node from start-up, how an isolated node joins an organized network, how two organized network (or network partitions) can merge when they come into range, and what happens when they split. It is helpful to refer again to the DSRMA state diagram in figure A-4.

The partition identifier (PID) is used to help organize and identify MARLIN networks and plays an important role in partition joining and merging. In general, the partition identifier is the MARLIN ID of one of the nodes in the partition and when two partitions merge they agree on a common partition identifier. If a single node is not in a network, it will report 0 as a PID to indicate it is a singleton. If a singleton node not in a network hears another node that is in a network transmit it will adopt the partition ID used by the node it hears before transmitting a join request in an appropriate slot. If two distinct partitions come into range, they will adopt the lower partition ID. The partition identifier is updated when a partition splits.

The first DSRMA state that a MARLIN node enters after start-up is the **Listening** state. When first activated, a node must listen for a minimum of one cycle before transmitting. This allows the node to detect existing networks or other nodes.

If neither existing networks nor other nodes are detected, the MARLIN node will enter the **Beaconing** state. While in this state, the node reports a PID of 0 to indicate it is a singleton and can not reach any node other than itself. As defined in appendix 3, the position of the RA slots in the cycle are determined by the PID. When in beacon state, the node should use its own id as a seed to this algorithm and determine the positions of the RA slots accordingly, select a non-RA slot at random, and transmit in that slot. The transmitted beacon consists of a STU header and DSRMA management packet. The node should beacon for two cycles in this non-RA slot. During the second transmission, the node should select a new non-RA slot, drop the first, and announce its ownership of the new slot. It will then transmit for two cycles in the new slot and repeat this operation. The node iteratively beacons for two cycles and chooses another slot, until another MARLIN node or network is discovered. A joining node may announce it can hear the beaconing node and transmit a slot request; if heard, the

beaconing node should grant the request and transition to participate state. Alternatively, the beaconing node may hear a one-way neighbor and transition to join state.

A node in **Join** state has at least one one-way neighbor and no slots. A joining node will select one of its two-way neighbors if there are any, or else one of its one-way neighbors (if there are more than one), determine a slot in which the target node is available to receive, and schedule a transmission in that slot. A target node should always have at least one slot available in each frame available to receive since the RA slots are reserved for that purpose. If the target is reporting a PID > 0, then an RA slot location can be determined from the PID, as defined in appendix 3. If the target is reporting PID = 0, then the target is itself in either in Join or Beacon state and an RA slot position can be determined by seeding the algorithm in appendix 3 with the target node ID. A joining node will use a PID of 0 to indicate it does not have connectivity to any node other than itself. In its join request, the requesting node should make a priority request to reserve a transmission slot. The requested slot should be a non-RA slot. The node should then listen for at least one cycle time to determine if its join request was successful. Success is achieved when the target node recognizes the requester as a two-way neighbor and the slot request is granted.

If this request is not successful, a random back-off scheme must be employed before attempting another join request transmission since a collision may have occurred with the first request. A specific back-off scheme is not mandated by this profile, however. A node will remain in Join state as long as it has at least one one-way neighbor. If a node in Join state determines it no longer has any one or two-way neighbors it transitions to Listen state.

A node is in **Participate** state if it has at least one two-way neighbor and at least one slot reserved for transmission. Such a node is in an organized network. A node i transitions to Participate state when it hears a neighbor j report it as a good two-way neighbor and identifying a slot reserved for node i's transmissions. When transitioning, node i will set PID(i) = PID(j) unless PID(j) = 0 (i.e., node j is itself in the beacon or the join state). If PID(j) = 0, then node i should set PID(i) = j. In the first case, node i will make a probationary acceptance of its new partition ID, setting a probation timer that will expire after a period T_{PID}, nominally equal to the Relay Control Hold Time T_{RH}. If the probation timer expires before a route to the PID is discovered, then node i will elect a new PID as described below.

When in Participate state, a node must engage in partition maintenance. This activity helps to manage the merging and splitting of connected components of the network. First, the activities at node i upon reception from a good, two-way neighbor j are described. If PID(i) = PID(j), there is nothing to do.

- If PID(i) < PID(j) and PID(i) is reachable or probationally so, then node i will not change its PID.
- If PID(i) < PID(j), PID(i) is not reachable and is not on probation, and PID(j) is reachable, then node i will set PID(i) = PID(j) after first setting a blocking timer on its old PID. The blocking timer will expire after a RC hold time T_{RH} or if the blocked PID again becomes reachable. When a partition identifier is blocked it is treated as ∞ when compared with another. Setting this blocking timer prevents node i from immediately switching back to

its old PID should another neighbor report it. Upon expiration of the blocking timer, the blocked value will again become eligible to be a PID.

- If PID(i) < PID(j), PID(i) is not reachable nor on probation, and PID(j) is not reachable, then node i will make a probationary acceptance of PID(j), setting PID(i) = PID(j) after first putting a blocking timer on its old PID to prevent immediately switching back as above. In addition, node i will set a probation timer to expire after a RC hold time period. If the probation timer expires before a route to the new PID is discovered, then node i will elect a new PID, as described below.
- If PID(i) > PID(j) and PID(j) is reachable, then set PID(i) = PID(j).
- If PID(i) > PID(j) > 0 but PID(j) is not reachable, then set PID(i) = PID(j) and set a probation timer as above.

Each time node i transmits, it checks that PID(i) is still reachable, or if not reachable is on probation. If PID(i) is not reachable and the probation timer has expired, then node i selects the smallest index k from among all nodes that are currently reachable including itself. If k < PID(i), then node i sets PID(i) = k before transmitting. If k > PID(i), then node i will do the same but will also set a blocking timer on the old PID(i) before switching.

When a node changes its partition identifier, it changes the positions of the RA slots in the cycle. If a node has one of the new RA slots reserved for transmission, it should transmit in that slot at the next opportunity. The slot will be dropped but only after a final transmission. If the node has dropped or will be dropping all its slots as a result of the PID change, it should make a priority request for a new non-RA slot at each opportunity. If a node drops all its slots, it should transition to join state.

Even in the absence of a PID change, a node may drop all of its slots due to conflict avoidance. If a node in participate drops all its slots but still has at least one good one-way neighbor, it transitions to Join state. If it finds it no longer has any neighbors, it transitions to Listen state.

A.5.9 Link Quality Sensing

Each MARLIN node must maintain a link quality measure LQ (j) $\in$ [0,1] and an integer link state LS(j) $\in$ {0, 1, 2, 3} for each neighbor j. The meaning of the states is

LS = 0	Not heard
LS = 1	Non-neighbor
LS = 2	Bad neighbor
LS = 3	Good neighbor

The link quality is an estimate of the probability of successful reception from this neighbor. Before a node j is first heard, LS(j) = 0; When node j is first heard, link quality should be initialized by setting LQ(j) = LQ_0 relatively large and LS(j) = 3. This gives the benefit of the doubt to new joiners. Link quality decision thresholds L_{mn} for transition from state m to state n are specified. Each time a node j is heard from, the receiving node should update its link

quality estimate. After this, the node must execute the following decision logic to update the link state.

If LS(j) = 3 and LQ(j) < L_{32}, set LS(j) = 2.
If LS(j) = 2 and LQ(j) < L_{21}, set LS(j) = 1.
If LS(j) = 1 and LQ(j) > L_{12}, set LS(j) = 2.
If LS(j) = 2 and LQ(j) > L_{23}, set LS(j) = 3.

The decision thresholds must satisfy $0 < L_{21} < L_{12} \leq L_{32} < L_{23} < 1$. The effect of the offset thresholds is to introduce hysteresis into the neighbor decision rule to reduce link state flapping. The initial link quality must satisfy $LQ_0 > L_{32}$.

The mechanisms just described are triggered by receptions from neighbors. The case that there are no further receptions must also be addressed. Each time a node j is heard from, an expiration timer for that neighbor is set. If a node is not heard from within 3 cycle times, the link quality and link state are updated by setting $LQ(j) = LQ_1 < L_{12}$ and LS(j) = 1. If a neighbor expires in this manner, a lost neighbor timer is set for the neighbor equal to 8 cycle times in the future. If the lost neighbor timer expires for a neighbor j, a node should set LS(j) = 0; this will allow it to be treated like a new joiner when it is next heard.

To determine link quality, each node must calculate the proportion of each neighbor's recent transmissions that were successfully received. If a neighbor has been heard from recently and is expected to be heard each time it transmits, then a node can determine a missed slot count MSC by comparing the sequence number in the STU header when received with the last sequence number heard, i.e.

MSC = Current Sequence Number – Last Sequence Number – 1.

Since 3 bits are used for the STU sequence number, this difference is modulo 8. If there are known conflicts with transmissions (as there could be, for example, with new joiners and one-way neighbors) then the number of missed transmissions may have to be estimated. Given an estimate of missed transmissions, the node should estimate its link quality for this neighbor by setting

$$LQ(j) \leftarrow \beta^{MSC+1} LQ(j) + (1 - \beta),$$

where β is a smoothing factor between 0 and 1.

A.6. MAC Reliability – Hop by hop ARQ

MARLIN implements reliability by means of a hop-by-hop ARQ protocol. If the reliability flag in the MAC Data header is set, the receiving node will acknowledge reception of the packet by sending a MAC Acknowledgement PDU after successfully receiving the PDU. Note that the acknowledgement is only hop by hop and not end-to-end. Reception of the acknowledgment packet indicates that the packet has successfully been transmitted to the next hop but does not guarantee delivery to the final destination. If an ACK packet is received indicating that the first transmission of one or more PDUs requiring reliable delivery was not received successfully, then that PDU should immediately be identified for retransmission and retransmitted at the first available opportunity. If an acknowledgment is not received within 2 cycles of the transmission of a packet with the reliability flag set, the sending node will retransmit the packet. Nodes may choose to retransmit after the intended recipient has had an opportunity to acknowledge reception. Nodes must attempt at least one retransmission of a PDU with the reliability flag set and may choose to drop the PDU after that.

A.7. Physical Interface

A.7.1 Crypto/Modem/Radio Interface

This MARLIN node controller standard specifies the timing of the signal transmitted over the air and is specifically designed for TDMA networks. However, this standard does not define the specific characteristics of the signal transmitted over the air. An MARLIN node controller may be embedded within a radio, in which case the interface may be entirely proprietary. In other cases, the MARLIN node controller will be a stand-alone device, connecting to the radio through a cryptographic device and external modem. The vast majority of link cryptographic devices in service require the use of a synchronous serial interface. Hence a synchronous serial interface is defined within this standard. In addition, an ethernet interface is presented as a desirable option.

A.7.2 Synchronous DCE interface

If the MARLIN Node Controller is not integrated into a radio, the MARLIN Node Controller shall implement a synchronous Data Communications Equipment (DCE) interface that is compatible with EIA-232, EIA-422 or Mil-Std-188-114A.

A.7.3 Optional Ethernet interface

The modem may provide an Ethernet interface.

A.7.4 Time

Because the MARLIN TDMA scheme requires precise timing in order to coordinate transmissons on the shared channel, each node must have access to a high precision source of time information. An MARLIN node controller must be able to convert this time source to universal coordinated time (UTC) in order to synchronize its TDMA timing with the other nodes within the network.

To achieve synchronization on time slot and cycle boundaries, the TDMA scheme employed in MARLIN requires an external source of UTC. Examples of precision time sources include Global Positioning System (GPS) or appropriately configured Network Time Protocol (NTP) servers. These time references, if they do not natively provide UTC, can be converted to UTC for use by MARLIN.

In order to be compliant with this specification, the MARLIN Node controller shall be able to determine time to within $\pm$ 10 ms of UTC.

A.8. PDU and Header formats

All fields are specified with least significant byte first, least significant bit first.

A.8.1 STU Header – Part 1

Bit Field	Number of Bits	Comment
Slot Synchronization Pattern	32	0x749713BF
Transmit Node	8	ID of transmitter
Partition Identifier	8	
UTC Flag	1	
Sequence Number	3	Bits 1-3
Cycle number	4	Bits 4-7
Slot number	8	
Slot Size	16	Slot size in ms
CRC	16	

Slot Synchronization Pattern
The slot synchronization pattern is used for achieving frame synchronization with the data for streaming media. This is considered to be a preamble and is not considered to be part of the STU Header 1 when calculating the CRC.

Transmit Node
The MARLIN node ID of the STU transmitter

Paritition Identifier
The partition identifier used by the STU transmitter.

UTC Flag
Flag to indicate access to precise Universal Coordinated Time. Set to 1 to indicate time accuracy of better than ± 5 ms.

Sequence Number
The sequence number is incremented with every transmission and is used to assist in determining if the node receiving the transmission has missed transmissions from this node.

Cycle number
This provides the cycle number of this STU.

Slot number
This provides the slot number of this STU.

Slot Size
The Slot size in ms. Together with other values, this allows a node to join a network without knowing the slot size, number of slots per cycle, or number or number of RA slots per cycle.

CRC
The 16 bit CRC is computed over the STU Header Part 1, beginning with the Transmit Node field and ending at the end of the Slot Size prior to the CRC. The computation of the CRC is described appendix 2.

A.8.2 STU Header – Part 2

Bit Field	Number of Bits	Comment
Spare	1	Set to 1
Extension bit	1	1 if 64 or more PDUs, 0 otherwise
PDU Count	6(14)	14 Bits if 64 or more packets
Size (PDU 1)	12	
Size (PDU 2)	12	
…		
Size(PDU N)	12	
Pad	0(4)	Pad to byte boundary if N is odd
CRC	16	

PDU Count
The PDU count is the total number of MAC layer PDUs transmitted in this STU. Note that this includes both management and data PDUs with the DSRMA Management PDU counted as PDU number 1. If there are 63 or fewer PDUs, this number is reported using the most significant 6 bits of the first byte. If there are 64 or more PDUs, the extension bit is set to 1 and the PDU count is represented as a 14-bit binary number with the least significant 6 bits taken from bits 2-7 of the first byte of the STU header and the remainder from the second byte. For example, suppose a STU contains 567 PDUs. The binary representation of the decimal 567 is 1 0 0 0 1 1 0 1 1 1 most significant bit first. Writing least significant bit first, the binary representation of the first two bytes in the STU header will then be 1 1 1 1 1 0 1 1 0 0 0 1 0 0 0 0.

PDU Size
This set of fields provides the size of each of the PDUs in the STU in bytes. As long as the STU headers are received correctly, the positions of all MAC PDUs in the slot can be correctly determined. The reported length of each PDU includes all headers and trailers.

CRC
The 16-bit CRC is computed over the entire STU Header 2, beginning with the PDU Count and ending at the size of the final PDU prior to the CRC. The computation of the CRC is described in appendix 2.

A.8.3 MAC Management PDU

Bit Field	Number of Bits	Comment
Type	4	Set to 2 for MAC Management
Request Flag	1	1 if slot requests are included, Bit 4
Interferer Flag	1	1 if a summary SOT is provided, Bit 5
Number RA slots	2	Bits 6 - 7
Number Slots	8	
Number Neighbors	8	
Demand	8	Transmitting node's demand
Total Demand	8	Demand summed over all one-hop neighbors and transmitting node
SOT/Request Vector	8·Ceil(Slots/8) or 8·Ceil(Slots/4)	Size depends on whether slot requests are included
Neighbors		
Neighbor ID #1	8	
Neighbor ID #2	8	
…		
Neighbor ID #N	8	
Status for Neighbor #1	8	Includes Neighbor type (bits 0-2), reporting flag (Bit 3), and LQ (Bits 4-7)
Status for Neighbor #2	8	
…	8	
Status for Neighbor #N	8	
Neighbor Reporting		
SOT for reported neighbour #1	8·Ceil(Slots/8)	
…		
SOT for reported neighbour #last	8·Ceil(Slots/8)	
Summary SOT for interferers	8·Ceil(Slots/8) or 0	Included if interferer flag = 1
CRC	16	

Type
The Type field is used to distinguish the DSRMA management packet from other MAC layer packets and is set to 2.

Request Flag
The request bit is set to 1 to indicate that this transmission includes slot requests and set to 0 if it does not. This determines whether a SOT or a SOT/Request Vector will be transmitted.

Interferer Flag
This bit is set to 1 if a summary SOT for interferers is provided at the end of the PDU and 0 if none is needed.

Number of RA Slots
This value reports the number of Random Acess Slots per cycle – 1 in binary. There can be 1, 2, 3, or 4 RA slots per cycle. At least one is required. A value of 0 denotes one RA slot.

Number of Slots
This reports the total number of slots per cycle – 1 in binary. A maximim of 256 slots per cycle can be supported.

Number of Neighbors
This is the number of neighbors for this node and can be 0.

Demand
This is the demand reported by this transmitter as described in section A.5.6 .

Neighborhood Demand
This is the sum of the demands reported by all one-hop neighbors plus the transmitting node.

SOT/Request Vector
If the request flag is set to 0, this is a bit vector with entries of 1 indicating slots used for transmission by this MARLIN node. The least significant bit of the first byte of this field corresponds to slot 0, the most significant bit of the first byte corresponds to slot 7, the first bit of the second byte of this field corresponds to slot 8 and so on. The field will be padded out to the byte boundary with zeros.
If the request flag is set to 1, then this vector includes slot request information as well as slot ownership information. The two least significant bits of the first byte of this field then correspond to slot 0, with values interpreted as:

0 = not owned and not requested,
1 = owned, i.e., reserved for transmission
2 = ordinary slot request,
3 = priority slot request.

The most significant two bits of the first byte correpsond to slot 3, the first two bits of the second byte correspond to slot 4, and so on. The bit vector is padded to the byte boundary with zeros.

Neighbor Identifiers
This indicates the IDs of the neighbors this node hears. The number of neighbors has been reported earlier in the PDU.

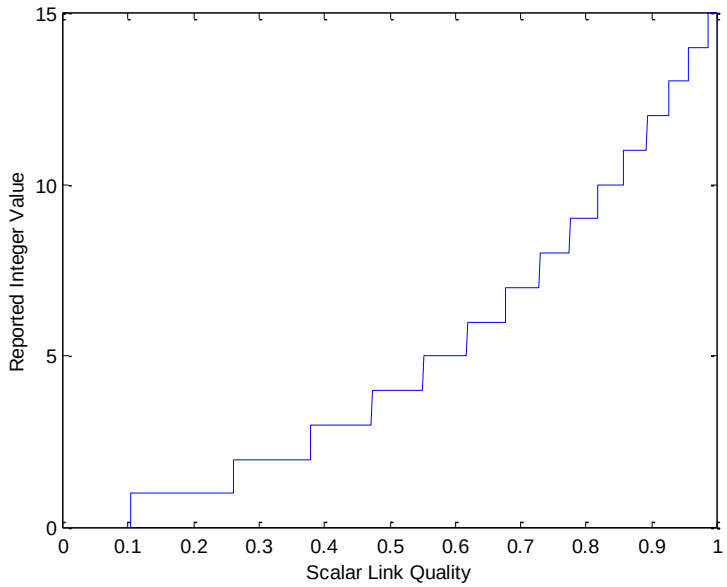
Neighbor Status

One byte is reported for each neighbor. Neighbors are reported in order in which the IDs appear in the list immediately preceding it. The first 3 bits, bits 0 – 2 are used to indicate neighbor status as described in section A.5.2 . The values employed are:

0 = Non-neighbor
1 = Bad neighbor
2 = Asymmetric (one-way) neighbor
3 = Symmetric (good, two-way) neighbor but not MLA
4 = MAC Layer Arbiter (MLA)

The next bit, bit 3, is used to indicate whether further reporting for this node will occur, i.e., this neighbor's demand value and SOT table. In normal operation when no priority requests are present, a node will report these entries for all neighbors in its MLA Selector set.

The final bits, bits 4-7 are used to report link quality (LQ) received from this neighbor. The Link Quality, as described earlier. Link Quality is a scalar between 0 and 1 (representing the probability of successful packet reception) and must be reported as 4-bit integer between 0 and 15. To give more granularity to the higher probabilities and therefore the region of interest, a non-linear mapping is employed, with the integer value reported being given by $2\cdot(17/2)^{LQ} - 2$, rounded to the nearest integer. The result of this operation is illustrated in the figure below.



Value	LQ	Value	LQ	Value	LQ	Value	LQ
15	(0.986, 1.0]	11	(0.856, 0.892]	7	(0.676, 0.728]	3	(0.379, 0.473]
14	(0.957, 0.986]	10	(0.817, 0.856]	6	(0.618, 0.676]	2	(0.261, 0.379]

13	(0.926, 0.957]	9	(0.775, 0.817]	5	(0.551, 0.618]	1	(0.104, 0.261]
12	(0.892, 0.926]	8	(0.728, 0.775]	4	(0.473, 0.551]	0	[0.0, 0.104]

SOT for Reported Neighbors
Each neighbor reported (for which bit 3 of the neighbor status is set to 1) has its SOT relayed. The format is the same as used previously. The least significant bit of the first byte of this field corresponds to slot 0, the most significant bit of the first byte corresponds to slot 7, the first bit of the second byte of this field corresponds to slot 8, and so on. An entry of 1 indicates the neighbor is using the slot for transmission, and a 0 indicates it is not. The field will be padded out to the byte boundary with zeros.

Summary Interference SOT
This SOT is included if the interferer flag is set previously and is omitted if not. If included, it reports the logical OR of the SOTs used by all interfering neighbors, i.e., bad neighbors and asymmetric neighbors. If there are no interfering neighbors, then there is nothing to report.

CRC
The DSRMA PDU is covered with a 16 bit cyclic redundancy check over the entire PDU, beginning with the type field. The first byte of the CRC contains the least significant byte of the CRC while the second byte contains the most significant byte of the CRC. The method for computing the CRC is provided in appendix 2.

A.8.4 MAC ACK Packets

Bit Field	Number of Bits	Comment
Type	4	Set to 3 for ACK
Cycle	4	
Slot	8	
Transmitter ID	8	
SACK Flag	1	
PDU Number	7 (15)	64 or more PDUs require extension
SACK bit-vector	8·Ceil(PDU/8)	If included
CRC	16	

Type
For the MAC Acknowledgment PDU, the type field is set to 3.

Cycle
This field indicates the number of the cycle being acknowledged.

Slot
This field indicates the slot being acknowledged.

Transmitter ID
This is the ID of the node that transmitted in the slot being acknowledged, i.e., the source of the data, not the node transmitting this ACK packet.

SACK Flag
The first bit of the next byte is a flag indicating whether this ACK packet is a cumulative ACK, indicated by setting the Flag = 0, or a selective ACK (SACK), Flag = 1.

PDU Number
The second bit is of the fourth byte in this packet is an extension bit. If this is set to 0, the remaining 6 bits of the fourth byte are used to indicate the PDU number, up to a maximum of 63. If the extension bit is set to 1, then the next 14 bits are used to indicate the PDU number, with the most significant bits coming from the 8 bits of the following byte and the least significant bits of this number coming from the final 6 bits of the first byte.

If the SACK flag is set to 0, then this is a cumulative ACK and the PDU number indicates the PDU number of the last PDU successfully received in the STU being acknowledged. In this case, the PDU number is the final entry in the ACK packet before the CRC. If the SACK flag is set to 1, then the PDU number is the PDU count of the STU. In this case, a SACK vector is the next and final entry in the ACK packet, before the CRC.

SACK Vector
This is a bit-vector indicating which PDUs in this STU were successfully received. An entry of 1 indicates successful reception of the PDU and an entry of 0 indicates unsuccessful reception.

The least significant bit of the first byte of this field corresponds to slot 0, the most significant bit of the first byte corresponds to slot 7, the first bit of the second byte of this field corresponds to slot 8, and so on. The field will be padded out to the byte boundary with zeros.

CRC
This is a 16-bit CRC over the entire ACK Packet, starting with the type field.

A.8.5 MAC Data Header

Field	Number of Bits	
Type	4	Set to 1 for Data
Reserved	3	Set to 0
Reliability Flag	1	
Next Hop ID	8	
DATA PAYLOAD	Variable	
CRC	16	

Type
This is set to 1 for MAC data.

Reliability Flag
The final bit of the first byte is set to 1 if reliable transmission, i.e., acknowledgment, of this PDU is requested. Otherwise, this is set to 0.

Next Hop ID
This is the ID of the next hop relay for this PDU with 255 for broadcast.

CRC
The DSRMA PDU is covered with a 16-bit cyclic redundancy check over the entire PDU, beginning with the type field, and including the payload. Recall that the length of this PDU, header and payload, is provided in the STU header – Part 2. The first byte of the CRC contains the least significant byte of the CRC while the second byte contains the most significant byte of the CRC. The method for computing the CRC is provided in appendix A-2.

A.8.6 LLC Header

Bit Field	Number of Bits	Comment
Frame Type	4	
Priority	4	
Source Address	8	
Destination Address	8	
PDU ID	16	
PDU Length	16	
Hop Count	5	
Compress Flag	1	
Fragment Flag	1	
Spare	1	Set to 1
Fragmentation Offset	8	

Frame Type
Two values are currently defined. RC control packets are encapsulated in LLC headers for delivery. All values must be relayed, even if not understood.

Data Type	Frame Type Value	Notes
IP Data	2	IP Traffic from Router Interface
Relay Control	3	RC Control Data
Vendor Specific	12-15	Vendor features
Not used	0, 1, 4 - 11	Reserved for future use

Priority
Sixteen priority levels are supported with 0 being lowest priority and 15 is highest priority. The Priority field specifies the queueing priority of data contained within the packet wrapped by the link-layer header. Packets are queued in order of priority, with packets being relayed given precedence over packets originating locally. Link-layer packets are all lower in priority than management packets. ARQ retransmissions, which are not specifically management PDUs, are queued ahead of any other link-layer generated PDUs.

Source Address
The source address is the 8 bit MARLIN Node ID of the source of this packet. Valid MARLIN Node IDs are in the range from 1 to 254. The ID 255 is used to indicate the network and the ID 0 is reserved to indicate PID non-reachable. The ID 255 is not a valid source address because the 255 destination address is reserved to indicate a broadcast.

Destination Address

The destination address is the 8 bit MARLIN Node ID of the final destination of this packet. Valid MARLIN Node IDs are in the range from 1 to 254. The IDs 0 and 255 are reserved.

PDU ID
The PDU ID is a 16 bit field which is used to identify duplicate PDUs. This 16 bit field is incremented by one for every link-layer packet created with 65,536 unique possible values before there is an ambiguity in the packet. The PDU ID does not change when the packet is relayed.

PDU Length
The PDU length field indicates the length of this LLC PDU including the header.

Hop Count
This is set by the transmitting node and decremented by 1 by each relay. If it reaches 0, the packet is dropped by the LLC relay.

Compress Flag
This flag is set to 1 to indicate the payload was compressed before fragmentation. When compression is employed, zlib compression as specified in Request For Comments (RFC) 1950 is used, with the original data passed to the zlib compress function. The zlib decompress function is used to recover the original data upon reception.

Fragment Flag
This flag is set to 1 to indicate that the source packet was fragmented and more packets are coming. The flag is set to 0 for the final fragment.

Fragmentation Offset
The fragmentation offset specifies the offset in units of 8 bytes for this particular fragment. For example, a value of 100 would indicate that the fragment offset was 800 bytes and that this portion of the data payload begins with the 800^{th} byte.

A.8.7 Relay Control PDU

Bit Field	Number of Bits	Notes
Originator ID	8	Node ID of the original message
Sequence Number	8	
Neighbor ID #1	8	
Link Cost #1	4	
Reserved	2	
Nieghbor Status #1	2	
Neighbor ID #2	8	
List Cost #2	4	
Reserved	2	
Neighbor Status #2	2	
…		
Neighbor ID #N	8	
Link Cost #N	4	
Reserved	2	
Neighbor Status #N	2	

Originator ID
This is the ID of the MARLIN node generating this RC Control Packet.

Sequence Number
This is incremented by 1 with each RC control PDU.

Neighbor IDs
These list the IDs of the neighbors being reported. The number of neighbors can be determined from the size of Relay Control PDU.

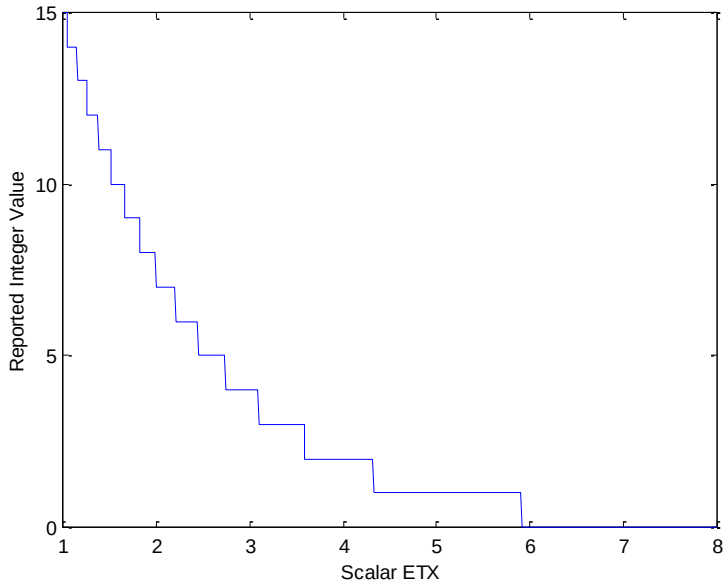
Link Costs
These fields report link costs for each of the neighbors reported above, in the order in which they appear in the previous list. The link cost reported is the expected transmission count (ETX) used in Optimized Link State Routing (OLSR) with link quality extensions. If LQ is the link quality from node j to node i and ILQ is the inverse link quality from i to j, reported by j, then the expected transmission count is

$$\mathrm{ETX} = \frac{1}{\mathrm{LQ} \times \mathrm{ILQ}}.$$

Node i can compute the LQ from j to i directly and can recover the ILQ from the link quality reported by j. The maximum possible inverse link quality should be assumed, so that, for

example an ILQ reported with integer 15 is interpreted as ILQ = 1.0, if 14 is reported, then i should interpret ILQ = 0.986, and so on. Refer to section A.6.1.

The computed ETX is reported as the 4-bit integer $y = 15 \cdot (1/2)^{ETX - 1}$, rounded to the nearest integer. The results of this computation are shown in the figure below. This non-linear transformation places greater discrimination in the range of greatest interest.



,

Value	ETX	Value	ETX	Value	ETX	Value	ETX
15	[1.0, 1.049)	11	[1.383, 1.515)	7	[2.0, 2.206)	3	[3.1, 3.585)
14	[1.049, 1.152)	10	[1.515, 1.659)	6	[2.206, 2.448)	2	[3.585, 4.322)
13	[1.152, 1.263)	9	[1.659, 1.819)	5	[2.448, 2.737)	1	[4.322, 5.907)
12	[1.263, 1.383)	8	[1.819, 2.0)	4	[2.737, 3.1)	0	[5.907, ∞)

When converting reported integers for ETX, the minimum possible ETX should be assumed. Thus, a value of 15 is interpreted as ETX = 1.0, a value of 14 is interpreted as 1.049, and so on. Refer to the table above.

Neighbor Status

Neighbor status indicates whether the neighbor being reported an MPR, MPRS, or neither.

Status = 0 Neighbor (but neither MPR nor MPRS)
Status = 1 MPR (but not MPRS)
Status = 2 MPRS (but not MPR)
Status = 3 MPR + MPRS

Appendix A.1: Expected Operational Requirements

A. As the size of a coalition battle group is typically limited, the number of platforms participating in a MARLIN network is assumed to be limited to no more than 16. The maximum network diameter is assumed to typically be no more than five hops. This provides almost 100 nm of coverage over UHF LOS, assuming a 25 nm UHF range, and considerably greater over HF.

B. The topologies are expected to change frequently and the MARLIN system will be expected to automatically reconfigure. Acceptable response times to the most dramatic topological changes (such as the addition of a new network participant or the merging of two partitions) are expected to be on the order of 2-3 minutes.

C. The number of simultaneous MARLIN subnets (i.e. on several bearers) per platform may be limited by spectrum availability or the number of legacy radios aboard. It is assumed that subnets will be limited to the use of one single-channel half-duplex radio with an omni-directional antenna. As the intent is to operate with the available legacy equipment, it is not assumed that spread spectrum and other interference avoidance mechanisms will be available at the physical layer.

D. Bulk encryption will be performed below the MARLIN controller. Encryption delays, as well as the delays associated with legacy voice radios (such as switching delays) may be significant, ruling out the use of fast handshakes in the MARLIN system implementation.

E. Burst rates of 2.4 kbps to 19.2 kbps are achievable over HF in 3 or 6 kHz channels and burst rates of 16 kbps to 64 kbps and higher are achievable over UHF in 25 kHz channels. Thus, MARLIN will be expected to support low and medium bandwidth applications, such as text chat, email, file transfer, whiteboarding, distribution of common operating and tactical picture (COTP), limited web browsing, and so on. Because of the delays inherent in the system, MARLIN will not be expected to carry real-time traffic such as Voice over IP.

F. A substantial portion of the data will be broadcast and multicast in nature.

Appendix A.2: 16-BIT Cyclic Redundancy Check

A CRC can be added to a packet to allow the recipient to confirm that the packet has not been corrupted in transmission. In this STANAG the same 16 bit CRC used in STANAG 5066 has been employed. The polynomial used to generate the CRC is

$$x^{16} + x^{15} + x^{12} + x^{11} + x^8 + x^6 + x^3 + 1.$$

The code fragment below employs this polynomial to set up a table used in the computation of the CRC.

```
 #define crcPoly 0x9299 //x^16 + x^15 + x^12 + x^11 + x^8 + x^6 + x^3 + 1
int i;
int j;
unsigned int Crc;

for (i = 0; i < 256; i++)
     {
          Crc = i;
          for (j = 8; j > 0; j--)
          {
           if (Crc & 1)
                Crc = (Crc >> 1) ^ crcPoly;
           else
               Crc = (Crc >> 1);
            }
               m_CrcTable16[i] = Crc;
          }
```

Once the table has been computed, the CRC for a particular block of data can be computed. In the code fragment below, pBuffer is a data buffer of bytes for which a CRC is to be computed over a length given by DataLength. The code fragment below places the resulting 16 bit CRC in the variable Crc.

```
unsigned int Crc = 0;
unsigned int Temp1;
unsigned int Temp2;

for ( i = 0; i < DataLength; i++)
     {
      Temp1 = (Crc >> 8) & 0xFF;
      Temp2 = m_CrcTable16[ ( Crc ^ pBuffer[i] ) & 0xFF ];
      Crc = Temp;
     }
```

Appendix A.3: Random Access Slot Position

The C-code fragment below is used to generate RA slot numbers for up to NumRaSlots. RaSlot[0] should be discarded and RaSlot[1] is the first valid RA slot number. If there are 4 RA slots, they would be given by RaSlot[1], RaSlot[2], RaSlot[3] and RaSlot[4] after making the calculation below with the node's Partition ID (PID) and the number of slots per cycle.

```
Random = PID
for( j = 0 ; j <=  NUMBER_OF_RA_SLOTS ; j++ )
    {
    Random = Random * 1103515245 + 12345;
    RaSlot[j] = (unsigned int)(Random/65536)%SLOTS_PER_CYCLE;
    for( k = 1 ; k < j ; k++ )
            {
            if(RaSlot[j] == RaSlot[k])
            //This is a duplicate.  Rewind and try again.
                    j--;
            }
    }
```

Appendix A.4: Abbreviations and Acronyms

This appendix defines the abbreviations and acronyms that appear in the document.

ACK	Acknowledgment
ARP	Address Resolution Protocol
ARQ	Automatic Repeat Request
CRC	Cyclic Redundancy Check
DA	Demand Assigned
DCE	Data Communications Equipment
DSRMA	Distributed Slot Reservation Media Access
ELOS	Extended Line of Sight
ETX	Expected Transmission Count
FA	Fixed Allocation
GPS	Global Positioning System
HF	High Frequency
ID	Identifier
ILQ	Inverse Link Quality
IP	Internet Protocol
LAN	Local Area Network
LLC	Link Layer Control
LOS	Line of Sight
LQ	Link Quality
LS	Link State
MAC	Media Access Control
MARLIN	Mobile Ad Hoc Relay Line of Sight Networking
MLA	MAC Layer Arbiter
MPR	Multi-Point Relay
MPRS	Multi-Point Relay Selector
MSC	Missed Slot Count
NTP	Network Time Protocol
OLSR	Optimized Link State Routing
OSI	Open System Interconnect
OSPF	Open Shortest Path First
PDU	Protocol Data Unit
PI	Physical Interface
PID	Partition Identifier
RA	Random Access
RC	Relay Control
RI	Router Interface
SACK	Selective Acknowledgment
SATCOM	Satellite Communications
SOT	Slot Ownership Table
STU	Slot Transmission Unit
TCP	Transport Control Protocol
TDMA	Time Division Media Access
UDP	User Datagram Protocol
UHF	Ultra-High Frequency
UTC	Coordinated Universal Time
UTZ	UNIX Time Zero
VHF	Very High Frequency

Appendix A.5: Typical Parameter Values

In order for MARLIN nodes to be interoperable, the following configuration parameters must be configurable and consistent across the network. Suggested values are supplied where applicable.

Slot Length (T = 380 ms): Every node in the MARLIN network must be configured to transmit within the bounds of the same slot length. Slots are calculated as per section A.5.3 . This slot length is advertised in STU Header 1.

Number of Fixed Allocation Slots/Cycle (M_{FA} = 1): Every node in the MARLIN network is configured to have at least 1 M_{FA} slot per cycle. This number is local to a node, and does not need to be consistent across a network. See section A.5.5

Number of Slots/Cycle (M = 32): Every node in the MARLIN network must be configured to the same number of slots in a cycle. Slots and Cycles are calculated as per section A.5.3 . This number is advertised in the DSRMA Management PDU.

Number of RA Slots / Cycle (M_{RA} = 2): Each node in the MARLIN network must be configured to use the same number of RA Slots per Cycle. RA Slot locations are calculated as per Appendix 3. This number is advertised in the DSRMA Management PDU.

Burst Rate (R = 64000kbps): Each node in the MARLIN network must be configured to transmit at a configurable data rate. The MARLIN node does not transmit this parameter explicitly.

Guard Time (T_g = 20): Each node in the MARLIN network must be configured to the maximum amount of Guard Time required by the most disadvantaged node in the network. This value is defined in section A.5.3

Transmitted Overhead (T_m, T_c): Each node should be configured to know the duration of time that is dedicated to external overhead in a slot. This should include Modem and Cryptographic preambles and/or trailers that may be present. It is not necessary that this be configured to the same value across the network. This value is defined in section A.5.3

Total Delay: Each node must be configured to know the worst case delay that can not be optimized. This is the sum of the transmit delay from the node with the longest transmit delay, the receive delay from the node with the longest receive delay, and the longest receive recovery or radio switching time.

Link Quality Parameters: Each node in the MARLIN network must evaluate and advertise Link Quality in the same fashion. Link Quality is calculated as per section A.5.9

, and advertised directly in the DSRMA Management PDU, and indirectly in the RC Control PDU as ETX. The parameters are:

- Initial Link Quality for new joiners ($\mathbf{LQ_0 = 0.7}$)
- Link Quality for Timed out Neighbors ($\mathbf{LQ_1 = 0.1}$)
- Smoothing Factor for Link Quality Estimation ($\beta = 0.7$)
- LQ Threshold from NOT_NEIGHBOR to BAD_NEIGHBOR ($\mathbf{L_{12} = 0.5}$)
- LQ Threshold from BAD_NEIGHBOR to NOT_NEIGHBOR ($\mathbf{L_{21} = 0.4}$)
- LQ Threshold from BAD_NEIGHBOR to GOOD_NEIGHBOR ($\mathbf{L_{23} = 0.7}$)
- LQ Threshold from GOOD_NEIGHBOR to BAD_NEIGHBOR ($\mathbf{L_{32}=0.5}$)

These parameters should be the same across the network. These parameters are not advertised.

Relay Control Parameters: Each node in the MARLIN network must advertise its Local Topology as defined in Section A.4. Each node will be configured with the same RC Reporting Interval ($\mathbf{T_{RI} = M \times T}$) and RC Hold timer ($\mathbf{T_{RH} = 3\ M \times T}$) values. These parameters are not advertised.

Arrival Rate Smoothing Factor ($\alpha = (0.5)^{1/M \times T}$): Each node in the MARLIN network must be configured to use the same demand function as defined in section 3.6. The result of the demand function is advertised in the DSRMA Management PDU, but the smoothing factor used in this computation α is not advertised.

PID Probation Timeout ($T_{PID} = T_{RH}$): Each node in the MARLIN network must be configured to hold on to a PID for a probationary period of time, as defined in section A.5.8 . This number is not advertised.

Number of ARQ retransmission Attempts: Each node in the MARLIN network must make at least one attempt to retransmit reliable data as defined in section A.6. This number does not need to be consistent across the network.

INTENTIONALLY BLANK

ANNEX B (OPTIONAL) TECHNICAL SPECIFICATIONS TO ENSURE INTEROPERABILITY OF SERIAL WAVEFORMS FOR 25 KHZ BANDWIDTH LINE OF SIGHT VHF AND UHF RADIO DATA COMMUNICATIONS

B.1. Introduction

B.1.1 Purpose

This Annex provides a detailed description of modem waveforms to be used in 25 kHz bandwidths to ensure interoperability within complying Very High Frequency (VHF) and Ultra High Frequency (UHF) radio networks.

B.1.2 Approach and Structure of this Document

The waveforms described in STANAG 4691 are designed to satisfy NATO requirements.

This Annex specifies waveforms for general applications, including broadcast, Automatic Repeat reQuest (ARQ) and Time Division Multiple Access (TDMA) operation.

B.1.3 Overview

This 96 kbps waveform was designed using the same approach as had been taken in the high data rate HF modem standard STANAG 4539. This is the most appropriate comparable standard even though it is intended for use with HF channels. Like STANAG 4539, this standard proposes modem waveforms which are intended to cope with some degree of multipath delay spread in a fading channel environment while achieving a very high bit/s/Hz through the use of QAM modulations. As in STANAG 4539, the waveform includes several self-identifying data rates and interleaver settings, with lower data rates providing more robustness with respect to both noise and channel impairments.

This waveform was intended to be primarily used with legacy radios that provide a wideband audio interface (~25 kHz). This interface is widely available in radios that support external secure voice. The wideband AM interface of the WSC-3 has been used extensively with a very similar waveform with a lower baud rate. This interface has been shown to support the 64 QAM constellations and has provided good performance both in the lab and in at-sea trials.

The user interfaces section of this STANAG have followed that of STANAG 4539, with one minor change. In STANAG 4539, the provision of an Ethernet interface is mandatory, whereas in this STANAG it has been specified as optional. The reason for this is that it is difficult to justify making an interface mandatory when there is insufficient detail in the specification to ensure interoperability using that interface.

This section presents a modem waveform and coding specification for data rates of 16, 32, 48, 64, 80 and 96 kbps operating within audio channel bandwidths of 25 kHz. This

self-identifying waveform family is required for 16-96 kbps operation for systems complying with this STANAG.
A block interleaver is used to obtain 4 interleaving lengths 20 ms, 80 ms, 320 ms and 1.28 s. A rate ½, a constraint length 9, convolutional code is used. For data rates of 96, 80 and 64 kbps, the code is punctured to rate $^{8}/_{9}$. For other rates the code is punctured to rate ¾ or rate ⅔. A full-tail-biting approach, with the block size equal to the interleaver length, is used.

The waveform uses Phase Shift Keying (PSK) and Quadrature Amplitude Modulation (QAM).

The data rate and interleaver settings are explicitly transmitted as a part of the waveform, both as part of the initial preamble and then periodically as both a reinserted preamble and in the periodic known symbol blocks. The receive modem is required to be able to deduce the data rate and interleaver setting either from the preamble or from the reinserted preamble portion of the waveform. A summary of the waveform characteristics is provided in tabular form in Table B.1.3 -1 below.

Table B.1.3 -1: Summary of Waveform Characteristics

Data Rate	Modulation	Block Size		Code Rate	Interleaver Settings (ms)	Reinserted Preamble Size (symbols)
		Data	Training			
16	QPSK	80	19	½	20, 80, 320, 1.280	64
32	8PSK			⅔		
48	16QAM			¾		
64	16QAM	180	19	$^{8}/_{9}$		
80	32QAM			$^{8}/_{9}$		
96	64QAM			$^{8}/_{9}$		

Both the data rate and interleaver settings are explicitly transmitted as a part of the waveform, both as part of the initial preamble and then periodically as both a reinserted preamble and in the periodic known symbol blocks. The receive modem is required to be able to deduce the data rate and interleaver setting either from the preamble or from the reinserted preamble portion of the waveform.

B.2. Modulation

The symbol rate for all symbols shall be 20000 symbols-per-second, which shall be accurate to a minimum of ±0.2 symbols-per-second (10 ppm) when the transmit data clock is generated by the modem and not provided by the data terminal equipment (DTE). Phase-shift keying (PSK) and quadrature amplitude modulation (QAM) modulation techniques shall be used. The sub-carrier (or pair of quadrature sub-carriers in the case of QAM) shall be centered at f_c accurate to a minimum of 10 ppm. The phase of the Quadrature sub-carrier relative to the In-phase carrier shall be 90 degrees. The correct relationship can be achieved by making the In-phase sub-carrier cos(f_c) and the Quadrature sub-carrier -sin(f_c).

When the modem signal is to be used in conjunction with the wideband audio interface of a V/UHF radio, f_c in the paragraph above shall be 12.5 kHz.

The power spectral density of the modulator output signal should be constrained to be at least 20 dB below the signal level measured at f_c, when tested outside of the band from f_c-12.5 kHz to f_c +12.5 kHz. The measured ripple shall be no more than ±2 dB in the range from f_c-9 kHz to f_c +9 kHz.

Table B.2-1: Modulator Output Signal Levels

Frequency Range	Modulator Output Power Spectral Density (PSD) (dB) Relative to PSD at f_c
$f = f_c$	0
$f < f_c$-12.5 kHz	< -20 dB
f_c -12.5 kHz < f < f_c + 12.5 kHz	+/- 2 dB
$f > f_c$+12.5 kHz	< -20 dB

The pulse shaping filter which shall be employed is a root-raised cosine filter, specified by the T/4 (80 kHz sampling rate) spaced coefficients of the finite impulse response (FIR) filter:

-0.01829642433379
-0.04269314885140
-0.03751317983988
 0.00602173849178
 0.06529614414503
 0.09410311027815
 0.05305164769730
-0.05499532418853
-0.17029763384600

-0.19871737022884
-0.06423715577700
0.23786182931496
0.62179741050913
0.94316532074438
1.06830988618379
0.94316532074438
0.62179741050913
0.23786182931496
-0.06423715577700
-0.19871737022884
-0.17029763384600
-0.05499532418853
0.05305164769730
0.09410311027815
0.06529614414503
0.00602173849178
-0.03751317983988
-0.04269314885140

B.2.1 Known Symbols

For all known symbols, which are defined as those symbols which are known to the receiver prior to reception, the modulation used shall be PSK, with the symbol mapping shown in Table B.2.1 -1 and Figure B.2.1 -1. No scrambling shall be applied to the known symbols.

Table B.2.1 -1: 8PSK Symbol Mapping

Symbol Number	Phase	In-Phase	Quadrature
0	0	1.000000	0.000000
1	$\pi/4$	0.707107	0.707107
2	$\pi/2$	0.000000	1.000000
3	$3\pi/4$	-0.707107	0.707107
4	π	-1.000000	0.000000
5	$5\pi/4$	-0.707107	-0.707107
6	$3\pi/2$	0.0000000	-1.000000
7	$7\pi/4$	0.707107	-0.707107

Note that the complex symbol values = $\exp[jn\pi/4]$ where n is the symbol number.

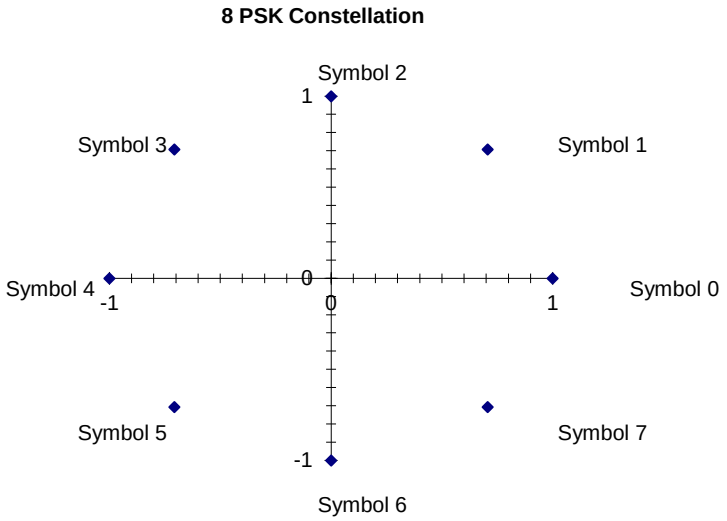


Figure B.2.1 -1: 8PSK Signal Constellation and Symbol Mapping

B.2.2 Data Symbols

For data symbols, the modulation used shall depend upon the data rate. Table B.2.2 -1 specifies the modulation that shall be used with each data rate.

Table B.2.2 -1: Modulation Used to Obtain Each Data Rate

Data Rate (kbps)	Modulation
16	QPSK
32	8PSK
48	16QAM
64	16QAM
80	32QAM
96	64QAM

All PSK constellations are scrambled to appear, on-air, as an 8PSK constellation. The QAM constellations have been modified from conventional rectangular constellations to improve the peak-to-average ratio.

B.2.2.1 PSK Data Symbols

For the PSK constellations, a distinction is made between the data bits and the symbol number for the purposes of scrambling QPSK modulation to appear as 8PSK, on-air. Scrambling is applied as a modulo 8 addition of a scrambling sequence to the 8PSK

symbol number. Transcoding is an operation which links a symbol to be transmitted to a group of data bits.

B.2.2.1.1 QPSK Symbol Mapping

For the 16 kbps user data rate, transcoding shall be achieved by linking one of the symbols specified in Table B.2.1 -1 to a set of two consecutive data bits (dibit) as shown in Table B.2.2.1.1 -1. In this table, the leftmost bit of the dibit shall be the older bit; i.e., fetched from the interleaver before the rightmost bit.

Table B.2.2.1.1 -1: Transcoding for 16 kbps

Dibit	Symbol
00	0
01	2
11	4
10	6

B.2.2.1.2 8PSK Symbol Mapping

For the 32 kbps user data rate, transcoding shall be achieved by linking one symbol to a set of three consecutive data bits (tribit) as shown in Table B.2.2.1.2 -1. In this table, the leftmost bit of the tribit shall be the oldest bit; i.e., fetched from the interleaver before the other two, and the rightmost bit is the most recent bit.

Table B.2.2.1.2 -1: Transcoding for 32 kbps

Tribit	Symbol
000	1
001	0
010	2
011	3
100	6
101	7
110	5
111	4

B.2.2.1.3 QAM Data Symbols

For the QAM constellations, no distinction is made between the number formed directly from the data bits and the symbol number. Each set of 4 bits (16QAM), 5 bits (32QAM) or 6 bits (64QAM) is mapped directly to a QAM symbol. For example, the four bit grouping 0111 would map to symbol 7 in the 16QAM constellation while the 6 bit grouping 100011 would map to symbol 35 in the 64QAM constellation. Again, in each case the leftmost bit

shall be the oldest bit, i.e. fetched from the interleaver before the other bits, and the rightmost bit is the most recent bit.

The mapping of bits to symbols for the QAM constellations has been selected to minimise the number of bit errors incurred when errors involve adjacent signalling points in the constellation.

B.2.2.1.4 The 16QAM Constellation

The constellation points which shall be used for 16QAM are shown in Figure B.2.2.1.4 -1 and specified in terms of their In-phase and Quadrature components in Table B.2.2.1.4 -1.

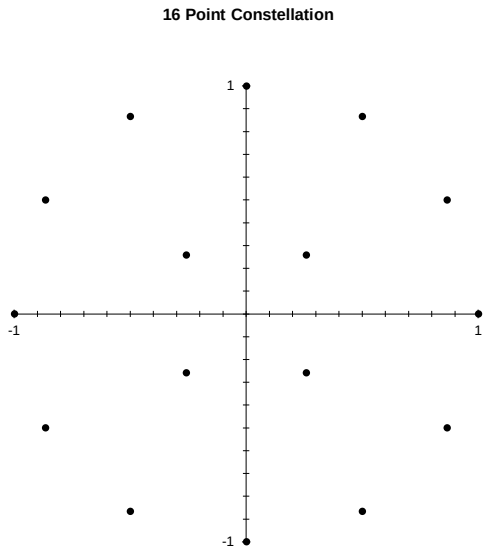


Figure B.2.2.1.4 -1: 16QAM Signalling Constellation

Table B.2.2.1.4 -1: In-phase and Quadrature Components of Each 16QAM Symbol

Symbol Number	In-Phase	Quadrature
0	0.866025	0.500000
1	0.500000	0.866025
2	1.000000	0.000000
3	0.258819	0.258819
4	-0.500000	0.866025
5	0.000000	1.000000
6	-0.866025	0.500000
7	-0.258819	0.258819
8	0.500000	-0.866025
9	0.000000	-1.000000
10	0.866025	-0.500000
11	0.258819	-0.258819
12	-0.866025	-0.500000
13	-0.500000	-0.866025
14	-1.000000	0.000000
15	-0.258819	-0.258819

B.2.2.1.5 The 32QAM Constellation

The constellation points which shall be used for 32QAM are shown in Figure B.2.2.1.5 -1 and specified in terms of their In-phase and Quadrature components in Table B.2.2.1.5 -1.

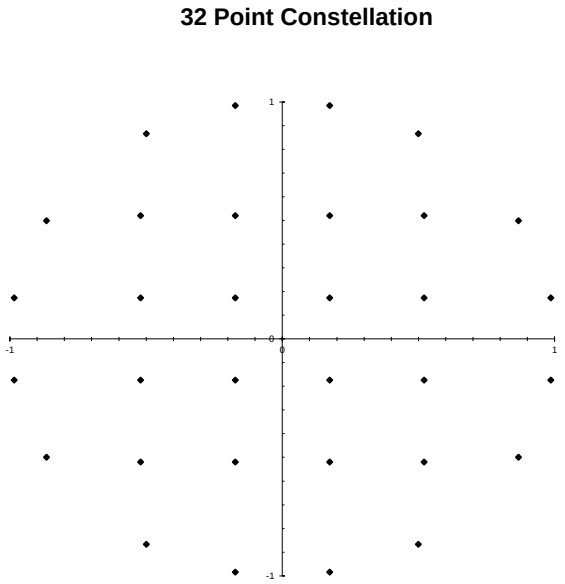


Figure B.2.2.1.5 -1: 32QAM Signalling Constellation

Table B.2.2.1.5 -1: In-phase and Quadrature Components of Each 32QAM Symbol

Symbol Number	In-Phase	Quadrature	Symbol Number	In-Phase	Quadrature
0	0.866380	0.499386	16	0.866380	-0.499386
1	0.984849	0.173415	17	0.984849	-0.173415
2	0.499386	0.866380	18	0.499386	-0.866380
3	0.173415	0.984849	19	0.173415	-0.984849
4	0.520246	0.520246	20	0.520246	-0.520246
5	0.520246	0.173415	21	0.520246	-0.173415
6	0.173415	0.520246	22	0.173415	-0.520246
7	0.173415	0.173415	23	0.173415	-0.173415
8	-0.866380	0.499386	24	-0.866380	-0.499386
9	-0.984849	0.173415	25	-0.984849	-0.173415
10	-0.499386	0.866380	26	-0.499386	-0.866380
11	-0.173415	0.984849	27	-0.173415	-0.984849
12	-0.520246	0.520246	28	-0.520246	-0.520246
13	-0.520246	0.173415	29	-0.520246	-0.173415
14	-0.173415	0.520246	30	-0.173415	-0.520246
15	-0.173415	0.173415	31	-0.173415	-0.173415

B.2.2.1.6 The 64QAM Constellation

The constellation points which shall be used for the 64QAM modulation are shown in Figure B.2.2.1.6 -1 and specified in terms of their In-phase and Quadrature components in Table B.2.2.1.6 -1.

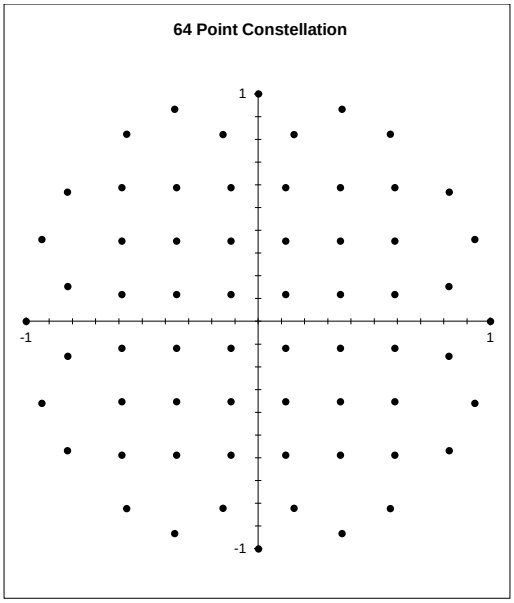


Figure B.2.2.1.6 -1: 64QAM Signalling Constellation

Table B.2.2.1.6 -1: In-phase and Quadrature Components of Each 64QAM Symbol

Symbol Number	In-Phase	Quadrature	Symbol Number	In-Phase	Quadrature
0	1.000000	0.000000	32	0.000000	1.000000
1	0.822878	0.568218	33	-0.822878	0.568218
2	0.821137	0.152996	34	-0.821137	0.152996
3	0.932897	0.360142	35	-0.932897	0.360142
4	0.000000	-1.000000	36	-1.000000	0.000000
5	0.822878	-0.568218	37	-0.822878	-0.568218
6	0.821137	-0.152996	38	-0.821137	-0.152996
7	0.932897	-0.360142	39	-0.932897	-0.360142
8	0.568218	0.822878	40	-0.568218	0.822878
9	0.588429	0.588429	41	-0.588429	0.588429
10	0.588429	0.117686	42	-0.588429	0.117686
11	0.588429	0.353057	43	-0.588429	0.353057
12	0.568218	-0.822878	44	-0.568218	-0.822878
13	0.588429	-0.588429	45	-0.588429	-0.588429
14	0.588429	-0.117686	46	-0.588429	-0.117686
15	0.588429	-0.353057	47	-0.588429	-0.353057
16	0.152996	0.821137	48	-0.152996	0.821137
17	0.117686	0.588429	49	-0.117686	0.588429
18	0.117686	0.117686	50	-0.117686	0.117686
19	0.117686	0.353057	51	-0.117686	0.353057
20	0.152996	-0.821137	52	-0.152996	-0.821137
21	0.117686	-0.588429	53	-0.117686	-0.588429
22	0.117686	-0.117686	54	-0.117686	-0.117686
23	0.117686	-0.353057	55	-0.117686	-0.353057
24	0.360142	0.932897	56	-0.360142	0.932897
25	0.353057	0.588429	57	-0.353057	0.588429
26	0.353057	0.117686	58	-0.353057	0.117686
27	0.353057	0.353057	59	-0.353057	0.353057
28	0.360142	-0.932897	60	-0.360142	-0.932897
29	0.353057	-0.588429	61	-0.353057	-0.588429
30	0.353057	-0.117686	62	-0.353057	-0.117686
31	0.353057	-0.353057	63	-0.353057	-0.353057

B.2.3 Data Scrambling

Data symbols for the 8PSK symbol constellation (16 kbps, 32 kbps) shall be scrambled by modulo 8 addition with a scrambling sequence. The data symbols for the 16QAM, 32QAM, and 64QAM constellations shall be scrambled by using an exclusive or (XOR) operation. Sequentially, the data bits forming each symbol (4 for 16QAM, 5 for 32QAM, and 6 for 64QAM) shall be XOR'd with an equal number of bits from the scrambling

sequence. The scrambling sequence generator polynomial shall be $x^9 + x^4 + 1$ and the generator shall be initialised to 1 at the start of: every second data frame for data rates 64, 80 and 96 kbps (which use a 180 symbol data block) and every fourth frame for data rates of 16, 32 and 48 kbps (which use an 80 symbol data block). In each case, the generator is initialized to 1 beginning with the first data frame following the preamble. A block diagram of the scrambling sequence generator is shown in Figure B.2.3 -1.

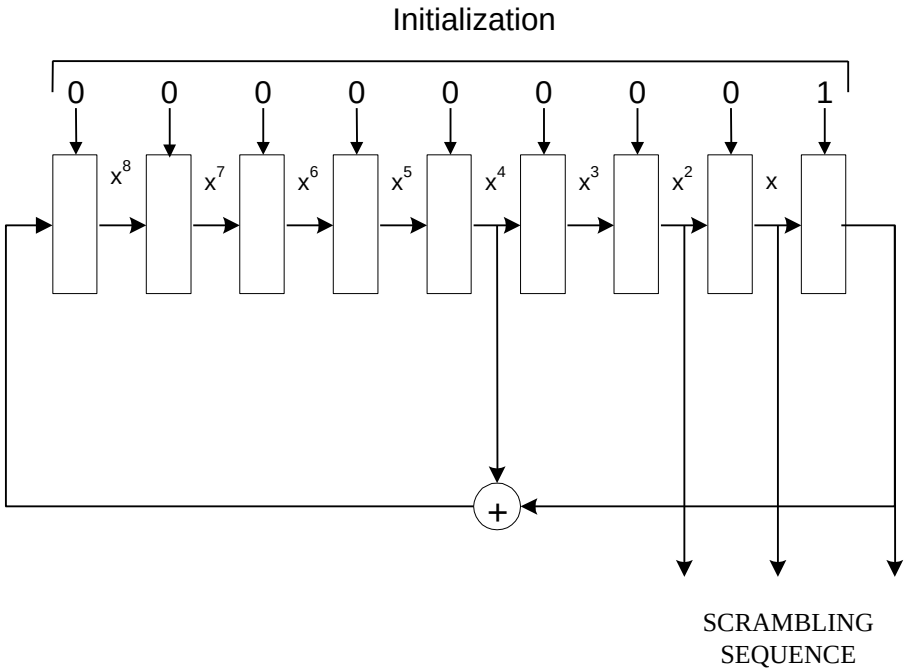


Figure B.2.3 -1: Scrambling Sequence Generator Illustrating Scrambling Generator for 8PSK Symbols

For 8PSK symbols (16 and 32 kbps), the scrambling shall be carried out taking the modulo 8 sum of the numerical value of the binary triplet consisting of the last (rightmost) three bits in the shift register, and the symbol number (transcoded value). For example, if the last three bits in the scrambling sequence shift register were 010 which has a numerical value equal 2, and the symbol number before scrambling was 6, symbol 0 would be transmitted since: (6+2) Modulo 8 = 0. For 16QAM symbols, scrambling shall be carried out by XORing the 4 bit number consisting of the last (rightmost) four bits in the shift register with the symbol number. For example, if the last 4 bits in the scrambling sequence shift register were 0101 and the 16QAM symbol number before scrambling was 3 (i.e. 0011), symbol 6 (0110) would be transmitted. For 32QAM symbols, scrambling shall be carried out by XORing the 5 bit number formed by the last (rightmost) five bits in the shift

register with the symbol number. For 64QAM symbols, scrambling shall be carried out by XORing the 6 bit number formed by the last (rightmost) six bits in the shift register with the symbol number.

After each data symbol is scrambled, the generator shall be iterated (shifted) the required number of times to produce all new bits for use in scrambling the next symbol (i.e., 3 iterations for 8PSK, 4 iterations for 16QAM, 5 iterations for 32QAM and 6 iterations for 64QAM). Since the generator is iterated after the bits are used, the first data symbol of every data frame shall, therefore, be scrambled by the appropriate number of bits from the initialisation value of 00000001.

The length of the scrambling sequence is 511 bits. For a 180 symbol data block with 6 bits per symbol, this means that the scrambling sequence will be repeated just slightly more than 2 times.

B.3. Frame structure

The frame structure that shall be used for the waveforms specified in this section is shown in Figure B.3-1. An initial 400 symbol preamble is followed by 64 frames of alternating data and known symbols. Each data frame shall consist of a data block followed by a mini-probe consisting of 19 symbols of known data. The data block shall consist of 180 data symbols for data rates of 96, 80, or 64 kbps, and 80 data symbols for data rates of 48, 32, or 16 kbps. After 64 data frames, a 64 symbol subset of the initial preamble is reinserted to facilitate late acquisition, Doppler shift removal, and sync adjustment. It should be noted that the total length of known data in this segment is actually 83 symbols: the 64 reinserted preamble symbols plus the preceding 19 symbol mini-probe segment which follows the last data block. Reinserted preambles will occur twice as often for data rates of 16-48 kbps as they do for data rates of 64-96 kbps because of the smaller size of the data block.

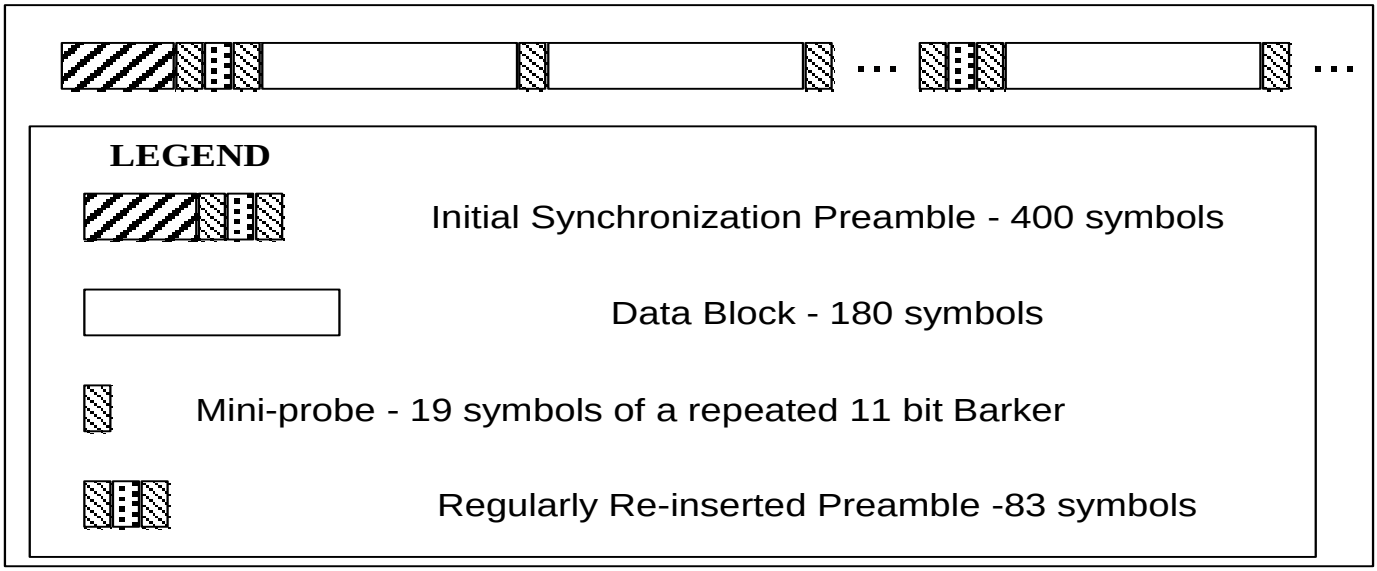


Figure B.3-1: Frame Structure for Data Rates of 64, 80, and 96 kbps

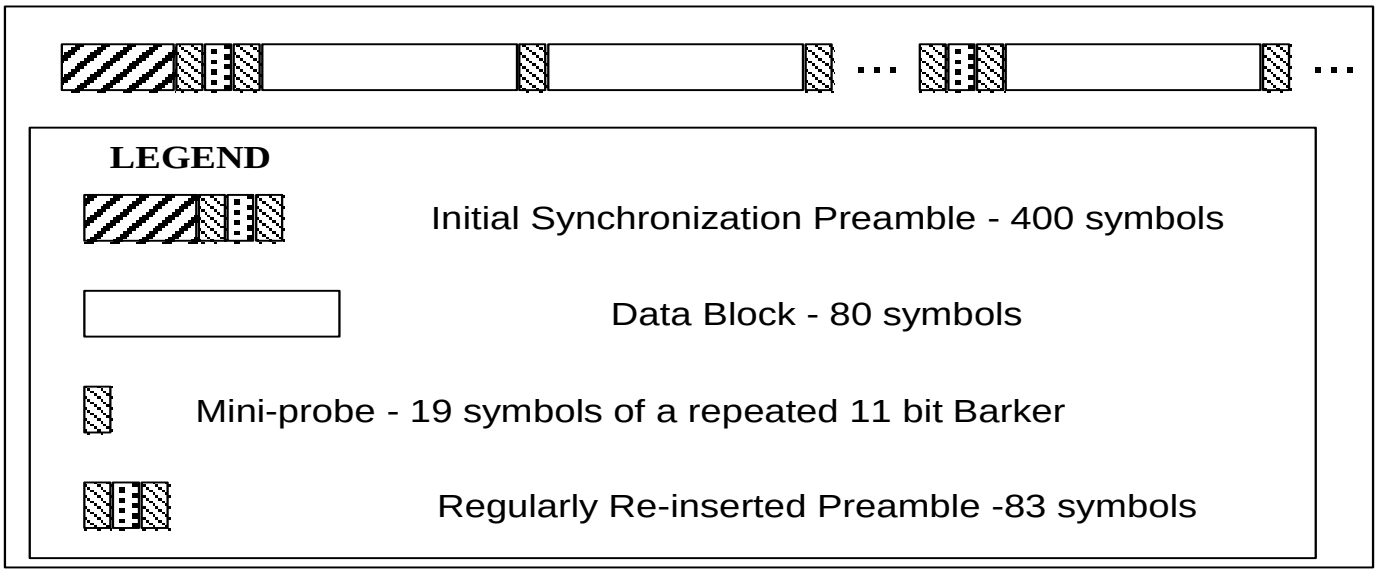


Figure B.3-2: Frame Structure for Data Rates of 16, 32, and 48 kbps

B.3.1 Synchronisation and Reinserted Preambles

The synchronisation preamble is used for rapid initial synchronisation. The reinserted preamble is used to facilitate acquisition of an ongoing transmission (acquisition on data).

B.3.1.1 Synchronisation Preamble

The mandatory portion of the preamble, which shall always be transmitted, and shall always appear at the beginning of an interleaver boundary, shall consist of 400 symbols.

The first 317 symbols are intended exclusively for synchronisation and Doppler offset removal purposes while the final 83 symbols, which are common with the reinserted preamble, also carry information regarding the data rate and interleaver settings. Expressed as a sequence of 8PSK symbols, using the symbol numbers given in Table B.2.1 -1 the second section of the synchronisation preamble shall be as follows:

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

4, 0, 4, 4, 0, 4, 4, 4, 0, 0, 0,
4, 0, 4, 4, 0, 4, 4, 4,

(D0, D0, D0, D0, D0, D0, D0, D0, D0, D0, D0 + 4, 0, 4, 4, 0, 4, 4, 4, 0, 0, 0) Modulo 8
(D1, D1, D1, D1, D1, D1, D1, D1, D1, D1, D1 + 4, 0, 4, 4, 0, 4, 4, 4, 0, 0, 0) Modulo 8
(D2, D2, D2, D2, D2, D2, D2, D2, D2, D2, D2 + 4, 0, 4, 4, 0, 4, 4, 4, 0, 0, 0) Modulo 8
(D3, D3, D3, D3, D3, D3, D3, D3, D3, D3, D3 + 4, 0, 4, 4, 0, 4, 4, 4, 0, 0, 0) Modulo 8

4, 4, 4, 0, 0, 0,
4, 0, 4, 4, 0, 4, 4, 4, 0, 0, 0,
4, 0, 4.

…where the data symbols D_0, D_1, D_2, and D_3 take one of the sets of values chosen from Table B.3.1.1 -1 to indicate the data rate and interleaver settings. The Modulo operations are meant to signify that each of the D values are used to shift the phase of a length 11 bit Barker code (01001000111) by performing modulo 8 addition of the D value with each of the Barker code 11 phase values (0 or 4). This operation can encode 8 bits of information using QPSK modulation of the 11 bit (chip) Barker codes. Since the four Barker code sequences occupy 44 symbols, the 19 symbol mini-probe following the

encoded Barkers is lengthened to 20 symbols to provide an additional symbol to pad the four 11 symbol Barker codes up to a total of 45 symbols.

Table B.3.1.1 -1: D0, D1, D2 8 PSK Symbol Values as a Function of Data Rate

Data Rate (kbps)	Interleaver Length			
	Ultra Short 20 ms	Short 80 ms	Medium 320 ms	Long 1.280 s
16	2,2,0	2,2,2	2,2,6	2,2,4
32	2,4,0	2,4,2	2,4,6	2,4,4
48	6,2,0	6,2,2	6,2,6	6,2,4
64	6,4,0	6,4,2	6,4,6	6,4,4
80	4,2,0	4,2,2	4,2,6	4,2,4
96	4,4,0	4,4,2	4,4,6	4,4,4

The synchronisation preamble may consist of two parts. The non-mandatory portion, which shall be transmitted before the mandatory portion described above if it is present, will contain a count-down and shall consist of N blocks of 400 8PSK symbols. The value of N shall be configurable to range from values of 0 to 15 (for N=0 this first section is not sent at all). When present, the 400 symbols of the non-mandatory, count-down section of the preamble shall be identical to those sent in the mandatory preamble with the exception of the values of the data symbol D_3 . The mapping chosen to create Table B.3.1.1 -1 used 3 bits to specify the data rate and two bits to specify the interleaver length. The phase of the Barker code was determined from 3 dibit words using Table B.2.2.1.1 -1, the dibit transcoding table, to transcode bit values into symbols. Numbering the bits from b_0 to b_7, with D_0 formed from the transcoding of b_7b_6, D_1 formed from the transcoding of b_5b_4, D_2 formed from the transcoding of b_3b_2, and D_3 formed from the transcoding of b_1b_0, it can be seen that for the mapping shown in Table B.3.1.1 -1, $b_7b_6b_5$ defines the data rate and b_3b_2 defines the interleaver setting. The second bit of the dibit forming D_1, b_4, is always one. This provides expansion capabilities for the future. D_3, is used to specify a count down as shown in Table B.3.1.1 -2.

Table B.3.1.1 -2: D0, D1, D2 8PSK Symbol Values Specifying Count of Preambles Before the Start of an Interleaver

Number of Preambles Before Interleaver Boundary	D3

>3	4
3	6
2	2
1	0

Not all preamble segments appear at the beginning of an interleaver boundary. There are three instances where this occurs: 1) during repeated preamble segments at the beginning of a transmission; 2) the single preamble inserted in the middle of the Long interleaver block for data rates of 64-96 kbps; and 3) the three preambles inserted in the long interleaver for rates of 16-48 kbps. The symbol value of the D_3 preamble segment immediately adjacent to the first data frame of an interleaver will be 0, indicating that this is the last preamble segment before an interleaver boundary. For those cases where the preamble is not adjacent to the interleaver boundary, D_3 encodes a count of the number of preambles prior to the interleaver boundary.

B.3.1.2 Reinserted Preamble

The reinserted preamble shall be identical to the final 64 symbols of the synchronisation preamble whenever it occurs immediately prior to the beginning of an interleaver block. In fact, the final 83 symbols are common between the synchronisation preamble and the contiguous block consisting of the reinserted preamble and the mini-probe which immediately precedes it. The 83 symbols of known data (including the 19 mini-probe symbols of the preceding data frame) are thus:

4, 0, 4, 4, 0, 4, 4, 4, 0, 0, 0,
4, 0, 4, 4, 0, 4, 4, 4,

(D_0, D_0, D_0, D_0, D_0, D_0, D_0, D_0, D_0, D_0, D_0 + 4, 0, 4, 4, 0, 4, 4, 4, 0, 0, 0) Modulo 8
(D_1, D_1, D_1, D_1, D_1, D_1, D_1, D_1, D_1, D_1, D_1 + 4, 0, 4, 4, 0, 4, 4, 4, 0, 0, 0) Modulo 8
(D_2, D_2, D_2, D_2, D_2, D_2, D_2, D_2, D_2, D_2, D_2 + 4, 0, 4, 4, 0, 4, 4, 4, 0, 0, 0) Modulo 8
(D_3, D_3, D_3, D_3, D_3, D_3, D_3, D_3, D_3, D_3, D_3 + 4, 0, 4, 4, 0, 4, 4, 4, 0, 0, 0) Modulo 8

4, 4, 4, 0, 0, 0,
4, 0, 4, 4, 0, 4, 4, 4, 0, 0, 0,
4, 0, 4.

…where in the case of a reinserted preamble at the beginning of an interleaver, the data symbols D_0, D_1 and D_2 again take one of the sets of values chosen from Table B.3.1.1 -1 to indicate the data rate and interleaver settings as described in the Synchronisation Preamble section above. In the case of a reinserted preamble which is not at the beginning of an interleaver, such as the preamble which will occur during the middle of the Long interleaver at 64-96 kbps, or one of the three preambles which occurs during the Long

interleaver for data rates of 16-48 kbps, the count of preambles remaining until the next interleaver begins is encoded into the data symbol D_3 as described in Table B.3.1.1 -2. Note that in every case, the first 19 of these symbols are the immediately preceding mini-probe, which follows the last of the data blocks.

B.3.2 Mini-Probes

Mini-probes 19 symbols in length shall be inserted following every data block and at the end of each preamble (where they are considered to be part of the preamble and a different cyclic shift of the sequence is used). Using the 8PSK symbol mapping, each mini-probe shall be based on an 11 bit Barker sequence. The sequence that shall be used, for all mini-probes other than that which is part of the preamble or reinserted preamble, specified in terms of the 8PSK symbol numbers, is given by:

4, 0, 4, 4, 0, 4, 4, 4, 0, 0, 0,
4, 0, 4, 4, 0, 4, 4, 4.

B.4. Coding and interleaving

The interleaver used shall be a block interleaver. Each block of input data shall also be encoded using a block encoding technique with a code block size equal to the size of the block interleaver. Thus, the input data bits will be sent as successive blocks of bits that span the duration of the interleaver length selected. Table B.4-1 shows the number of input data bits per block as function of both data rate and interleaver length. Note that an “input data block” should not be confused with the 180 or 80 symbol data block that is part of a data frame in the waveform format. The bits from an input data block will be mapped through the coding and interleaving to the number of data frames, and thus 180 or 80 symbol data blocks, that define the interleaver length.

Table B.4-1: Input Data Block Size in Bits as a Function of Data Rate and

Interleaver Length

Data Rate (kbps)	Interleaver Length in ms			
	20	80	320	1280
	Input Data Block Size (bits)			
16	320	1280	5120	20480
32	640	2560	10240	40960
48	960	3840	15360	61440
64	1280	5120	20480	81920
80	1600	6400	25600	102400
96	1920	7680	30720	122880

B.4.1 Block Boundary Alignment

Each code block shall be interleaved within a single interleaver block of the same size. The boundaries of these blocks shall be aligned such that the beginning of the first data frame following each reinserted preamble shall coincide with an interleaver boundary. Thus for an interleaver length of 8 frames, the first eight data frames following a reinserted preamble will contain all of the encoded bits for a single input data block. The first data symbol from the first data frame in each interleaver set shall have as its Most Significant Bit (MSB) the first bit fetched from the interleaver.

B.4.2 Block Encoding

The full-tail-biting and puncturing techniques shall be used with a rate ½ convolutional code to produce a block code that is the same length as the interleaver. For those data rates where the code is punctured to rate $^{8}/_{9}$, ¾ or ⅔ the punctured block shall still fit exactly within the interleaver.

B.4.3 Rate ½ Convolutional Code

A constraint length 9, rate ½ convolutional code shall be used prior to puncturing. Figure B.5.3-1 is a pictorial representation of the encoder. The two generator polynomials used shall be:

$T_1 = x^8 + x^6 + x^5 + x^4 + 1$
$T_2 = x^8 + x^7 + x^6 + x^5 + x^3 + x^1 + 1$

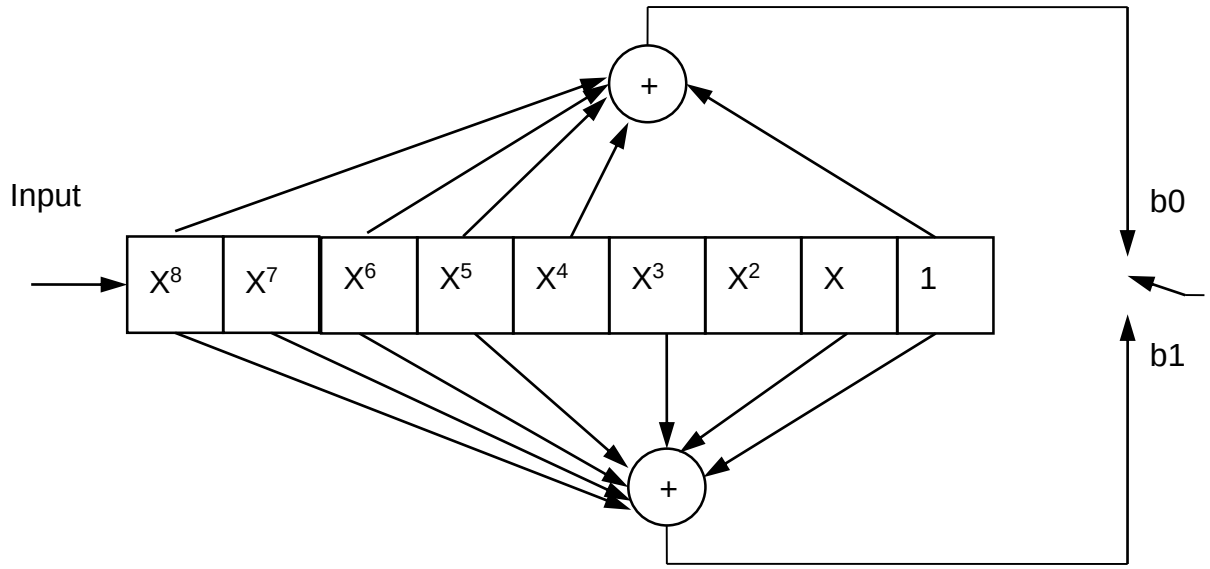


Figure B.5.3-1: Constraint Length 9, Rate ½ Convolutional Encoder

The two summing nodes in the figure represent modulo 2 addition. For each bit input to the encoder, two bits are taken from the encoder, with the upper output bit, $T_1(x)$, taken first.

B.4.4 Full-Tail-Biting Encoding

To begin encoding each block of input data, the encoder shall be preloaded by shifting in the first eight input data bits without taking any output bits. These eight input bits shall be temporarily saved so that they can be used to "flush" the encoder. The first two coded output bits shall be taken after the ninth bit has been shifted in, and shall be defined to be the first two bits of the resulting block code. After the last input data bit has been encoded, the first eight "saved" data bits shall be encoded. Note that the encoder shift register should not be changed before encoding these saved bits; i.e., it should be filled with the last nine input data bits. The eight "saved" data bits are encoded by shifting them into the encoder one at a time, beginning with the earliest of the eight. The encoding thus continues by taking the two resulting coded output bits as each of the saved eight bits is shifted in. These encoded bits shall be the final bits of the resulting (unpunctured) block code. Prior to puncturing, the resulting block code will have exactly twice as many bits as the input information bits. Puncturing of the rate ½ code to the required rate $^8/_9$ shall be done prior to sending bits to the interleaver.

B.4.5 Puncturing to Rate $^{8}/_{9}$

In order to obtain a rate $^{8}/_{9}$ code from the rate ½ code used, the output of the encoder must be punctured by not transmitting 7 bit out of every 16. Puncturing shall be performed by using a puncturing mask of 1 1 1 0 1 0 0 1 0 1 0 1 0 1 0 1, applied to the bits output from the encoder. In this notation a 1 indicates that the bit is retained and a 0 indicates that the bit is not transmitted. For an encoder generated sequence of:

$T_1(k), T_2(k), T_1(k+1), T_2(k+1), T_1(k+2), T_2(k+2) \ldots$

the transmitted sequence shall be:

$T_1(k), T_2(k), T_1(k+1), T_1(k+2) \ldots$

Defining $T_1(0)$, $T_2(0)$ to be the first two bits of the block code generated as defined in Paragraph B.4.2 , then the value of k in the above sequences shall be an integral multiple of 8. The block code shall be punctured in this manner before being input to the interleaver.

B.4.6 Puncturing to Rate ¾

In order to obtain a rate ¾ code from the rate ½ code used, the output of the encoder must be punctured by not transmitting 2 bits out of every 6. Puncturing shall be performed by using a puncturing mask of ***1 1 1 0 1 0***, applied to the bits output from the encoder. In this notation a 1 indicates that the bit is retained and a 0 indicates that the bit is not transmitted. For an encoder generated sequence of:

$T_1(k), T_2(k), T_1(k+1), T_2(k+1), T_1(k+2), T_2(k+2), T1(k+3), \ldots$

the transmitted sequence shall be:

$T_1(k), T_2(k), T_1(k+1), T_1(k+2), T1(k+3), \ldots$

Defining $T_1(0)$, $T_2(0)$ to be the first two bits of the block code generated as defined in Paragraph B.4.2 , then the value of k in the above sequences shall be an integral multiple of 3. The block code shall be punctured in this manner before being input to the interleaver.

B.4.7 Puncturing to Rate ⅔

In order to obtain a rate ⅔ code from the rate ½ code used, the output of the encoder must be punctured by not transmitting 1 bit out of every 4. Puncturing shall be performed by using a puncturing mask of ***1 1 1 0***, applied to the bits output from the encoder. In this

notation a 1 indicates that the bit is retained and a 0 indicates that the bit is not transmitted. For an encoder generated sequence of:

$T_1(k), T_2(k), T_1(k+1), T_2(k+1), T_1(k+2), \ldots$

the transmitted sequence shall be:

$T_1(k), T_2(k), T_1(k+1), T_1(k+2), \ldots$

Defining $T_1(0)$, $T_2(0)$ to be the first two bits of the block code generated as defined in Paragraph B.4.2 , then the value of k in the above sequences shall be an integral multiple of 2. The block code shall be punctured in this manner before being input to the interleaver.

B.4.8 Block Interleaver Structure

The block interleaver used is designed to separate neighbouring bits in the punctured block code as far as possible over the span of the interleaver with the largest separations resulting for the bits that were originally closest to each other. Because of the many different combinations of data rates and interleaver lengths, a flexible interleaver structure is needed.

B.4.8.1 Interleaver Size in Bits

The interleaver shall consist of a single dimension array, numbered from 0 to its size in bits –1. The array size shall depend on both the data rate and interleaver length selected as shown in Table B.4.8.1 -1.

Table B.4.8.1 -1: Interleaver Size in Bits as a Function of Data Rate and Interleaver Length

Data Rate (kbps)	Interleaver Length			
	20 ms	80 ms	320 ms	1.280 s
	Interleaver Size in Bits			
16	640	2560	10240	40960
32	960	3840	15360	61440
48	1280	5120	20480	81920
64	1440	5760	23040	92160
80	1800	7200	28800	115200
96	2160	8640	34560	138240

B.4.8.2 Interleaver Load

The punctured block code bits shall be loaded into the interleaver array beginning with location 0. The location for loading each successive bit shall be obtained from the previous location by incrementing by the “Interleaver Increment Value” specified in Table B.4.8.2 -1, modulo the “Interleaver Size in Bits.”

Defining the first punctured block code bit to be B(0), then the load location for B(n) is given by:

Load Location = (n * Interleaver Increment Value) Modulo (Interleaver Size in Bits)

Thus for 16 kbps, with a 20 ms interleaver (640 bit size with an increment of 227), the first 8 interleaver load locations are: 0, 227, 454, 41, 268, 495, 82, and 309.

Table B.4.8.2 -1: Interleaver Increment Value as a Function of Data Rate and Interleaver Length

Data Rate (kbps)	Interleaver Length			
	20 ms	80 ms	320 ms	1.280 s
	Interleaver Increment Value			
16	227	681	1847	12417
32	307	1369	6043	22387
48	357	1821	7637	23637
64	569	1673	8177	13897
80	671	1031	6721	29491
96	859	2491	11149	44983

B.4.8.3 Interleaver Fetch

The fetching sequence for all data rates and interleaver lengths shall start with location 0 of the interleaver array and increment the fetch location by 1. This is a simple linear fetch from beginning to end of the interleaver array.

B.5. Operational Features and Message Protocols

B.5.1 User Interfaces

B.5.1.1 Conventional Asynchronous Interface

The modem shall be capable of interfacing with an asynchronous DTE. In this case the DTE provides (accepts) asynchronous Words consisting of a Start Bit, an N bit Character, and some minimum number of Stop Bits. Additional Stop Bits are provided (accepted) by the DTE between Words as necessary to accommodate gaps between their occurrence. Interoperability shall be provided for those cases where the value of N, the number of Bits in the Character, is 5, 6, 7, or 8 (including any parity bits), and the minimum number of Stop Bits is 1 or 2. Hence interoperability is defined for those cases where the number of Bits in the Word is N+2 and N+3. In these cases the entire N+2 or N+3 bits of the Word shall be conveyed contiguously in the modulated signal. Additional Stop Bits shall be conveyed as necessary to accommodate gaps in data from the DTE; there shall be no modem-defined null character incorporated into the modulated signal.

B.5.1.2 High Speed Asynchronous User Interface with Flow Control

Certain high speed user interfaces provide data to (and accept data from) the modem in units of 8 bit bytes. Furthermore, the Input Data Blocks shown in Table B.4-1 are all multiples of 8 bit bytes. An optional mode shall be provided to accommodate the special

case of an 8 bit character (which includes any parity check bits) and a 1.0 unit interval Stop Bit. In this optional mode, the 8 bit Character shall be aligned with the 180 or 80 symbol modem data block boundary, and no Start or Stop Bits shall be transmitted. In this mode of operation it is assumed that the DTE data rate is greater than that which can be accommodated by the modem. Consequently flow control shall be used to temporarily stop data flow from the DTE to the modem when the modems input buffer becomes full. Conversely, when the modems input buffer becomes empty, the modem shall assume that the DTE has finished its message, and the modem shall initiate its normal message-termination procedure. This method of operation obviates the need for the transmission of Null characters for the purpose of “rate padding.” Consequently, no Null characters shall be transmitted for this purpose.

B.5.1.3 Ethernet Interface (Optional)

The modem shall provide an Ethernet interface. The bytes shall be aligned with Input Data Block boundaries.

B.5.2 Onset of Transmission

The modem shall begin a transmission no later than 100 ms after it has received an entire input data block (enough bits to fill a coded and interleaved block), or upon receipt of the last input data bit, whichever occurs first. The latter would only occur when the message is shorter than one interleaver block. A transmission shall be defined as beginning with the keying of the radio, followed by the output of the preamble waveform after the configured pre-key delay, if any. In order to be considered compliant with this standard, a modem must be able to meet this condition with no pre-key delay.

B.5.3 End of Message

The use of an end-of-message (EOM) in the transmit waveform shall be a configurable option. When the use of an EOM has been selected, a 32-bit EOM pattern shall be appended after the last input data bit of the message. The EOM, expressed in hexadecimal notation is 4B65A5B2, where the left most bit is sent first. If the last bit of the EOM does not fill out an input data block, the remaining bits in the input data block shall be set to zero before encoding and interleaving the block.

If the use of an EOM has been inhibited, and the last input data bit does not fill out an input data block, the remaining bits in the input data block shall be set to zero before encoding and interleaving the block. It is anticipated that the use of an EOM will only be inhibited when an ARQ data protocol uses ARQ blocks which completely fill (or nearly so) the selected input data block size (interleaver block).

B.5.4 Termination of a Transmission

Upon receipt of a radio silence (or equivalent) command, the modem shall immediately un-key the radio and terminate its transmit waveform.

In normal operation, the modem shall terminate a transmission only after the transmission of the final data frame, including a mini-probe, associated with the final interleaver block. Note that a data frame consists of a 180 or 80 symbol data block followed by a mini-probe. The entire final mini-probe shall be transmitted before the transmitter power is turned off.

B.5.5 Termination of Receive Data Processing

B.5.5.1 Detection of EOM

The modem shall always scan all of the decoded bits for the 32-bit EOM pattern defined in Paragraph B.5.3 . Upon detection of the EOM the modem shall return to the acquisition mode. The modem shall continue to deliver decoded bits to the user (DTE) until the final bit immediately preceding the EOM has been delivered.

B.5.5.2 Command to Return to Acquisition

Upon receipt of a command to terminate reception, the modem shall return to the acquisition mode and terminate the delivery of decoded bits to the user (DTE).

B.5.5.3 Receipt of a Specified Number of Data Blocks

The maximum message duration measured in number of Input Data Blocks (interleaver blocks) shall be a configurable parameter. Setting this parameter to zero shall specify that an unlimited number may be received. Once the modem has decoded and delivered to the user (DTE), the number of bits corresponding to the configured maximum message duration, the modem shall return to the acquisition mode and terminate the delivery of decoded bits to the user (DTE). Note that for a given interleaver length, this parameter also specifies the maximum message duration in time, independent of the bit rate. Note that this parameter is the maximum duration and that the transmit end always has the option of using an EOM for shorter transmissions.

Operation with a specified number of input data blocks may be used by an ARQ or TDMA protocol where the size of the packet is fixed, or occasionally changed to accommodate changing propagation conditions. In this case we anticipate that this parameter (maximum message duration) will be sent to the receiving end of the link as part of the ARQ protocol or specified in the TDMA protocols. It would then have to be sent to the receiving modem through the remote control interface or be preconfigured by an operator since it is not embedded in the waveform itself as the data rate and interleaver length parameters are.

B.5.5.4 Initiation of a Transmission

If, and only if, the Modem is configured to operate in half-duplex mode with transmit override, the initiation of a transmission by the user (DTE) shall cause the modem to terminate the receive processing and the delivery of decoded bits to the user (DTE).

B.5.6 TDMA Compatibility

B.5.6.1 Jitter

The total uncertainty in the time between the receipt of Request to Send (RTS) and the onset of the transmission (defined above) shall be less than 20 ms.

B.5.6.2 Minimum Time Between Back-to-Back Receptions

The modem shall be capable of receiving a subsequent transmission within 10 ms of the end of an initial transmission if the modem is configured to ignore the autobaud information in the waveform and the same data rate and interleaver settings are used for the initial and subsequent transmissions. If the modem is configured to accept autobaud data rate and interleaver settings, the minimum time between back to back receptions shall be 1.2 * the interleaver length.

B.6. Performance Requirements

The minimum performance requirements for modem performance in an Additive White Gaussian Noise (AWGN) environment are specified in this section. In addition to the AWGN specification included here, the modem/radio system should be tested using a channel simulator to ensure adequate performance in a multipath fading environment. Performance specification for testing of this sort is not possible without precisely specifying the channel simulator to be used and is consequently beyond the scope of this standard. Nonetheless, it is strongly suggested that when evaluating modems based on this standard, performance in multipath fading environments should be considered. Fair testing will result if the same channel simulator is used to test all modem/radio systems under consideration.

B.6.1 Bit Error Rate (BER) Performance

BER performance shall be measured with a channel simulator of at least 25 kHz bandwidth, programmed to simulate an AWGN channel. The signal to noise ratio (SNR) shall be measured within the 25 kHz audio bandwidth. The AWGN channel shall consist of a single, non-fading path. Each condition shall be measured for at least 30 minutes for 1.0E-5 BER using the shortest interleaver setting.

Table B.6-1: Minimum SNR to Achieve 10^{-5} BER in AWGN

Modem Data Rate (kbps)	SNR (dB in 25 kHz) for 1e-5 BER
16	8
32	13
48	16
64	18
80	21
96	23

ANNEX C (OPTIONAL) TECHNICAL SPECIFICATIONS TO ENSURE INTEROPERABILITY OF SERIAL WAVEFORMS FOR 100 KHZ BANDWIDTH LINE OF SIGHT VHF AND UHF RADIO DATA COMMUNICATIONS

C.1. Introduction

C.1.1 Purpose

This Annex provides a detailed description of modem waveforms to ensure interoperability within complying Very High Frequency (VHF) and Ultra High Frequency (UHF) radio networks.

C.1.2 Approach and Structure of this Document

The waveforms described in AComP-4691 are designed to satisfy NATO requirements. This Annex specifies waveforms for general applications, including broadcast, Automatic Repeat reQuest (ARQ) and time-division multiple access (TDMA) operation.

C.1.3 Overview

This section presents a modem waveform and coding specification for data rates of 32, 64, 128, 192, 256, 320 and 384 kilobits per second (kbps) operating within channel bandwidths of 100 kHz. This self-identifying waveform family is required for 32 - 384 kbps operation for systems complying with this STANAG.

A block interleaver is used to obtain 4 interleaving lengths 3 milliseconds (ms), 24 ms, 192 ms and 1.536 s. A rate ½, a constraint length 9, convolutional code is used. For data rates of 384, 320 and 256 kbps, the code is punctured to rate $^{8}/_{9}$. For rates of 192 and 128 kbps, the code is punctured to rate ⅔. A full-tail-biting approach, with the block size equal to the interleaver length, is used.

The waveform uses Phase Shift Keying (PSK) and Quadrature Amplitude Modulation (QAM).

The data rate and interleaver settings are explicitly transmitted as a part of the waveform, both as part of the initial preamble and then periodically as both a reinserted preamble and in the periodic known symbol blocks. The receive modem is required to be able to deduce the data rate and interleaver setting either from the preamble or from the reinserted preamble portion of the waveform.

A summary of the basic characteristics is provided in Table C.1.3 -1 below.

Table C.1.3 -1: Summary of Basic Waveform Characteristics

Data Rate	Modulation	Block Size		Code Rate	Interleaver Settings (ms)	Reinserted Preamble Size (symbols)
		Data	Training			
32	BPSK	96	23	½	3, 24, 192, 1536	64
64	QPSK			½		
128	8PSK			⅔		
192	16QAM	216	23	⅔		
256	16QAM			8/9		
320	32QAM			8/9		
384	64QAM			8/9		

C.2. Modulation

The symbol rate for all symbols shall be 80000 symbols-per-second, which shall be accurate to a minimum of ±0.8 symbols-per-second (10 parts per million (ppm)) when the transmit data clock is generated by the modem and not provided by the data terminal equipment (DTE). Phase-shift keying (PSK) and quadrature amplitude modulation (QAM) modulation techniques shall be used. The sub-carrier (or pair of quadrature sub-carriers in the case of QAM) shall be centred at f_c accurate to a minimum of 0.1 ppm. The phase of the Quadrature sub-carrier relative to the In-phase carrier shall be 90 degrees. The correct relationship can be achieved by making the In-phase sub-carrier cos(f_c) and the Quadrature sub-carrier -sin(f_c).

The power spectral density of the modulator output signal should be constrained to be at least 20 decibels (dB) below the signal level measured at f_c, when tested outside of the band from f_c-50 kHz to f_c +50 kHz. The measured ripple shall be no more than ±2 dB in the range from f_c-35 kHz to f_c +35 kHz. The pulse shaping filter which shall be employed is a root-raised cosine filter, specified by the T/2 (160 kHz sampling rate) spaced coefficients of the finite impulse response (FIR) filter:

0.009296833061855815,
-0.007502635967975916,
-0.002933029151645450,
0.021220659078919377,
-0.018296424333794074,
-0.037513179839879340,
0.065296144145033014,
0.053051647697298407,
-0.170297633845999025,

-0.064237155776998595,
0.621797410509131732,
1.068309886183790747,
0.621797410509131732,
-0.064237155776998595,
-0.170297633845999025,
0.053051647697298407,
0.065296144145033014,
-0.037513179839879340,
-0.018296424333794074,
0.021220659078919377,
-0.002933029151645450,
-0.007502635967975916,
0.009296833061855815

C.2.1 Known Symbols

For all known symbols, which are defined as those symbols which are known to the receiver prior to reception, the modulation used shall be PSK, with the symbol mapping shown in Table C.2.1 -1 and Figure C.2.1 -1. No scrambling shall be applied to the known symbols.

Table C.2.1 -1: 8PSK Symbol Mapping

Symbol Number	Phase	In-Phase	Quadrature
0	0	1.000000	0.000000
1	$\pi/4$	0.707107	0.707107
2	$\pi/2$	0.000000	1.000000
3	$3\pi/4$	-0.707107	0.707107
4	π	-1.000000	0.000000
5	$5\pi/4$	-0.707107	-0.707107
6	$3\pi/2$	0.0000000	-1.000000
7	$7\pi/4$	0.707107	-0.707107

Note that the complex symbol values = exp[jnπ/4] where n is the symbol number.

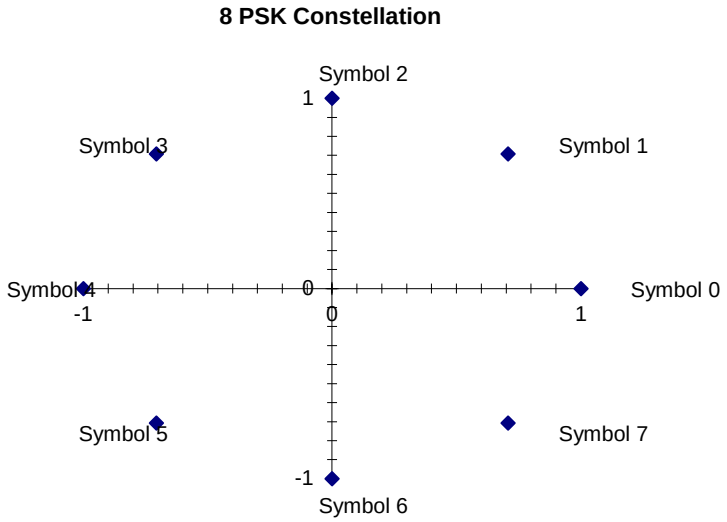


Figure C.2.1 -1: 8PSK Signal Constellation and Symbol Mapping

C.2.2 Data Symbols

For data symbols, the modulation used shall depend upon the data rate. Table C.2.2 -1 specifies the modulation that shall be used with each data rate.

Table C.2.2 -1: Modulation Used to Obtain Each Data Rate

Data Rate (kbps)	Modulation
32	BPSK
64	QPSK
128	8PSK
192	16QAM
256	16QAM
320	32QAM
384	64QAM

All PSK constellations are scrambled to appear, on-air, as an 8PSK constellation. The QAM constellations have been modified from conventional rectangular constellations to improve the peak-to-average ratio.

C.2.2.1 PSK Data Symbols

For the PSK constellations, a distinction is made between the data bits and the symbol number for the purposes of scrambling the QPSK modulation to appear as 8PSK, on-air. Scrambling is applied as a modulo 8 addition of a scrambling sequence to the 8PSK

symbol number. Transcoding is an operation which links a symbol to be transmitted to a group of data bits.

C.2.2.1.1 BPSK Symbol Mapping

For the 32 kbps user data rate, transcoding shall be achieved by linking one of the symbols specified in Table B.2.1 -1 to a set of two consecutive data bits (dibit) as shown in Table C.2.2.1.1 -1.

Table C.2.2.1.1 -1: Transcoding for 32 kbps

Bit	Symbol
0	0
1	4

C.2.2.1.2 QPSK Symbol Mapping

For the 64 kbps user data rates, transcoding shall be achieved by linking one of the symbols specified in Table C.2.1 -1 to a set of two consecutive data bits (dibit) as shown in Table C.2.2.1.2 -1. In this table, the leftmost bit of the dibit shall be the older bit; i.e., fetched from the interleaver before the rightmost bit.

Table C.2.2.1.2 -1: Transcoding for 64 kbps

Dibit	Symbol
00	0
01	2
11	4
10	6

C.2.2.1.3 8PSK Symbol Mapping

For the 128 kbps user data rate, transcoding shall be achieved by linking one symbol to a set of three consecutive data bits (tribit) as shown in Table C.2.2.1.3 -1. In this table, the leftmost bit of the tribit shall be the oldest bit; i.e., fetched from the interleaver before the other two, and the rightmost bit is the most recent bit.

Table C.2.2.1.3 -1: Transcoding for 128 kbps

Tribit	Symbol
000	1
001	0
010	2
011	3
100	6
101	7
110	5
111	4

C.2.2.1.4 QAM data symbols

For the QAM constellations, no distinction is made between the number formed directly from the data bits and the symbol number. Each set of 4 bits (16QAM), 5 bits (32QAM) or 6 bits (64QAM) is mapped directly to a QAM symbol. For example, the four bit grouping 0111 would map to symbol 7 in the 16QAM constellation while the 6 bit grouping 100011 would map to symbol 35 in the 64QAM constellation. Again, in each case the leftmost bit shall be the oldest bit, i.e. fetched from the interleaver before the other bits, and the rightmost bit is the most recent bit.

The mapping of bits to symbols for the QAM constellations has been selected to minimise the number of bit errors incurred when errors involve adjacent signalling points in the constellation.

C.2.2.1.5 The 16QAM constellation

The constellation points which shall be used for 16QAM are shown in Figure C.2.2.1.5 -1 and specified in terms of their In-phase and Quadrature components in Table C.2.2.1.5 -1.

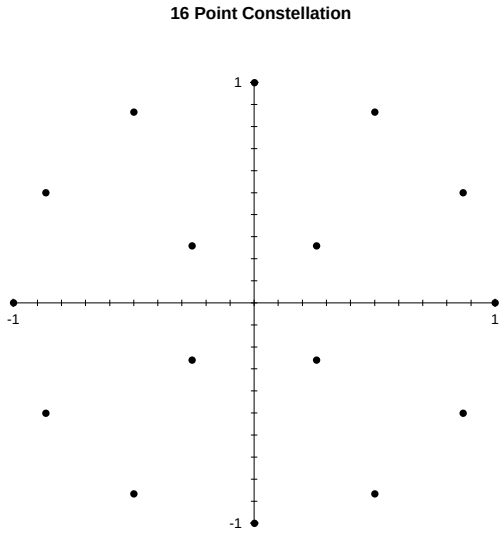


Figure C.2.2.1.5 -1: 16QAM Signalling Constellation

Table C.2.2.1.5 -1: In-phase and Quadrature Components of Each 16QAM Symbol

Symbol Number	In-Phase	Quadrature
0	0.866025	0.500000
1	0.500000	0.866025
2	1.000000	0.000000
3	0.258819	0.258819
4	-0.500000	0.866025
5	0.000000	1.000000
6	-0.866025	0.500000
7	-0.258819	0.258819
8	0.500000	-0.866025
9	0.000000	-1.000000
10	0.866025	-0.500000
11	0.258819	-0.258819
12	-0.866025	-0.500000
13	-0.500000	-0.866025
14	-1.000000	0.000000
15	-0.258819	-0.258819

C.2.2.1.6 The 32QAM constellation

The constellation points which shall be used for 32QAM are shown in Figure B.2.2.1.5 -1 and specified in terms of their In-phase and Quadrature components in Table B.2.2.1.5 -1.

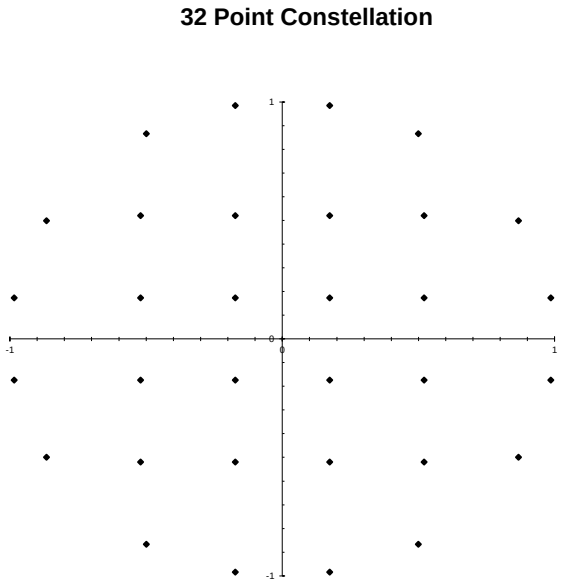


Figure B.2.2.1.5 -1: 32QAM Signalling Constellation

Table B.2.2.1.5 -1: In-phase and Quadrature Components of Each 32QAM Symbol

Symbol Number	In-Phase	Quadrature	Symbol Number	In-Phase	Quadrature
0	0.866380	0.499386	16	0.866380	-0.499386
1	0.984849	0.173415	17	0.984849	-0.173415
2	0.499386	0.866380	18	0.499386	-0.866380
3	0.173415	0.984849	19	0.173415	-0.984849
4	0.520246	0.520246	20	0.520246	-0.520246
5	0.520246	0.173415	21	0.520246	-0.173415
6	0.173415	0.520246	22	0.173415	-0.520246
7	0.173415	0.173415	23	0.173415	-0.173415
8	-0.866380	0.499386	24	-0.866380	-0.499386
9	-0.984849	0.173415	25	-0.984849	-0.173415
10	-0.499386	0.866380	26	-0.499386	-0.866380
11	-0.173415	0.984849	27	-0.173415	-0.984849
12	-0.520246	0.520246	28	-0.520246	-0.520246
13	-0.520246	0.173415	29	-0.520246	-0.173415
14	-0.173415	0.520246	30	-0.173415	-0.520246
15	-0.173415	0.173415	31	-0.173415	-0.173415

C.2.2.1.7 The 64QAM constellation

The constellation points which shall be used for the 64QAM modulation are shown in Figure C.2.2.1.7 -1 and specified in terms of their In-phase and Quadrature components in Table C.2.2.1.7 -1.

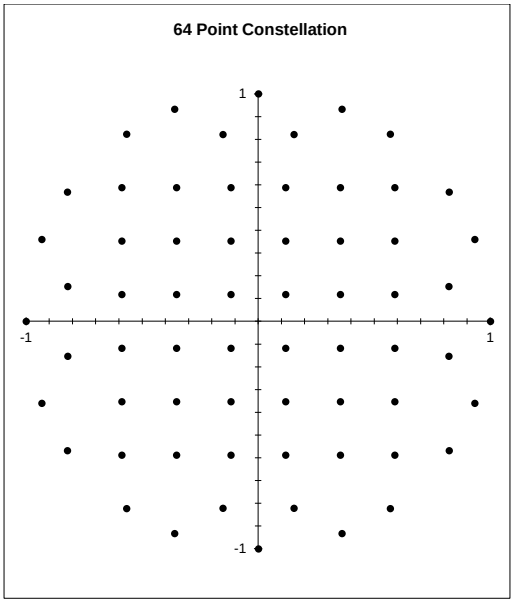


Figure C.2.2.1.7 -1: 64QAM Signalling Constellation

Table C.2.2.1.7 -1: In-phase and Quadrature Components of Each 64QAM Symbol

Symbol Number	In-Phase	Quadrature	Symbol Number	In-Phase	Quadrature
0	1.000000	0.000000	32	0.000000	1.000000
1	0.822878	0.568218	33	-0.822878	0.568218
2	0.821137	0.152996	34	-0.821137	0.152996
3	0.932897	0.360142	35	-0.932897	0.360142
4	0.000000	-1.000000	36	-1.000000	0.000000
5	0.822878	-0.568218	37	-0.822878	-0.568218
6	0.821137	-0.152996	38	-0.821137	-0.152996
7	0.932897	-0.360142	39	-0.932897	-0.360142
8	0.568218	0.822878	40	-0.568218	0.822878
9	0.588429	0.588429	41	-0.588429	0.588429
10	0.588429	0.117686	42	-0.588429	0.117686
11	0.588429	0.353057	43	-0.588429	0.353057
12	0.568218	-0.822878	44	-0.568218	-0.822878
13	0.588429	-0.588429	45	-0.588429	-0.588429
14	0.588429	-0.117686	46	-0.588429	-0.117686
15	0.588429	-0.353057	47	-0.588429	-0.353057
16	0.152996	0.821137	48	-0.152996	0.821137
17	0.117686	0.588429	49	-0.117686	0.588429
18	0.117686	0.117686	50	-0.117686	0.117686
19	0.117686	0.353057	51	-0.117686	0.353057
20	0.152996	-0.821137	52	-0.152996	-0.821137
21	0.117686	-0.588429	53	-0.117686	-0.588429
22	0.117686	-0.117686	54	-0.117686	-0.117686
23	0.117686	-0.353057	55	-0.117686	-0.353057
24	0.360142	0.932897	56	-0.360142	0.932897
25	0.353057	0.588429	57	-0.353057	0.588429
26	0.353057	0.117686	58	-0.353057	0.117686
27	0.353057	0.353057	59	-0.353057	0.353057
28	0.360142	-0.932897	60	-0.360142	-0.932897
29	0.353057	-0.588429	61	-0.353057	-0.588429
30	0.353057	-0.117686	62	-0.353057	-0.117686
31	0.353057	-0.353057	63	-0.353057	-0.353057

C.2.3 Data Scrambling

Data symbols for the 8PSK symbol constellation (32 kbps, 64 kbps, 128 kbps) shall be scrambled by modulo 8 addition with a scrambling sequence. The data symbols for the 16QAM, 32QAM, and 64QAM constellations shall be scrambled by using an exclusive or (XOR) operation. Sequentially, the data bits forming each symbol (4 for 16QAM, 5 for 32QAM, and 6 for 64QAM) shall be XOR'd with an equal number of bits from the

scrambling sequence. The scrambling sequence generator polynomial shall be $x^9 + x^4 + 1$ and the generator shall be initialised to 1 at the start of each data frame for data rates greater than 128 kbps (which use a 216 symbol data block) and initialized to 1 at the start of every second frame, beginning with the first data frame following the preamble, for data rates of 128 kbps and below. A block diagram of the scrambling sequence generator is shown in Figure C.2.3 -1.

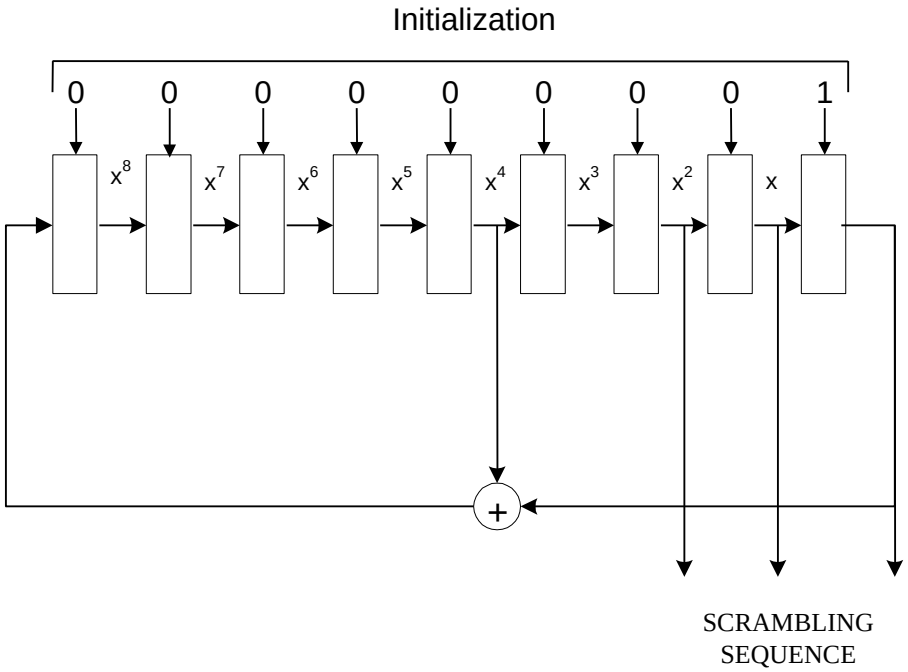


Figure C.2.3 -1: Scrambling Sequence Generator Illustrating Scrambling Generator for 8PSK Symbols

For 8PSK symbols (32, 64 and 128 kbps), the scrambling shall be carried out taking the modulo 8 sum of the numerical value of the binary triplet consisting of the last (rightmost) three bits in the shift register, and the symbol number (transcoded value). For example, if the last three bits in the scrambling sequence shift register were 010 which has a numerical value equal 2, and the symbol number before scrambling was 6, symbol 0 would be transmitted since: (6+2) Modulo 8 = 0. For 16QAM symbols, scrambling shall be carried out by XORing the 4 bit number consisting of the last (rightmost) four bits in the shift register with the symbol number. For example, if the last 4 bits in the scrambling sequence shift register were 0101 and the 16QAM symbol number before scrambling was 3 (i.e. 0011), symbol 6 (0110) would be transmitted. For 32QAM symbols, scrambling shall be carried out by XORing the 5 bit number formed by the last (rightmost) five bits in the

shift register with the symbol number. For 64QAM symbols, scrambling shall be carried out by XORing the 6 bit number formed by the last (rightmost) six bits in the shift register with the symbol number.

After each data symbol is scrambled, the generator shall be iterated (shifted) the required number of times to produce all new bits for use in scrambling the next symbol (i.e., 3 iterations for 8PSK, 4 iterations for 16QAM, 5 iterations for 32QAM and 6 iterations for 64QAM). Since the generator is iterated after the bits are used, the first data symbol of every data frame shall, therefore, be scrambled by the appropriate number of bits from the initialisation value of 00000001.

The length of the scrambling sequence is 511 bits. For a 216 symbol data block with 6 bits per symbol, this means that the scrambling sequence will be repeated just slightly more than 2 ½ times. There will be no repetition of symbols during the interval.

C.3. Frame structure

The frame structure that shall be used for the waveforms specified in this section is shown in Figure C.3-1. An initial 240 symbol preamble is followed by 64 frames of alternating data and known symbols. Each data frame shall consist of a data block followed by a mini-probe consisting of 23 symbols of known data. The data block shall consist of 216 data symbols for data rates of 384, 320, 256, or 192 kbps, and 96 data symbols for data rates of 128, 64 or 32 kbps. After 64 data frames, a 64 symbol subset of the initial preamble is reinserted to facilitate late acquisition, Doppler shift removal, and sync adjustment. It should be noted that the total length of known data in this segment is actually 87 symbols: the 64 reinserted preamble symbols plus the preceding 23 symbol mini-probe segment which follows the last data block. Reinserted preambles will occur twice as often for data rates of 32-128 kbps as they do for data rates of 192-384 kbps because of the smaller size of the data block.

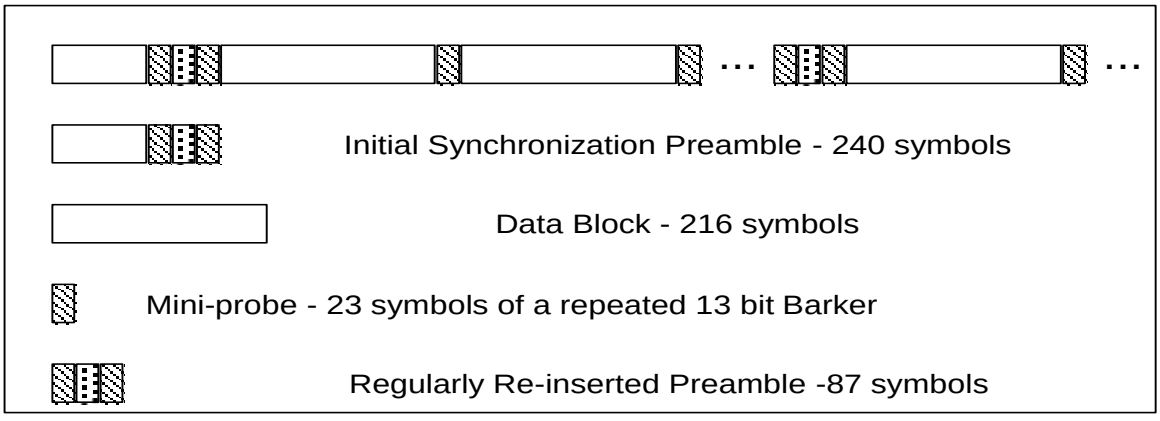


Figure C.3-1: Frame Structure for Data Rates of 192, 256, 320 and 384 kbps

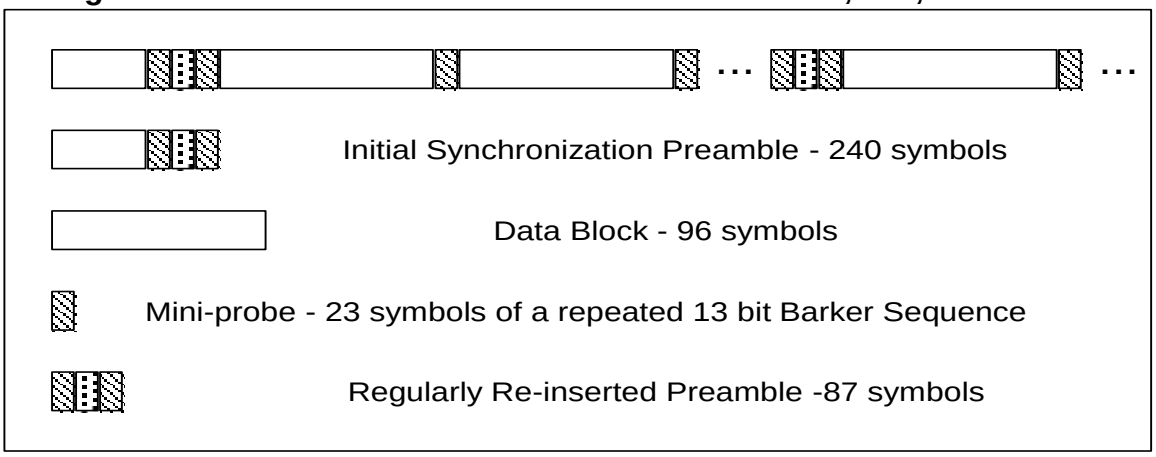


Figure C.3-2: Frame Structure for Data Rates of 32, 64, and 128 kbps

C.3.1 Synchronisation and Reinserted Preambles

The synchronisation preamble is used for rapid initial synchronisation. The reinserted preamble is used to facilitate acquisition of an ongoing transmission (acquisition on data).

C.3.1.1 Synchronisation Preamble

The mandatory portion of the preamble, which shall always be transmitted, and shall always appear at the beginning of an interleaver boundary, shall consist of 240 symbols. The first 153 symbols are intended exclusively for synchronisation and Doppler offset removal purposes while the final 87 symbols, which are common with the reinserted preamble, also carry information regarding the data rate and interleaver settings. Expressed as a sequence of 8PSK symbols, using the symbol numbers given in Table C.2.1 -1 the second section of the synchronisation preamble shall be as follows:

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

0, 4, 0, 4, 0, 0, 4, 4, 0, 0, 0, 0, 0,
0, 4, 0, 4, 0, 0, 4, 4, 0, 0, 0, 0, 0,
0,

(D_0, D_0, D_0, D_0, D_0, D_0, D_0, D_0, D_0, D_0, D_0 + 4, 0, 4, 4, 0, 4, 4, 4, 0, 0, 0) Modulo 8
(D_1, D_1, D_1, D_1, D_1, D_1, D_1, D_1, D_1, D_1, D_1 + 4, 0, 4, 4, 0, 4, 4, 4, 0, 0, 0) Modulo 8
(D_2, D_2, D_2, D_2, D_2, D_2, D_2, D_2, D_2, D_2, D_2 + 4, 0, 4, 4, 0, 4, 4, 4, 0, 0, 0) Modulo 8

4, 0, 0, 0, 0, 0,
0, 4, 0, 4, 0, 0, 4, 4, 0, 0, 0, 0, 0
0, 4, 0, 4, 0, 0, 4, 4.

…where the data symbols D_0, D_1, and D_2 take one of the sets of values chosen from Table C.3.1.1 -1 to indicate the data rate and interleaver settings. The Modulo operations are meant to signify that each of the D values are used to shift the phase of a length 11 bit Barker code (01001000111) by performing modulo 8 addition of the D value with each of the Barker code 11 phase values (0 or 4). This operation can encode 6 bits of information using QPSK modulation of the 11 bit (chip) Barker codes. Note that the 13 bit Barker codes before and after the 11 bit Barkers encoding the D values have been extended from the usual 23 bits to 27 bits.

Table C.3.1.1 -1: D0, D1, D2 8 PSK Symbol Values as a Function of Data Rate and Interleaver Length at the Start of an Interleaver

Data Rate (kbps)	Interleaver Length in Frames			
	3 ms	24 ms	192 ms	1536 ms
32	0, 2, 0	0, 2, 2	0, 2, 6	0, 2, 4
64	0, 6, 0	0, 6, 2	0, 6, 6	0, 6, 4
128	0, 4, 0	0, 4, 2	0, 4, 6	0, 4, 4
192	2, 0, 0	2, 0, 2	2, 0, 6	2, 0, 4
256	2, 2, 0	2, 2, 2	2, 2, 6	2, 2, 4
320	2, 6, 0	2, 6, 2	2, 6, 6	2, 6, 4
384	2, 4, 0	2, 4, 2	2, 4, 6	2, 4, 4

.

The synchronisation preamble may consist of two parts. The non-mandatory portion, which shall be transmitted before the mandatory portion described above if it is present, will contain a count-down and shall consist of N blocks of 240 8-PSK symbols. The value of N shall be configurable to range from values of 0 to 511 (for N=0 this first section is not sent at all). When present, the 240 symbols of the non-mandatory, count-down section of the preamble shall be identical to those sent in the mandatory preamble with the exception of the values of the data symbols D_0, D_1, and D_2. The mapping that was chosen to create Table C.3.1.1 -1 used 3 bits to specify the data rate and two bits to specify the interleaver length. The phase of the Barker code was determined from 3 dibit words using Table C.2.2.1.1 -1, the dibit transcoding table, to transcode bit values into symbols. Numbering the bits from b_0 to b_5, with D_0 formed from the transcoding of b_5b_4, D_1 formed from the transcoding of b_3b_2 and D_2 formed from the transcoding of b_1b_0, it can be seen that for the mapping shown in Table C.3.1.1 -1 b_5 is always zero when specifying the data rate and interleaver. When b_5 is 1, the last two bits, corresponding to D_2, are used to specify a count down. The values of the data symbols D_0, D_1, and D_2 for the count down are shown in Table C.3.1.1 -2.

Table C.3.1.1 -2: D0, D1, D2 8PSK Symbol Values Specifying Count of Preambles Before the Start of an Interleaver

Number of Preambles Before Interleaver Boundary	D0, D1, D2

>3	(4 or 6), (any), 4
3	(4 or 6), (any), 6
2	(4 or 6), (any), 2
1	(4 or 6), (any), 0

Not all preamble segments appear at the beginning of an interleaver boundary. There are three instances where this occurs: 1) during repeated preamble segments at the beginning of a transmission; 2) multiple reinserted preambles during a long interleaver; and 3) the single preamble inserted in the middle of the Medium interleaver block for data rates of 128 kbps and below. In each of these cases, the data inserted into the preamble differs from the data included in a preamble at the beginning of an interleaver. Specifically, b5 is set to 1 and a count is encoded into the last two bits, corresponding to D_2 as specified in Table C.3.1.1 -2.

C.3.1.2 Reinserted Preamble

The reinserted preamble shall be identical to the final 64 symbols of the synchronisation preamble whenever it occurs immediately prior to the beginning of an interleaver block. In fact, the final 87 symbols are common between the synchronisation preamble and the contiguous block consisting of the reinserted preamble and the mini-probe which immediately precedes it. The 87 symbols of known data (including the 23 mini-probe symbols of the preceding data frame) are thus:

0, 4, 0, 4, 0, 0, 4, 4, 0, 0, 0, 0, 0,
0, 4, 0, 4, 0, 0, 4, 4, 0, 0, 0, 0, 0,
0,

(D_0, D_0, D_0, D_0, D_0, D_0, D_0, D_0, D_0, D_0 + 4, 0, 4, 4, 0, 4, 4, 4, 0, 0, 0) Modulo 8
(D_1, D_1, D_1, D_1, D_1, D_1, D_1, D_1, D_1, D_1 + 4, 0, 4, 4, 0, 4, 4, 4, 0, 0, 0) Modulo 8
(D_2, D_2, D_2, D_2, D_2, D_2, D_2, D_2, D_2, D_2 + 4, 0, 4, 4, 0, 4, 4, 4, 0, 0, 0) Modulo 8

4, 0, 0, 0, 0, 0,
0, 4, 0, 4, 0, 0, 4, 4, 0, 0, 0, 0, 0
0, 4, 0, 4, 0, 0, 4, 4.

…where in the case of a reinserted preamble at the beginning of an interleaver, the data symbols D_0, D_1, and D_2 again take one of the sets of values chosen from Table C.3.1.1 -1 to indicate the data rate and interleaver settings as described in the Synchronisation Preamble section above. In the case of a reinserted preamble that is not at the beginning of an interleaver (e.g. the preamble that will occur during the middle of the medium interleaver at 128 kbps and below or one of the preambles that occur during the long interleaver), the count of preambles remaining until the next interleaver begins is encoded

into the data symbols D_0, D_1, and D_2 as described in Table C.3.1.1 -2. Note that in every case, the first 23 of these symbols are the immediately preceding mini-probe, which follows the last of the data blocks.

C.3.2 Mini-Probes

Mini-probes 23 symbols in length shall be inserted following every data block and at the end of each preamble (where they are considered to be part of the preamble). Using the 8PSK symbol mapping, each mini-probe shall be based on a 13 bit Barker sequence. The sequence that shall be used, specified in terms of the 8PSK symbol numbers, is given by:

0, 4, 0, 4, 0, 0, 4, 4, 0, 0, 0, 0, 0,
0, 4, 0, 4, 0, 0, 4, 4, 0, 0.

C.4. Coding and interleaving

The interleaver used shall be a block interleaver. Each block of input data shall also be encoded using a block encoding technique with a code block size equal to the size of the block interleaver. Thus, the input data bits will be sent as successive blocks of bits that span the duration of the interleaver length selected. Table C.4-1 shows the number of input data bits per block as function of both data rate and interleaver length. Note that an “input data block” should not be confused with the 216 or 96 symbol data block that is part of a data frame in the waveform format. The bits from an input data block will be mapped through the coding and interleaving to the number of data frames, and thus 216 or 96 symbol data blocks, that define the interleaver length.

TABLE C.4-1: Input Data Block Size in Bits as a Function of Data Rate and

Interleaver Length

Data Rate (kbps)	Interleaver Length in ms			
	3	24	192	1.536
	Input data block size (bits)			
32	96	768	6144	49152
64	192	1536	12288	98304
128	384	3072	24576	196608
192	576	4608	36864	294912
256	768	2048	16384	131072
320	960	7680	61440	491520
384	1152	9216	73728	589824

C.4.1 Block Boundary Alignment

Each code block shall be interleaved within a single interleaver block of the same size. The boundaries of these blocks shall be aligned such that the beginning of the first data frame following each reinserted preamble shall coincide with an interleaver boundary. Thus for an interleaver length of 8 frames, the first eight data frames following a reinserted preamble will contain all of the encoded bits for a single input data block. The first data symbol from the first data frame in each interleaver set shall have as its most significant bit (MSB) the first bit fetched from the interleaver.

C.4.2 Block Encoding

The full-tail-biting and puncturing techniques shall be used with a rate ½ convolutional code to produce a block code that is the same length as the interleaver. For those data rates where the code is punctured to rate $^{8}/_{9}$, the punctured block shall still fit exactly within the interleaver.

C.4.3 Rate ½ Convolutional Code

A constraint length 9, rate ½ convolutional code shall be used prior to puncturing. Figure 5.3-1 is a pictorial representation of the encoder. The two generator polynomials used shall be:

$T_1 = x^8 + x^6 + x^5 + x^4 + 1$
$T_2 = x^8 + x^7 + x^6 + x^5 + x^3 + x^1 + 1$

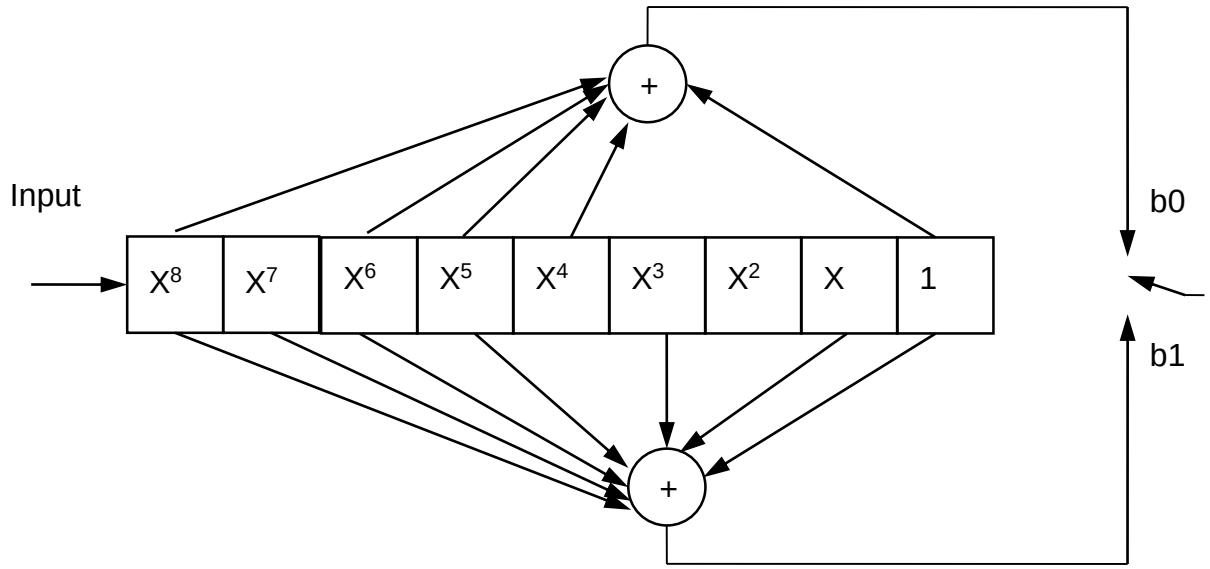


Figure C.4.3 -1: Constraint Length 9, Rate ½ Convolutional Encoder

The two summing nodes in the figure represent modulo 2 addition. For each bit input to the encoder, two bits are taken from the encoder, with the upper output bit, $T_1(x)$, taken first.

C.4.4 Full-Tail-Biting Encoding

To begin encoding each block of input data, the encoder shall be preloaded by shifting in the first eight input data bits without taking any output bits. These eight input bits shall be temporarily saved so that they can be used to "flush" the encoder. The first two coded output bits shall be taken after the ninth bit has been shifted in, and shall be defined to be the first two bits of the resulting block code. After the last input data bit has been encoded, the first eight "saved" data bits shall be encoded. Note that the encoder shift register should not be changed before encoding these saved bits; i.e., it should be filled with the last nine input data bits. The eight "saved" data bits are encoded by shifting them into the encoder one at a time, beginning with the earliest of the eight. The encoding thus continues by taking the two resulting coded output bits as each of the saved eight bits is shifted in. These encoded bits shall be the final bits of the resulting (unpunctured) block code. Prior to puncturing, the resulting block code will have exactly twice as many bits as the input information bits. Puncturing of the rate ½ code to the required rate $^8/_9$ shall be done prior to sending bits to the interleaver.

C.4.5 Puncturing to Rate $^{8}/_{9}$

In order to obtain a rate $^{8}/_{9}$ code from the rate ½ code used, the output of the encoder must be punctured by not transmitting 7 bits out of every 16. Puncturing shall be performed by using a puncturing mask of 1 1 1 0 1 0 0 1 0 1 0 1 0 1 0 1, applied to the bits output from the encoder. In this notation a 1 indicates that the bit is retained and a 0 indicates that the bit is not transmitted. For an encoder generated sequence of:

$T_1(k), T_2(k), T_1(k+1), T_2(k+1), T_1(k+2), T_2(k+2) \ldots$

the transmitted sequence shall be:

$T_1(k), T_2(k), T_1(k+1), T_1(k+2) \ldots$

Defining $T_1(0)$, $T_2(0)$ to be the first two bits of the block code generated as defined in Paragraph C.4.2 , then the value of k in the above sequences shall be an integral multiple of 8. The block code shall be punctured in this manner before being input to the interleaver.

C.4.6 Puncturing to Rate ⅔

In order to obtain a rate ⅔ code from the rate ½ code used, the output of the encoder must be punctured by not transmitting 1 bit out of every 4. Puncturing shall be performed by using a puncturing mask of 1 1 1 0, applied to the bits output from the encoder. In this notation a 1 indicates that the bit is retained and a 0 indicates that the bit is not transmitted. For an encoder generated sequence of:

$T_1(k), T_2(k), T_1(k+1), T_2(k+1), T_1(k+2), \ldots$

the transmitted sequence shall be:

$T_1(k), T_2(k), T_1(k+1), T_1(k+2), \ldots$

Defining $T_1(0)$, $T_2(0)$ to be the first two bits of the block code generated as defined in Paragraph C.4.2 , then the value of k in the above sequences shall be an integral multiple of 2. The block code shall be punctured in this manner before being input to the interleaver.

C.4.7 Block Interleaver Structure

The block interleaver used is designed to separate neighbouring bits in the punctured block code as far as possible over the span of the interleaver with the largest separations resulting for the bits that were originally closest to each other. Because of the 28 different combinations of data rates and interleaver lengths, a flexible interleaver structure is needed.

C.4.7.1 Interleaver Size in Bits

The interleaver shall consist of a single dimension array, numbered from 0 to its size in bits –1. The array size shall depend on both the data rate and interleaver length selected as shown in Table C.4.7.1 -1.

Table C.4.7.1 -1: Interleaver Size in Bits as a Function of Data Rate and Interleaver Length

Data Rate (kbps)	Interleaver Length in ms			
	3	24	192	1.536
	Interleaver Size in Bits			
32	192	1536	12288	98304
64	384	3072	24576	196608
128	576	4608	36864	294912
192	864	6912	55296	442368
256	864	6912	55296	442368
320	1080	8640	69120	552960
384	1296	10368	82944	663552

C.4.7.2 Interleaver Load

The punctured block code bits shall be loaded into the interleaver array beginning with location 0. The location for loading each successive bit shall be obtained from the previous location by incrementing by the “Interleaver Increment Value” specified in Table C.4.7.2 -1, modulo the “Interleaver Size in Bits.”

Defining the first punctured block code bit to be B(0), then the load location for B(n) is given by:

Load Location = (n * Interleaver Increment Value) Modulo (Interleaver Size in Bits)

Thus for 32 kbps, with a 3 ms interleaver (192 bit size with an increment of 59), the first 8 interleaver load locations are: 0, 59, 118, 177, 44, 103, 162, and 29.

Table C.4.7.2 -1: Interleaver Increment Value as a Function of Data Rate and

Interleaver Length

Data Rate (kbps)	Interleaver Length in ms			
	3	24	192	1.536
	Interleaver Increment Value			
32	59	547	1883	16117
64	119	661	9749	30127
128	229	991	5137	56491
192	341	1201	11981	70357
256	341	1201	11981	70357
320	371	2491	19591	102851
384	551	3001	17971	101983

C.4.7.3 Interleaver Fetch

The fetching sequence for all data rates and interleaver lengths shall start with location 0 of the interleaver array and increment the fetch location by 1. This is a simple linear fetch from beginning to end of the interleaver array.

C.5. Operational features and message protocols

C.5.1 User Interfaces

C.5.1.1 Synchronous Data Communications Equipment (DCE) Interface

The modem shall implement a synchronous DCE interface that is compatible with EIA-232, EIA-422 or Mil-Std-188-114A.

C.5.1.2 Conventional Asynchronous Interface

The modem shall be capable of interfacing with an asynchronous DTE. In this case the DTE provides (accepts) asynchronous Words consisting of a Start Bit, an N bit Character, and some minimum number of Stop Bits. Additional Stop Bits are provided (accepted) by the DTE between Words as necessary to accommodate gaps between their occurrences. Interoperability shall be provided for those cases where the value of N, the number of Bits in the Character, is 5, 6, 7, or 8 (including any parity bits), and the minimum number of Stop Bits is 1 or 2. Hence interoperability is defined for those cases where the number of Bits in the Word is N+2 and N+3. In these cases the entire N+2 or N+3 bits of the Word shall be conveyed contiguously in the modulated signal. Additional Stop Bits shall be conveyed as necessary to accommodate gaps in data from the DTE; there shall be no modem-defined null character incorporated into the modulated signal.

C.5.1.3 High Speed Asynchronous User Interface with Flow Control

Certain high speed user interfaces provide data to (and accept data from) the modem in units of 8 bit bytes. Furthermore, the Input Data Blocks shown in Table C.4-1 are all multiples of 8 bit bytes. An optional mode shall be provided to accommodate the special case of an 8 bit character (which includes any parity check bits) and a 1.0 unit interval Stop Bit. In this optional mode, the 8 bit Character shall be aligned with the 216 or 96 symbol modem frame boundary, and no Start or Stop Bits shall be transmitted. In this mode of operation it is assumed that the DTE data rate is greater than that which can be accommodated by the modem. Consequently flow control shall be used to temporarily stop data flow from the DTE to the modem when the modems input buffer becomes full. Conversely, when the modems input buffer becomes empty, the modem shall assume that the DTE has finished its message, and the modem shall initiate its normal message-termination procedure. This method of operation obviates the need for the transmission of Null characters for the purpose of "rate padding." Consequently, no Null characters shall be transmitted for this purpose.

C.5.1.4 Ethernet Interface (Optional)

The modem may provide an Ethernet interface. The bytes shall be aligned with Input Data Block boundaries.

C.5.2 Onset of Transmission

The modem shall begin a transmission no later than 100 ms after it has received an entire input data block (enough bits to fill a coded and interleaved block), or upon receipt of the last input data bit, whichever occurs first. The latter would only occur when the message is shorter than one interleaver block. A transmission shall be defined as beginning with the keying of the radio, followed by the output of the preamble waveform after the configured pre-key delay, if any.

C.5.3 End of Message

The use of an end-of-message (EOM) in the transmit waveform shall be a configurable option. When the use of an EOM has been selected, a 32-bit EOM pattern shall be appended after the last input data bit of the message. The EOM, expressed in hexadecimal notation is 4B65A5B2, where the left most bit is sent first. If the last bit of the EOM does not fill out an input data block, the remaining bits in the input data block shall be set to zero before encoding and interleaving the block.

If the use of an EOM has been inhibited, and the last input data bit does not fill out an input data block, the remaining bits in the input data block shall be set to zero before encoding and interleaving the block. It is anticipated that the use of an EOM will only be inhibited when an ARQ data protocol uses ARQ blocks which completely fill (or nearly so) the selected input data block size (interleaver block). Without this feature, the use of an EOM would require the transmission of an additional interleaver block under these circumstances.

C.5.4 Termination of a Transmission

Upon receipt of a radio silence (or equivalent) command, the modem shall immediately un-key the radio and terminate its transmit waveform.

In normal operation, the modem shall terminate a transmission only after the transmission of the final data frame, including a mini-probe, associated with the final interleaver block. Note that a data frame consists of a 216 or 96 symbol data block followed by a mini-probe. The entire final mini-probe shall be transmitted before the transmitter power is turned off.

C.5.5 Termination of Receive Data Processing

There are a number of events which shall cause the modem to cease processing the received signal to recover data, and return to the acquisition mode. These are necessary because a modem is not able to acquire a new transmission while it is attempting to demodulate and decode data.

C.5.5.1 Detection of EOM

The modem shall always scan all of the decoded bits for the 32-bit EOM pattern defined in Paragraph C.5.3 . Upon detection of the EOM the modem shall return to the acquisition mode. The modem shall continue to deliver decoded bits to the user (DTE) until the final bit immediately preceding the EOM has been delivered.

C.5.5.2 Command to Return to Acquisition

Upon receipt of a command to terminate reception, the modem shall return to the acquisition mode and terminate the delivery of decoded bits to the user (DTE).

C.5.5.3 Receipt of a Specified Number of Data Blocks

The maximum message duration measured in number of Input Data Blocks (interleaver blocks) shall be a configurable parameter. Setting this parameter to zero shall specify that an unlimited number may be received. Once the modem has decoded and delivered to the user (DTE), the number of bits corresponding to the configured maximum message duration, the modem shall return to the acquisition mode and terminate the delivery of decoded bits to the user (DTE). Note that for a given interleaver length, this parameter also specifies the maximum message duration in time, independent of the bit rate. Note that this parameter is the maximum duration and that the transmit end always has the option of using an EOM for shorter transmissions.

Operation with a specified number of input data blocks may be used by an ARQ or TDMA protocol where the size of the packet is fixed, or occasionally changed to accommodate changing propagation conditions. In this case we anticipate that this parameter (maximum message duration) will be sent to the receiving end of the link as part of the ARQ protocol or specified in the TDMA protocols. It would then have to be sent to the receiving modem through the remote control interface or be preconfigured by an operator since it is not embedded in the waveform itself as the data rate and interleaver length parameters are.

C.5.5.4 Initiation of a Transmission

If, and only if, the Modem is configured to operate in half-duplex mode with transmit override, the initiation of a transmission by the user (DTE) shall cause the modem to terminate the receive processing and the delivery of decoded bits to the user (DTE).

C.5.6 TDMA Compatibility

C.5.6.1 Jitter

The total uncertainty in the time between the receipt of Request to Send (RTS) and the onset of the transmission (defined above) shall be less than 20 ms.

C.5.6.2 Minimum Time Between Back-to-Back Receptions

The modem shall be capable of receiving a subsequent transmission within 10 ms of the end of an initial transmission if the modem is configured to ignore the autobaud information in the waveform and the same data rate and interleaver settings are used for the initial and subsequent transmissions. If the modem is configured to accept autobaud data rate and interleaver settings, the minimum time between back to back receptions shall be the greater of 1.2 * the interleaver length or 20 ms.

C.5.6.3 Associated Communications (Radio) Equipment

The radio must provide a passband of at least 90 kHz with a ripple of not more than ±2 dB across the passband. The radio must support linear modulations and must support a signal to noise and distortion (SINAD) of not less than 30 dB for the 64 QAM modulation defined herein.

C.6. Performance Requirements

The minimum performance requirements for the high data rate mode are specified in this section. These requirements are not exhaustive, but are intended to ensure that equipment contains a high-quality implementation of the standard.

C.6.1 Bit Error Rate (BER) Performance

BER performance shall be measured with the channel simulator programmed to simulate an additive white Gaussian noise (AWGN) channel. The AWGN channel shall consist of a single, non-fading path. Each condition shall be measured for at least 15 minutes for 1.0E-5 BER using the shortest interleaver setting. Table C.6.1 -1 below shows the Signal to Noise Ratio (SNR), measured in 100 kHz, for which the modem must achieve a bit error rate of 1e-5 or better.

Table C.6.1 -1: Minimum Performance Specifications for Achieving a Bit Error Rate of 1e-5 in an AWGN Channel

Modem Data Rate (kbps)	SNR (dB in 100 kHz) for 1e-5 BER
32	4
64	7
128	12
192	14
256	17
320	20
384	22

C.6.2 Doppler Shift Test

The modem shall acquire and maintain synchronisation for at least 5 minutes with test signals with high SNR and a Doppler shifts of +800 Hz and –800 Hz.

C.6.3 Doppler Sweep Performance

The AWGN BER test at the highest rate shall be repeated with a test signal having a frequency offset that continuously varies at a rate of 40 Hz/s between the limits of +800 Hz and -800 Hz, such that a plot of frequency offset versus time describes a periodic “triangle” waveform having a period of 160 (3200/40) seconds. Over a test duration of 1 hour, the modem shall achieve a BER of 1.0E-5 or less at a high SNR.

ANNEX D (OPTIONAL) TECHNICAL SPECIFICATIONS TO ENSURE INTEROPERABILITY OF SERIAL WAVEFORMS FOR 300 KHZ BANDWIDTH LINE OF SIGHT VHF AND UHF RADIO DATA COMMUNICATIONS

D.1. Introduction

D.1.1 Purpose

The purpose of this waveform specification is to define the requirements for a 112.5 kbps to 1152 kbps V/UHF modem waveform and provide a detailed description of modem functions to ensure interoperability within complying VHF and UHF radio networks.

D.1.2 Approach and Structure of this Document

The waveforms described in this document were designed to satisfy NATO requirements. This document specifies waveforms for general applications, including broadcast, Automatic Repeat reQuest (ARQ) and time-division multiple access (TDMA) operation.

D.1.3 Overview

This section presents a modem waveform and coding specification for data rates of 112.5, 225, 450, 600, 768, 960, and 1152 kilobits per second (kbps) operating within channel bandwidths of 300 kHz. This self-identifying waveform family is required for –112.5 - 1152 kbps operation for systems complying with this document.

A block interleaver shall be used to obtain four interleaving lengths for each data rate, referred to as Very Short, Short, Long, and Very Long in this document. For 112.5, 225, 450, and 600 kbps the Very Short interleaver is 1.6 ms; all other data rates have a Very Short interleaver of 2.5 ms. For 112.5, 225, 450, and 600 kbps the Short interleaver is 8 ms; all other data rates have a Short interleaver of 10 ms. For all data rates, the Long, and Very Long interleaver lengths are 80, and 640 milliseconds (ms) respectively. A rate ½, a constraint length 7, convolutional code is used. For data rates of 768, 960, and 1152 kbps, the code is punctured to rate $^{5}/_{6}$. For rates of 450 and 600 kbps, the code is punctured to rate ⅔. A full-tail-biting approach, with the block size equal to the interleaver length, is used.

The waveform uses Phase Shift Keying (PSK) and Quadrature Amplitude Modulation (QAM).

The data rate and interleaver settings are explicitly transmitted as a part of the waveform, both as part of the initial preamble and then periodically as both a reinserted preamble and in the periodic known symbol blocks. The receive modem is required to be able to deduce the data rate and interleaver setting either from the preamble or from the reinserted preamble portion of the waveform. A summary of the waveform characteristics is provided in tabular form in Table B.1.3 -1 below.

Table B.1.3 -1: Summary of Waveform Characteristics

Data Rate	Modulation	Block Size		Code Rate	Interleaver Settings (ms)	Reinserted Preamble Size (symbols)
		Data	Training			
112.5	BPSK	360	23	½	1.6, 8, 80, 640	123
225	QPSK			½		
450	8PSK			⅔		
600	16QAM			⅔		
768	16QAM	576	23	5/6	2.5, 10, 80, 640	87
960	32QAM			5/6		
1152	64QAM			5/6		

D.2. Modulation

The symbol rate for all symbols shall be 240000 symbols-per-second, which shall be accurate to a minimum of ±2.4 symbols-per-second (10 parts per million (ppm)) when the transmit data clock is generated by the modem and not provided by the data terminal equipment (DTE). Phase-shift keying (PSK) and quadrature amplitude modulation (QAM) modulation techniques shall be used. The sub-carrier (or pair of quadrature sub-carriers in the case of QAM) shall be centered at f_c accurate to a minimum of 0.1 ppm. The phase of the Quadrature sub-carrier relative to the In-phase carrier shall be 90 degrees. The correct relationship can be achieved by making the In-phase sub-carrier cos(f_c) and the Quadrature sub-carrier -sin(f_c).

The power spectral density of the modulator output signal should be constrained to be at least 20 decibels (dB) below the signal level measured at f_c, when tested outside of the band from f_c -150 kHz to f_c +150 kHz. The measured ripple shall be no more than ±2 dB in the range from f_c -100 kHz to f_c +100 kHz. The pulse shaping filter which shall be employed is a root-raised cosine filter, specified by the T/2 (480 kHz sampling rate) spaced coefficients of the finite impulse response (FIR) filter:

0.021220659078919377,
-0.018296424333794137,
-0.037513179839879389,
0.065296144145032972,
0.053051647697298421,
-0.170297633845998941,
-0.064237155776998608,
0.621797410509131954,
1.068309886183790969,

0.621797410509131954,
-0.064237155776998608,
-0.170297633845998941,
0.053051647697298421,
0.065296144145032972,
-0.037513179839879389,
-0.018296424333794137,
0.021220659078919377

D.2.1 Known Symbols

For all known symbols, which are defined as those symbols which are known to the receiver prior to reception, the modulation used shall be PSK, with the symbol mapping shown in Table B.2.1 -1 and Figure B.2.1 -1. No scrambling shall be applied to the known symbols.

Table B.2.1 -1: 8PSK Symbol Mapping

Symbol Number	Phase	In-Phase	Quadrature
0	0	1.000000	0.000000
1	$\pi/4$	0.707107	0.707107
2	$\pi/2$	0.000000	1.000000
3	$3\pi/4$	-0.707107	0.707107
4	π	-1.000000	0.000000
5	$5\pi/4$	-0.707107	-0.707107
6	$3\pi/2$	0.0000000	-1.000000
7	$7\pi/4$	0.707107	-0.707107

Note that the complex symbol values = exp[jnπ/4] where n is the symbol number.

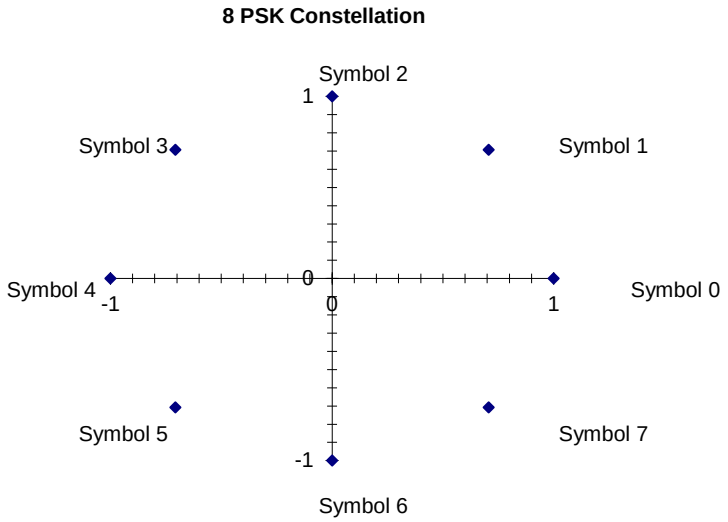


Figure B.2.1 -1: 8PSK Signal Constellation and Symbol Mapping

D.2.2 Data Symbols

For data symbols, the modulation used shall depend upon the data rate. Table D.2.2 -1 specifies the modulation that shall be used with each data rate.

Table D.2.2 -1: Modulation Used to Obtain Each Data Rate

Data Rate (kbps)	Modulation
112.5	BPSK
225	QPSK
450	8PSK
600	16QAM
768	16QAM
960	32QAM
1152	64QAM

All PSK constellations are scrambled to appear, on-air, as an 8PSK constellation. The QAM constellations have been modified from conventional rectangular constellations to improve the peak-to-average ratio.

D.2.2.1 PSK Data Symbols

For the PSK constellations, a distinction is made between the data bits and the symbol number for the purposes of scrambling QPSK modulation to appear as 8PSK, on-air. Scrambling is applied as a modulo 8 addition of a scrambling sequence to the 8PSK

symbol number. Transcoding is an operation which links a symbol to be transmitted to a group of data bits.

D.2.2.1.1 BPSK Symbol Mapping

For the 112.5 kbps user data rate, transcoding shall be achieved by linking one of the symbols specified in Table B.2.1 -1 to a set of two consecutive data bits (dibit) as shown in Table D.2.2.1.1 -1.

Table D.2.2.1.1 -1: Transcoding for 112.5 kbps

Bit	Symbol
0	0
1	4

D.2.2.1.2 QPSK Symbol Mapping

For the 225 kbps user data rate, transcoding shall be achieved by linking one of the symbols specified in Table B.2.1 -1 to a set of two consecutive data bits (dibit) as shown in Table D.2.2.1.2 -1. In this table, the leftmost bit of the dibit shall be the older bit; i.e., fetched from the interleaver before the rightmost bit.

Table D.2.2.1.2 -1: Transcoding for 225 kbps

Dibit	Symbol
00	0
01	2
11	4
10	6

D.2.2.1.3 8PSK Symbol Mapping

For the 450 kbps user data rate transcoding shall be achieved by linking one symbol to a set of three consecutive data bits (tribit) as shown in Table D.2.2.1.3 -1. In this table, the leftmost bit of the tribit shall be the oldest bit; i.e., fetched from the interleaver before the other two and the rightmost bit is the most recent bit.

Table D.2.2.1.3 -1: Transcoding for 450 kbps

Tribit	Symbol
000	1
001	0
010	2
011	3
100	6
101	7
110	5
111	4

D.2.2.1.4 QAM Data Symbols

For the QAM constellations, no distinction is made between the number formed directly from the data bits and the symbol number. Each set of 4 bits (16QAM), 5 bits (32QAM) or 6 bits (64QAM) is mapped directly to a QAM symbol. For example, the four bit grouping 0111 would map to symbol 7 in the 16QAM constellation while the 6 bit grouping 100011 would map to symbol 35 in the 64QAM constellation. Again, in each case the leftmost bit shall be the oldest bit, i.e. fetched from the interleaver before the other bits, and the rightmost bit is the most recent bit.

The mapping of bits to symbols for the QAM constellations has been selected to minimize the number of bit errors incurred when errors involve adjacent signaling points in the constellation.

D.2.2.1.5 The 16QAM Constellation

The constellation points which shall be used for 16QAM are shown in Figure D.2.2.1.5 -1 and specified in terms of their In-phase and Quadrature components in Table D.2.2.1.5 -1.

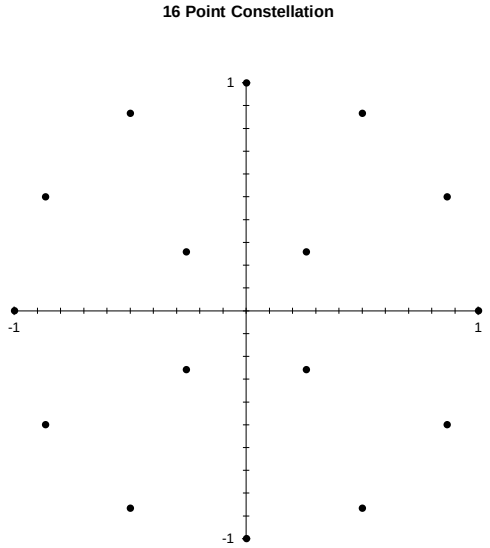


Figure D.2.2.1.5 -1: 16QAM Signalling Constellation

Table D.2.2.1.5 -1: In-phase and Quadrature Components of Each 16QAM Symbol

Symbol Number	In-Phase	Quadrature
0	0.866025	0.500000
1	0.500000	0.866025
2	1.000000	0.000000
3	0.258819	0.258819
4	-0.500000	0.866025
5	0.000000	1.000000
6	-0.866025	0.500000
7	-0.258819	0.258819
8	0.500000	-0.866025
9	0.000000	-1.000000
10	0.866025	-0.500000
11	0.258819	-0.258819
12	-0.866025	-0.500000
13	-0.500000	-0.866025
14	-1.000000	0.000000
15	-0.258819	-0.258819

D.2.2.1.6 The 32QAM Constellation

The constellation points which shall be used for 32QAM are shown in Figure D.2.2.1.6 -1 and specified in terms of their In-phase and Quadrature components in Table D.2.2.1.6 -1.

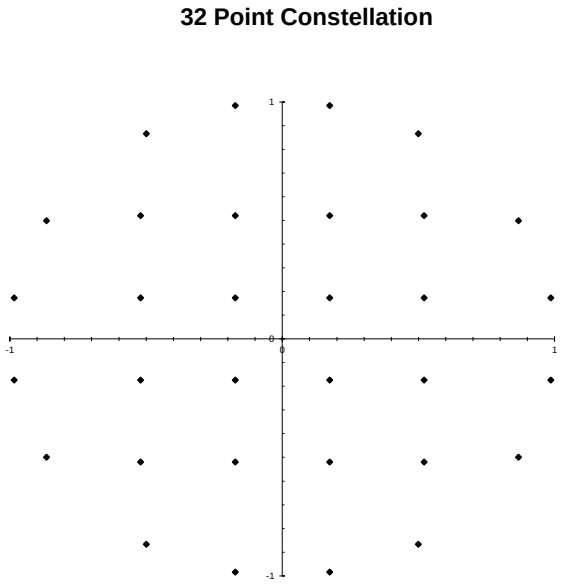


Figure D.2.2.1.6 -1: 32QAM Signalling Constellation

Table D.2.2.1.6 -1: In-phase and Quadrature Components of Each 32QAM Symbol

Symbol Number	In-Phase	Quadrature	Symbol Number	In-Phase	Quadrature
0	0.866380	0.499386	16	0.866380	-0.499386
1	0.984849	0.173415	17	0.984849	-0.173415
2	0.499386	0.866380	18	0.499386	-0.866380
3	0.173415	0.984849	19	0.173415	-0.984849
4	0.520246	0.520246	20	0.520246	-0.520246
5	0.520246	0.173415	21	0.520246	-0.173415
6	0.173415	0.520246	22	0.173415	-0.520246
7	0.173415	0.173415	23	0.173415	-0.173415
8	-0.866380	0.499386	24	-0.866380	-0.499386
9	-0.984849	0.173415	25	-0.984849	-0.173415
10	-0.499386	0.866380	26	-0.499386	-0.866380
11	-0.173415	0.984849	27	-0.173415	-0.984849
12	-0.520246	0.520246	28	-0.520246	-0.520246

13	-0.520246	0.173415	29	-0.520246	-0.173415
14	-0.173415	0.520246	30	-0.173415	-0.520246
15	-0.173415	0.173415	31	-0.173415	-0.173415

D.2.2.1.7 The 64QAM Constellation

The constellation points which shall be used for the 64QAM modulation are shown in Figure D.2.2.1.7 -1 and specified in terms of their In-phase and Quadrature components in Table D.2.2.1.7 -1.

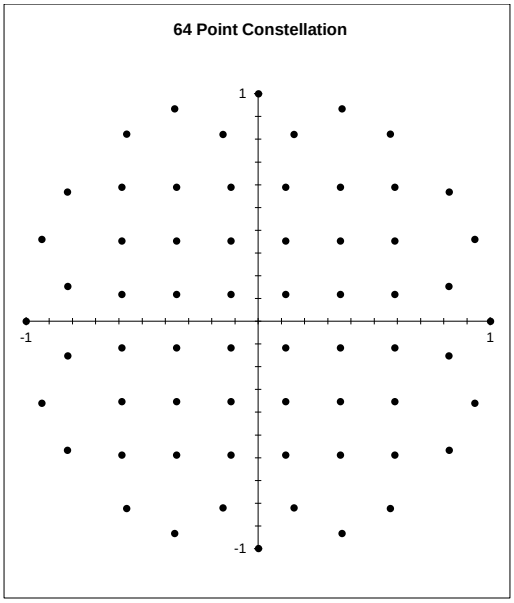


Figure D.2.2.1.7 -1: 64QAM Signalling Constellation

Table D.2.2.1.7 -1: In-phase and Quadrature Components of Each 64QAM Symbol

Symbol Number	In-Phase	Quadrature	Symbol Number	In-Phase	Quadrature
0	1.000000	0.000000	32	0.000000	1.000000
1	0.822878	0.568218	33	-0.822878	0.568218
2	0.821137	0.152996	34	-0.821137	0.152996
3	0.932897	0.360142	35	-0.932897	0.360142
4	0.000000	-1.000000	36	-1.000000	0.000000
5	0.822878	-0.568218	37	-0.822878	-0.568218
6	0.821137	-0.152996	38	-0.821137	-0.152996
7	0.932897	-0.360142	39	-0.932897	-0.360142
8	0.568218	0.822878	40	-0.568218	0.822878
9	0.588429	0.588429	41	-0.588429	0.588429
10	0.588429	0.117686	42	-0.588429	0.117686
11	0.588429	0.353057	43	-0.588429	0.353057
12	0.568218	-0.822878	44	-0.568218	-0.822878
13	0.588429	-0.588429	45	-0.588429	-0.588429
14	0.588429	-0.117686	46	-0.588429	-0.117686
15	0.588429	-0.353057	47	-0.588429	-0.353057
16	0.152996	0.821137	48	-0.152996	0.821137
17	0.117686	0.588429	49	-0.117686	0.588429
18	0.117686	0.117686	50	-0.117686	0.117686
19	0.117686	0.353057	51	-0.117686	0.353057
20	0.152996	-0.821137	52	-0.152996	-0.821137
21	0.117686	-0.588429	53	-0.117686	-0.588429
22	0.117686	-0.117686	54	-0.117686	-0.117686
23	0.117686	-0.353057	55	-0.117686	-0.353057
24	0.360142	0.932897	56	-0.360142	0.932897
25	0.353057	0.588429	57	-0.353057	0.588429
26	0.353057	0.117686	58	-0.353057	0.117686
27	0.353057	0.353057	59	-0.353057	0.353057
28	0.360142	-0.932897	60	-0.360142	-0.932897
29	0.353057	-0.588429	61	-0.353057	-0.588429
30	0.353057	-0.117686	62	-0.353057	-0.117686
31	0.353057	-0.353057	63	-0.353057	-0.353057

D.2.3 Data Scrambling

Data symbols for the 8PSK symbol constellation (112.5 kbps, 225 kbps, and 450 kbps) shall be scrambled by modulo 8 addition with a scrambling sequence. The data symbols

for the 16QAM, 32QAM and 64QAM constellations 600 kbps, 768 kbps, 960 kbps, and 1152 kbps) shall be scrambled by using an exclusive or (XOR) operation. Sequentially, the data bits forming each symbol (4 for 16QAM, 5 for 32QAM, and 6 for 64QAM) shall be XOR'd with an equal number of bits from the scrambling sequence. The scrambling sequence generator polynomial shall be $x^9 + x^4 + 1$ and the generator shall be initialized to 1 at the start of each data frame. A block diagram of the scrambling sequence generator is shown in Figure D.2.3 -1.

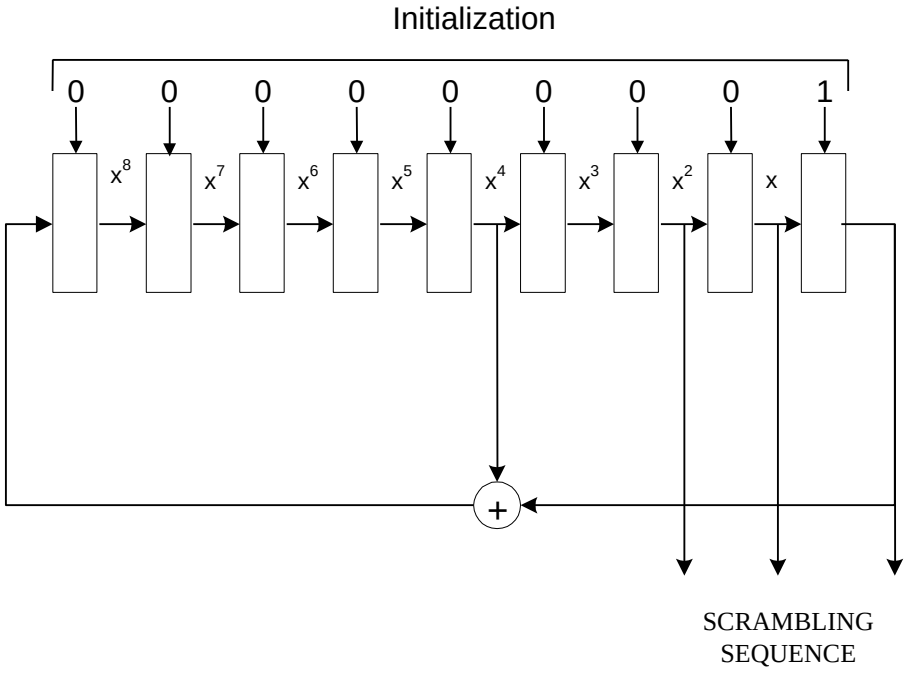


Figure D.2.3 -1: Scrambling Sequence Generator Illustrating Scrambling Generator for 8PSK Symbols

For 8PSK symbols (112.5, 225, and 450 kbps), the scrambling shall be carried out taking the modulo 8 sum of the numerical value of the binary triplet consisting of the last (rightmost) three bits in the shift register, and the symbol number (transcoded value). For example, if the last three bits in the scrambling sequence shift register were 010 which has a numerical value equal 2, and the symbol number before scrambling was 6, symbol 0 would be transmitted since: (6+2) Modulo 8 = 0. For 16QAM symbols, scrambling shall be carried out by XORing the 4 bit number consisting of the last (rightmost) four bits in the shift register with the symbol number. For example, if the last 4 bits in the scrambling sequence shift register were 0101 and the 16QAM symbol number before scrambling was 3 (i.e. 0011), symbol 6 (0110) would be transmitted. For 32QAM symbols, scrambling

shall be carried out by XORing the 5 bit number formed by the last (rightmost) five bits in the shift register with the symbol number. For 64QAM symbols, scrambling shall be carried out by XORing the 6 bit number formed by the last (rightmost) six bits in the shift register with the symbol number.
After each data symbol is scrambled, the generator shall be iterated (shifted) the required number of times to produce all new bits for use in scrambling the next symbol (i.e., 3 iterations for 8PSK, 4 iterations for 16QAM, 5 iterations for 32QAM and 6 iterations for 64QAM). Since the generator is iterated after the bits are used, the first data symbol of every data frame shall, therefore, be scrambled by the appropriate number of bits from the initialization value of 00000001.

The length of the scrambling sequence is 511 bits. For a 576 symbol data block with 6 bits per symbol, this means that the scrambling sequence will be repeated just slightly less than 7 times.

D.3. Frame structure

The frame structure that shall be used for all data rates is described in the sections below.

D.3.1 Synchronisation and Reinserted Preambles

The synchronisation preamble is used for rapid initial synchronisation. The reinserted preamble is used to facilitate acquisition of an ongoing transmission (acquisition on data).

D.3.2 Frank-Heimiller and Barker Sequences

All data rate frames shall use combinations of Frank-Heimiller (length 16) and Barker (length 11) sequences to form the preamble portions of the transmission. The Barker sequence shall be used to encode the data rate and interleaver properties of the transmission and to mark interleaver boundary points.

The Frank-Heimiller sequence used for this waveform is expressed as a series of 8PSK symbols as follows:

0, 0, 0, 0, 0, 2, 4, 6, 0, 4, 0, 4, 0, 6, 4, 2

Depending on the particular data rate, each preamble shall use a cyclic repetition of a portion of this sequence.

The 11 symbol Barker sequence shall be used to encode one or two bits of data for each sequence of 11 8PSK symbols. The base Barker sequence, expressed as a series of 8PSK symbols is shown below. When the Barker sequence is phase modulated all 11 symbols are phase shifted by the same degree. Both BPSK and QPSK modulations are employed to encode data on Barker sequences within this specification.

4, 0, 4, 4, 0, 4, 4, 4, 0, 0, 0

The sections below detail the 8PSK symbols that comprise each Barker sequence as required for particular encoded data value

D.3.3 112.5, 225, 450, and 600 kbps Data Rates

The initial 600 symbol preamble shall be followed by 100 frames of alternating data and known symbols. Each data frame shall consist of a data block followed by a mini-probe consisting of 23 symbols of known data. The data block shall consist of 360 data symbols.

After 100 data frames, the reinserted preamble shall be transmitted, with a length of 123 symbols. This is done to facilitate late acquisition, Doppler shift removal, and sync adjustment.

It should be noted that the last data block in of the 100 data frames is not followed by a 23 symbol mini-probe segment. Instead, the known data portion of the 123 symbol preamble immediately follows this final data block.

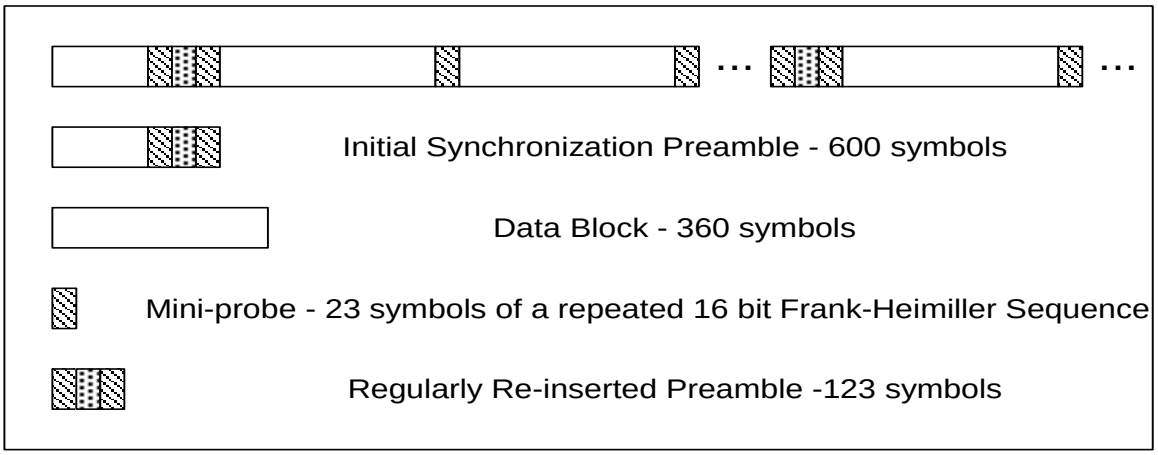


Figure D.3.3 -1: Frame Structure for Data Rates of 180, 360, 720 and 960 kbps

D.3.3.1 Synchronization Preamble

The mandatory portion of the preamble, which shall always be transmitted, and shall always appear at the beginning of a transmission, shall consist of 600 symbols. The first 477 symbols are intended exclusively for synchronisation and Doppler offset removal purposes while the final 123 symbols, which are common with the reinserted preamble, also carry information regarding the data rate and interleaver settings. Expressed as a

sequence of symbols, using the symbol numbers given in Table D.2.1 -1 the second section of the synchronisation preamble shall be as follows:

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

The remaining 123 symbols of the synchronization preamble shall be the same as the reinserted preamble, as defined in the section below.

D.3.3.2 Reinserted Preamble

The reinserted preamble shall be defined for the 112.5, 225, 450, and 600 kbps data rates using a series of Frank-Heimiller and Barker sequences. The interleaver length and data rate shall be encoded in the preamble using the Barker sequences, as shown in Table D.3.3.2 -1.

Table D.3.3.2 -1: Reinserted Preamble: 112.5, 225, 450, and 600 kbps

Number of Symbols	Sequence	Data
24	Frank	0, 0, 0, 0, 0, 2, 4, 6, 0, 4, 0, 4, 0, 6, 4, 2, 0, 0, 0, 0, 0, 2, 4, 6
11	Barker	112.5 or 450 kbps: 0, 4, 0, 0, 4, 0, 0, 0, 4, 4, 4

		225 or 600 kbps: 4, 0, 4, 4, 0, 4, 4, 4, 0, 0, 0
11	Barker	112.5 or 225 kbps: 0, 4, 0, 0, 4, 0, 0, 0, 4, 4, 4 450 or 600 kbps: 4, 0, 4, 4, 0, 4, 4, 4, 0, 0, 0
20	Frank	0, 0, 0, 0, 0, 2, 4, 6, 0, 4, 0, 4, 0, 6, 4, 2, 0, 0, 0, 0
11	Barker	Interleaver boundary: 0, 4, 0, 0, 4, 0, 0, 0, 4, 4, 4 Not interleaver boundary: 4, 0, 4, 4, 0, 4, 4, 4, 0, 0, 0
11	Barker	Very Short or Long interleaver: 0, 4, 0, 0, 4, 0, 0, 0, 4, 4, 4 Short or Very Long interleaver: 4, 0, 4, 4, 0, 4, 4, 4, 0, 0, 0
11	Barker	Very Short or Short interleaver: 0, 4, 0, 0, 4, 0, 0, 0, 4, 4, 4 Long or Very Long interleaver: 4, 0, 4, 4, 0, 4, 4, 4, 0, 0, 0
24	Frank	0, 0, 0, 0, 0, 2, 4, 6, 0, 4, 0, 4, 0, 6, 4, 2, 0, 0, 0, 0, 0, 2, 4, 6

There shall be a total of 123 symbols formed using these sequences.

D.3.4 768, 960, and 1152 kbps Data Rates

The initial 600 symbol preamble shall be followed by 64 frames of alternating data and known symbols. Each data frame shall consist of a data block followed by a mini-probe consisting of 23 symbols of known data. The data block shall consist of 576 data symbols.

After 64 data frames, the reinserted preamble shall be transmitted, with a length of 87 symbols. This is done to facilitate late acquisition, Doppler shift removal, and sync adjustment.

It should be noted that the last data block in of the 64 data frames is not followed by a 23 symbol mini-probe segment. Instead, the known data portion of the 87 symbol preamble immediately follows this final data block.

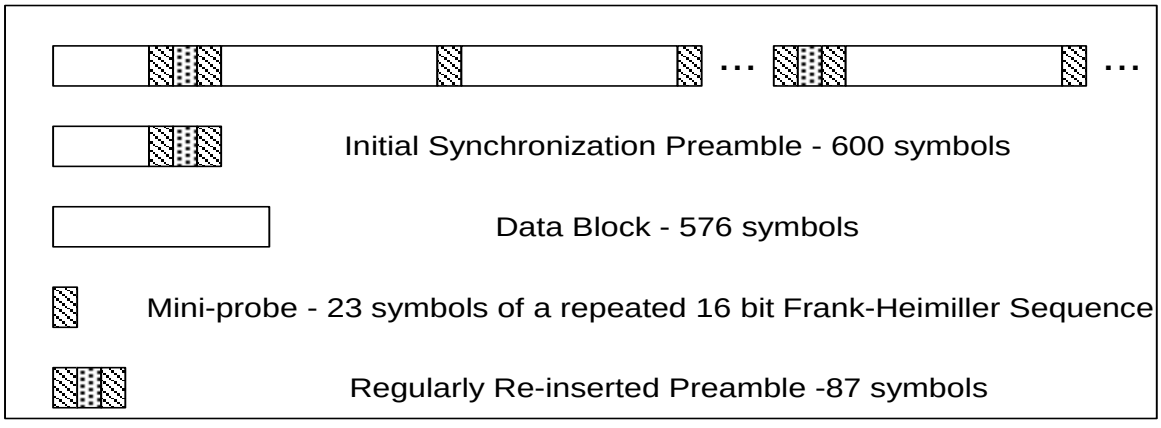


Figure D.3.4 -1: Frame Structure for Data Rates of 768, 960, and 1152 kbps

D.3.4.1 Synchronization Preamble

The mandatory portion of the preamble, which shall always be transmitted, and shall always appear at the beginning of an interleaver boundary, shall consist of 600 symbols. The first 513 symbols are intended exclusively for synchronisation and Doppler offset removal purposes while the final 87 symbols, which are common with the reinserted preamble, also carry information regarding the data rate and interleaver settings. Expressed as a sequence of symbols, using the symbol numbers given in Table D.2.1 -1 the second section of the synchronisation preamble shall be as follows:

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

It should be noted that the last 417 symbols of this pseudo random sequence are common with the 417 symbols of the pseudo random sequence employed for data rates less than 768 kbps. The remaining symbols of the synchronization preamble shall be the same as the reinserted preamble, as defined in the section below.

D.3.4.2 Reinserted Preamble

The reinserted preamble shall be defined for the 768, 960, and 1152 kbps data rates using a series of Frank-Heimiller and Barker sequences. The interleaver length and data rate shall be encoded in the preamble using the Barker sequences, as shown in Table D.3.4.2 -1.

Table D.3.4.2 -1: Reinserted Preamble: 768, 960, 1152 kbps

Number of Symbols	Sequence	Data

27	Frank	0, 0, 0, 0, 0, 2, 4, 6, 0, 4, 0, 4, 0, 6, 4, 2, 0, 0, 0, 0, 0, 2, 4, 6, 0, 4, 0
11	Barker	768 kbps: 4, 0, 4, 4, 0, 4, 4, 4, 0, 0, 0 960 kbps: 6, 2, 6, 6, 2, 6, 6, 6, 2, 2, 2 1152 kbps: 2, 6, 2, 2, 6, 2, 2, 2, 6, 6, 6
11	Barker	Interleaver boundary: 4, 0, 4, 0, 4, 4, 0, 0, 4, 4, 4, 4, 4 Not interleaver boundary: 4, 0, 4, 4, 0, 4, 4, 4, 0, 0, 0
11	Barker	Very Short Interleaver: 4, 0, 4, 4, 0, 4, 4, 4, 0, 0, 0 Short Interleaver: 6, 2, 6, 6, 2, 6, 6, 6, 2, 2, 2 Long Interleaver: 2, 6, 2, 2, 6, 2, 2, 2, 6, 6, 6 Very Long Interleaver: 0, 4, 0, 0, 4, 0, 0, 0, 4, 4, 4
27	Frank	0, 0, 0, 0, 0, 2, 4, 6, 0, 4, 0, 4, 0, 6, 4, 2, 0, 0, 0, 0, 0, 2, 4, 6, 0, 4, 0

There is a total of 87 symbols formed using these sequences.

D.3.5 Mini-Probes

For all data rates, mini-probes 23 symbols in length shall be inserted following every data block, with the exception of the last data block prior to the reinserted preamble. Using the 8PSK symbol mapping, each mini-probe shall be based on a 16 bit Frank-Heimiller sequence. The sequence that shall be used, specified in terms of the 8PSK symbol numbers, is given by:

0, 0, 0, 0, 0, 2, 4, 6, 0, 4, 0, 4, 0, 6, 4, 2, 0, 0, 0, 0, 0, 2, 4

D.4. Coding and interleaving

The interleaver used shall be a block interleaver. Each block of input data shall also be encoded using a block encoding technique with a code block size equal to the size of the block interleaver. Thus, the input data bits will be sent as successive blocks of bits that span the duration of the interleaver length selected.

Table D.4-1 shows the interleaver length in milliseconds for each combination of data rate and interleaver type (very short, short, long, and very long).

Table D.4-1: Interleaver Length in Milliseconds as a Function of Data Rate and

Interleaver Type

Data Rate (kbps)	Interleaver Length			
	Very Short	Short	Long	Very Long
	Input Data Block Size (ms)			
112.5	1.6	8	80	640
225	1.6	8	80	640
450	1.6	8	80	640
600	1.6	8	80	640
768	2.5	10	80	640
960	2.5	10	80	640
1152	2.5	10	80	640

Table D.4-2 shows the number of input data bits per block as function of both data rate and interleaver length. Note that an “input data block” should not be confused with the 360 or 576 symbol data block that is part of a data frame in the waveform format. The bits from an input data block, when expanded by the application of the forward error correction coding, are sufficient to fill the specified interleaver.

Table D.4-2: Input Data Block Size in Bits as a Function of Data Rate and Interleaver Length

Data Rate (kbps)	Interleaver Length			
	Very Short	Short	Long	Very Long
	Input Data Block Size (bits)			
112.5	180	900	9000	72000
225	360	1800	18000	144000
450	720	3600	36000	288000
600	960	4800	48000	384000
768	1920	7680	61440	491520
960	2400	9600	76800	614400

1152	2880	11520	92160	737280

D.4.1 Block Boundary Alignment

Each code block shall be interleaved within a single block interleaver of the same size. The boundaries of these blocks shall be aligned such that the beginning of the first data frame following each reinserted preamble shall coincide with an interleaver boundary. Thus for an interleaver length of 8 frames, the first eight data frames following a reinserted preamble will contain all of the encoded bits for a single input data block. The first data symbol from the first data frame in each interleaver set shall have as its most significant bit (MSB) the first bit fetched from the interleaver.

D.4.2 Block Encoding

The full-tail-biting and puncturing techniques shall be used with a rate ½ convolutional code to produce a block code that is the same length as the interleaver. For those data rates where the code is punctured to rate $^{5}/_{6}$ or rate ⅔, the punctured block shall still fit exactly within the interleaver.

D.4.3 Rate ½ Convolutional Code

A constraint length 7, rate 1/2 convolutional code shall be used prior to puncturing. Figure D.4.3 -1 is a pictorial representation of the encoder. The two generator polynomials used shall be:

$T_1 = x^6 + x^4 + x^3 + x + 1$
$T_2 = x^6 + x^5 + x^4 + x^3 + 1$

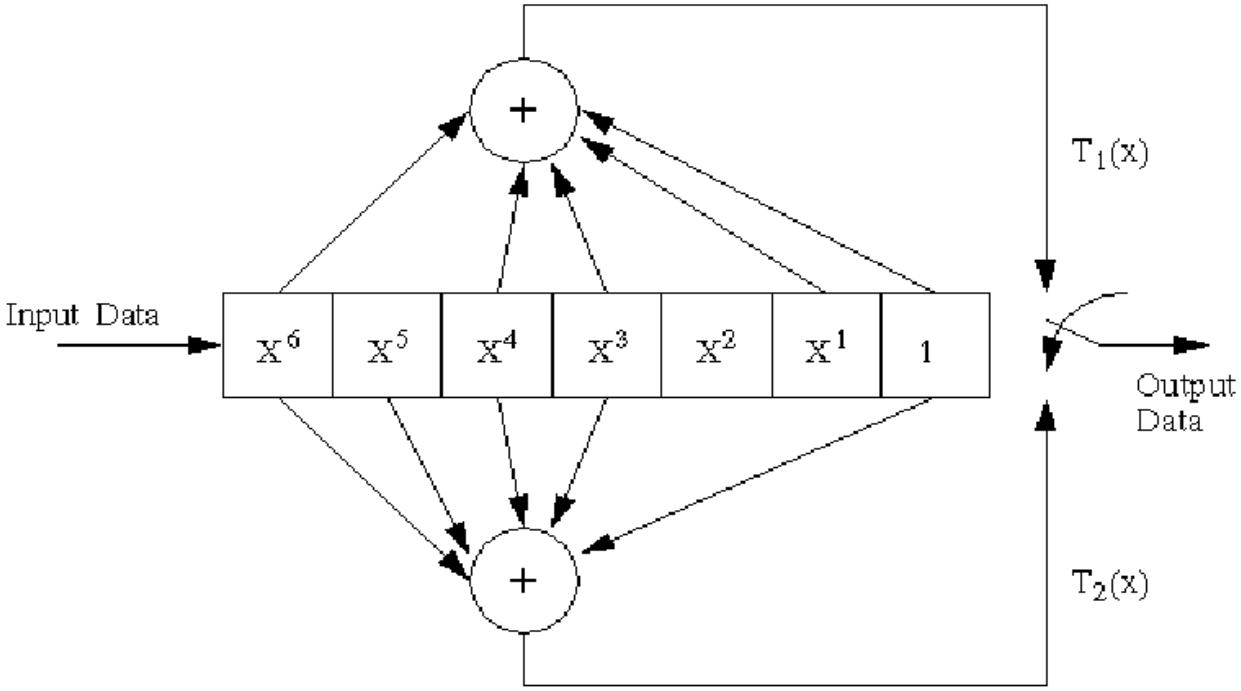


Figure D.4.3 -1: Constraint Length 7, Rate ½ Convolutional Encoder

The two summing nodes in the figure represent modulo 2 addition. For each bit input to the encoder, two bits are taken from the encoder, with the upper output bit, $T_1(x)$, taken first.

D.4.4 Full-Tail-Biting Encoding

To begin encoding each block of input data, the encoder shall be preloaded by shifting in the first eight input data bits without taking any output bits. These eight input bits shall be temporarily saved so that they can be used to "flush" the encoder. The first two coded output bits shall be taken after the ninth bit has been shifted in, and shall be defined to be the first two bits of the resulting block code. After the last input data bit has been encoded, the first eight "saved" data bits shall be encoded. Note that the encoder shift register should not be changed before encoding these saved bits; i.e., it should be filled with the last nine input data bits. The eight "saved" data bits are encoded by shifting them into the encoder one at a time, beginning with the earliest of the eight. The encoding thus continues by taking the two resulting coded output bits as each of the saved eight bits is

shifted in. These encoded bits shall be the final bits of the resulting (unpunctured) block code. Prior to puncturing, the resulting block code will have exactly twice as many bits as the input information bits. Puncturing of the rate ½ code to the required rate $^5/_6$ shall be done prior to sending bits to the interleaver.

D.4.5 Puncturing to Rate $^5/_6$

In order to obtain a rate $^5/_6$ code from the rate ½ code used, the output of the encoder must be punctured by not transmitting 4 bits out of every 10. Puncturing shall be performed by using a puncturing mask of 1 1 1 0 0 1 1 0 0 1, applied to the bits output from the encoder. In this notation a 1 indicates that the bit is retained and a 0 indicates that the bit is not transmitted. For an encoder generated sequence of:

T1(k), T2(k), T1(k+1), T2(k+1), T1(k+2), T2(k+2) . . .

the transmitted sequence shall be:

T1(k), T2(k), T1(k+1), T2(k+2) . . .

Defining T1(0), T2(0) to be the first two bits of the block code generated as defined in Paragraph D.4.2 , then the value of k in the above sequences shall be an integral multiple of 5. The block code shall be punctured in this manner before being input to the interleaver.

D.4.6 Puncturing to Rate ⅔

In order to obtain a rate ⅔ code from the rate ½ code used, the output of the encoder must be punctured by not transmitting 1 bit out of every 4. Puncturing shall be performed by using a puncturing mask of 1 1 1 0, applied to the bits output from the encoder. In this notation a 1 indicates that the bit is retained and a 0 indicates that the bit is not transmitted. For an encoder generated sequence of:

T1(k), T2(k), T1(k+1), T2(k+1), T1(k+2), . . .

the transmitted sequence shall be:

T1(k), T2(k), T1(k+1), T1(k+2), . . .

Defining T1(0), T2(0) to be the first two bits of the block code generated as defined in Paragraph 4.2, then the value of k in the above sequences shall be an integral multiple of 2. The block code shall be punctured in this manner before being input to the interleaver.

D.4.7 Block Interleaver Structure

The block interleaver used is designed to separate neighbouring bits in the punctured block code as far as possible over the span of the interleaver with the largest separations resulting for the bits that were originally closest to each other. Because of the 28 different combinations of data rates and interleaver lengths, a flexible interleaver structure is needed.

D.4.7.1 Interleaver Size in Bits

The interleaver shall consist of a single dimension array, numbered from 0 to its size in bits –1. The array size shall depend on both the data rate and interleaver length selected as shown in Table D.4.7.1 -1.

Table D.4.7.1 -1: Interleaver Size in Bits as a Function of Data Rate and Interleaver Length

Data Rate (kbps)	Interleaver Length			
	Very Short	Short	Long	Very Long
	Interleaver Size in Bits			
112.5	360	1800	18000	144000
225	720	3600	36000	288000
450	1080	5400	54000	432000
600	1440	7200	72000	576000
768	2304	9216	73728	589824
960	2880	11520	92160	737280
1152	3456	13824	110592	884736

D.4.7.2 Interleaver Load

The punctured block code bits shall be loaded into the interleaver array beginning with location 0. The location for loading each successive bit shall be obtained from the previous location by incrementing by the “Interleaver Increment Value” specified in Table D.4.7.2 -1, modulo the “Interleaver Size in Bits.”

Defining the first punctured block code bit to be B(0), then the load location for B(n) is given by:

Load Location = (n * Interleaver Increment Value) Modulo (Interleaver Size in Bits)

Thus for 112.5 kbps, with Very Short interleaver (108 bit size with an increment of 41), the first 8 interleaver load locations are: 0, 41, 82, 15, 56, 97, 30, and 71.

Table D.4.7.2 -1: Interleaver Increment Value as a Function of Data Rate and Interleaver Length

Data Rate (kbps)	Interleaver Length			
	Very Short	Short	Long	Very Long
	Interleaver Increment Value			
112.5	67	289	2681	19303
225	109	1307	5363	38557
450	157	1501	8221	60031
600	217	1997	13789	77509
768	337	2701	1315	135565
960	451	3331	36271	168101
1152	505	4867	43471	185095

D.4.7.3 Interleaver Fetch

The fetching sequence for all data rates and interleaver lengths shall start with location 0 of the interleaver array and increment the fetch location by 1. This is a simple linear fetch from beginning to end of the interleaver array.

D.5. Operational features and message protocols

D.5.1 User Interfaces

D.5.1.1 Synchronous Data Communications Equipment (DCE) Interface

The modem shall implement a synchronous DCE interface that is compatible with EIA-232, EIA-422 or Mil-Std-188-114A.

D.5.1.2 Conventional Asynchronous Interface (Optional)

The modem may be capable of interfacing with an asynchronous DTE; this is not required behaviour. Conventional RS-232 asynchronous serial interfaces will not support all data rates defined by this standard.

D.5.1.3 Ethernet Interface (Optional)

The modem shall provide an Ethernet interface. The bytes shall be aligned with the Input Data Block boundary following an initial or reinserted preamble.

D.5.2 Onset of Transmission

The modem shall begin a transmission no later than 25 ms after it has received an entire input data block (enough bits to fill a coded and interleaved block), or upon receipt of the last input data bit, whichever occurs first. The latter would only occur when the message is shorter than one interleaver block. A transmission shall be defined as beginning with the keying of the radio, followed by the output of the preamble waveform after the configured pre-key delay, if any.

D.5.3 End of Message

The use of an end-of-message (EOM) in the transmit waveform shall be a configurable option. When the use of an EOM has been selected, a 32-bit EOM pattern shall be appended after the last input data bit of the message. The EOM, expressed in hexadecimal notation is 4B65A5B2, where the left most bit is sent first. If the last bit of the EOM does not fill out an input data block, the remaining bits in the input data block shall be set to zero before encoding and interleaving the block.

If the use of an EOM has been inhibited and the last input data bit does not fill out an input data block, the remaining bits in the input data block shall be set to zero before encoding and interleaving the block. It is anticipated that the use of an EOM will only be inhibited when an ARQ data protocol uses ARQ blocks which completely fill (or nearly so) the selected input data block size (interleaver block). Without this feature, the use of an EOM would require the transmission of an additional interleaver block under these circumstances.

D.5.4 Termination of a Transmission

Upon receipt of a radio silence (or equivalent) command, the modem shall immediately un-key the radio and terminate the transmit waveform.

In normal operation, the modem shall terminate a transmission only after the transmission of the final data frame, including a mini-probe, associated with the final interleaver block. Note that a data frame consists of a 360 or 576 symbol data block followed by a mini-probe. The entire final mini-probe shall be transmitted before the modem unkeys.

D.5.5 Termination of Receive Data Processing

There are a number of events which shall cause the modem to cease processing the received signal to recover data, and return to the acquisition mode. These are necessary because a modem is not able to acquire a new transmission while it is attempting to demodulate and decode data.

D.5.5.1 Detection of EOM

The modem shall always scan all of the decoded bits for the 32-bit EOM pattern defined in Paragraph B.5.3 . Upon detection of the EOM the modem shall return to the acquisition mode. The modem shall continue to deliver decoded bits to the user (DTE) until the final bit immediately preceding the EOM has been delivered.

D.5.5.2 Command to Return to Acquisition

Upon receipt of a command to terminate reception, the modem shall return to the acquisition mode and terminate the delivery of decoded bits to the user (DTE).

D.5.5.3 Receipt of a Specified Number of Data Blocks

The maximum message duration measured in number of Input Data Blocks (interleaver blocks) shall be a configurable parameter. Setting this parameter to zero shall specify that an unlimited number may be received. Once the modem has decoded and delivered to the user (DTE), the number of bits corresponding to the configured maximum message duration, the modem shall return to the acquisition mode and terminate the delivery of decoded bits to the user (DTE). Note that for a given interleaver length, this parameter also specifies the maximum message duration in time, independent of the bit rate. Note that this parameter is the maximum duration and that the transmit end always has the option of using an EOM for shorter transmissions.

Operation with a specified number of input data blocks may be used by an ARQ or TDMA protocol where the size of the packet is fixed, or occasionally changed to accommodate changing propagation conditions. In this case we anticipate that this parameter (maximum message duration) will be sent to the receiving end of the link as part of the ARQ protocol or specified in the TDMA protocols. It would then have to be sent to the receiving modem through the remote control interface or be preconfigured by an operator since it is not embedded in the waveform itself as the data rate and interleaver length parameters are.

D.5.5.4 Initiation of a Transmission

If, and only if, the Modem is configured to operate in half-duplex mode with transmit override, the initiation of a transmission by the user (DTE) shall cause the modem to terminate the receive processing and the delivery of decoded bits to the user (DTE).

D.5.6 TDMA Compatibility

D.5.6.1 Jitter

The total uncertainty in the time between the receipt of Request to Send (RTS) and the onset of the transmission (defined above) shall be less than 20 ms.

D.5.6.2 Minimum Time Between Back-to-Back Receptions

The modem shall be capable of receiving a subsequent transmission within 10 ms of the end of an initial transmission if the modem is configured to ignore the autobaud information in the waveform and the same data rate and interleaver settings are used for the initial and subsequent transmissions. If the modem is configured to accept autobaud data rate and interleaver settings, the minimum time between back to back receptions shall be 1.2 * the interleaver length.

D.5.6.3 Associated Communications (Radio) Equipment

The radio must provide a passband of at least 215 kHz with a ripple of not more than +/- 2 dB across the passband. The radio must support linear modulations and must support a signal to noise and distortion (SINAD) of not less than 30 dB for the 64 QAM modulation defined herein.

D.6. Performance Requirements

The minimum performance requirements for the high data rate mode are specified in this section. These requirements are not exhaustive, but are intended to ensure that equipment contains a high-quality implementation of the standard.

D.6.1 Bit Error Rate (BER) Performance

BER performance shall be measured with the channel simulator programmed to simulate an additive white Gaussian noise (AWGN) channel. The AWGN channel shall consist of a single, non-fading path. Each condition shall be measured for at least 15 minutes for 1.0E-5 BER using the shortest interleaver setting. Table D.6.1 -1 shows the Signal to Noise Ratio (SNR), measured in 300 kHz, for which the modem must achieve a bit error rate of 1e-5 or better.

Table D.6.1 -1: Minimum Performance Specifications for Achieving a Bit Error Rate

of 1e-5 in an AWGN Channel

Modem Data Rate (kbps)	SNR (dB in 300 kHz) for 1e-5 BER
112.5	4
225	7
450	12
600	14
768	17
960	20
1152	22

D.6.2 Doppler Shift Test

The modem shall acquire and maintain synchronisation for at least 5 minutes with test signals with high SNR and Doppler shifts of +800 Hz and –800 Hz.

D.6.3 Doppler Sweep Performance

The AWGN BER test at the highest rate shall be repeated with a test signal having a frequency offset that continuously varies at a rate of 40 Hz/s between the limits of +800 Hz and -800 Hz, such that a plot of frequency offset versus time describes a periodic “triangle” waveform having a period of 160 (3200/40) seconds. During a test of duration 1 hour, the modem shall achieve a BER of 1.0E-5 or less at a high SNR.

INTENTIONALLY BLANK

ANNEX E (OPTIONAL) TECHNICAL SPECIFICATIONS TO ENSURE INTEROPERABILITY OF SERIAL WAVEFORMS FOR 500 KHZ BANDWIDTH LINE OF SIGHT VHF AND UHF RADIO DATA COMMUNICATIONS

E.1. Introduction

E.1.1 Purpose

The purpose of this waveform specification is to define the requirements for a 180 kbps to 1920 kbps QAM UHF modem waveform and provide a detailed description of modem functions to ensure interoperability within complying VHF and UHF radio networks.

E.1.2 Approach and Structure of this Document.

The waveforms described in this document were designed to satisfy NATO requirements. This document specifies waveforms for general applications, including broadcast, Automatic Repeat reQuest (ARQ) and time-division multiple access (TDMA) operation.

E.1.3 Overview

This section presents a modem waveform and coding specification for data rates of 180, 360, 720, 960, 1280, 1600, and 1920 kilobits per second (kbps) operating within channel bandwidths of 500 kHz. This self-identifying waveform family is required for 180 - 1920 kbps operation for systems complying with this document.

A block interleaver shall be used to obtain four interleaving lengths for each data rate, referred to as Very Short, Short, Long, and Very Long in this document. For 180, 360, 720, and 960 kbps the Very Short interleaver is 0.6 ms; all other data rates have a Very Short interleaver of 1.5 ms. For all data rates, the Short, Long, and Very Long interleaver lengths are 12, 96, and 768 milliseconds (ms) respectively. A rate ½, a constraint length 7, convolutional code is used. For data rates of 1280, 1600, and 1920 kbps, the code is punctured to rate $^{5}/_{6}$. For rates of 720 and 960 kbps, the code is punctured to rate ⅔. A full-tail-biting approach, with the block size equal to the interleaver length, is used.

The waveform uses Phase Shift Keying (PSK) and Quadrature Amplitude Modulation (QAM).

The data rate and interleaver settings are explicitly transmitted as a part of the waveform, both as part of the initial preamble and then periodically as both a reinserted preamble and in the periodic known symbol blocks. The receive modem is required to be able to deduce the data rate and interleaver setting either from the preamble or from the reinserted preamble portion of the waveform. A summary of the waveform characteristics is provided in tabular form in Table E.1.3 -1 below.

Table E.1.3 -1: Summary of Waveform Characteristics

Data Rate	Modulation	Block Size		Code Rate	Interleaver Settings (ms)	Reinserted Preamble Size (symbols)
		Data	Training			
180	BPSK	216	23	½	0.6, 12, 96, 768	183
360	QPSK			½		
720	8PSK			⅔		
960	16QAM			⅔		
1280	16QAM	576	23	5/6	1.5, 12, 96, 768	87
1600	32QAM			5/6		
1920	64QAM			5/6		

E.2.Modulation

The symbol rate for all symbols shall be 400000 symbols-per-second, which shall be accurate to a minimum of ±4 symbols-per-second (10 parts per million (ppm)) when the transmit data clock is generated by the modem and not provided by the data terminal equipment (DTE). Phase-shift keying (PSK) and quadrature amplitude modulation (QAM) modulation techniques shall be used. The sub-carrier (or pair of quadrature sub-carriers in the case of QAM) shall be centered at f_c accurate to a minimum of 0.1 ppm. The phase of the Quadrature sub-carrier relative to the In-phase carrier shall be 90 degrees. The correct relationship can be achieved by making the In-phase sub-carrier cos(f_c) and the Quadrature sub-carrier -sin(f_c).

The power spectral density of the modulator output signal should be constrained to be at least 20 decibels (dB) below the signal level measured at f_c, when tested outside of the band from f_c -250 kHz to f_c +250 kHz. The measured ripple shall be no more than ±2 dB in the range from f_c -175 kHz to f_c +175 kHz. The pulse shaping filter which shall be employed is a root-raised cosine filter, specified by the T/2 (800 kHz sampling rate) spaced coefficients of the finite impulse response (FIR) filter:

0.021220659078919377,
-0.018296424333794137,
-0.037513179839879389,
0.065296144145032972,
0.053051647697298421,
-0.170297633845998941,
-0.064237155776998608,
0.621797410509131954,
1.068309886183790969,
0.621797410509131954,
-0.064237155776998608,

-0.170297633845998941,
0.053051647697298421,
0.065296144145032972,
-0.037513179839879389,
-0.018296424333794137,
0.021220659078919377

E.2.1 Known Symbols

For all known symbols, which are defined as those symbols that are known to the receiver prior to reception, the modulation used shall be PSK, with the symbol mapping shown in Table E.2.1 -1 and Figure E.2.1 -1. No scrambling shall be applied to the known symbols.

Table E.2.1 -1: 8PSK Symbol Mapping

Symbol Number	Phase	In-Phase	Quadrature
0	0	1.000000	0.000000
1	$\pi/4$	0.707107	0.707107
2	$\pi/2$	0.000000	1.000000
3	$3\pi/4$	-0.707107	0.707107
4	π	-1.000000	0.000000
5	$5\pi/4$	-0.707107	-0.707107
6	$3\pi/2$	0.0000000	-1.000000
7	$7\pi/4$	0.707107	-0.707107

Note that the complex symbol values = exp[jnπ/4] where n is the symbol number.

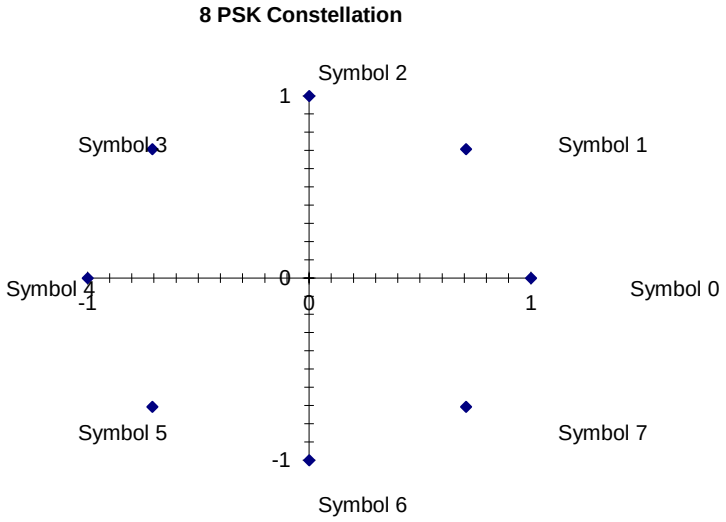


Figure E.2.1 -1: 8PSK Signal Constellation and Symbol Mapping

E.2.2 Data Symbols

For data symbols, the modulation used shall depend upon the data rate. Table E.2.2 -1 specifies the modulation that shall be used with each data rate.

Table E.2.2 -1: Modulation Used to Obtain Each Data Rate

Data Rate (kbps)	Modulation
180	BPSK
360	QPSK
720	8PSK
960	16QAM
1280	16QAM
1600	32QAM
1920	64QAM

All PSK constellations are scrambled to appear, on-air, as an 8PSK constellation. The QAM constellations have been modified from conventional rectangular constellations to improve the peak-to-average ratio.

E.2.2.1 PSK Data Symbols

For the PSK constellations, a distinction is made between the data bits and the symbol number for the purposes of scrambling the QPSK modulation to appear as 8PSK, on-air. Scrambling is applied as a modulo 8 addition of a scrambling sequence to the 8PSK symbol number. Transcoding is an operation which links a symbol to be transmitted to a group of data bits.

E.2.2.1.1 BPSK Symbol Mapping

For the 180 kbps user data rate, transcoding shall be achieved by linking one of the symbols specified in Table E.2.2 -1 to a set of two consecutive data bits (dibit) as shown in Table E.2.2.1.1 -1.

Table E.2.2.1.1 -1: Transcoding for 180 kbps

Bit	Symbol
0	0
1	4

E.2.2.1.2 QPSK Symbol Mapping

For the 360 kbps user data rate, transcoding shall be achieved by linking one of the symbols specified in Table E.2.1 -1 to a set of two consecutive data bits (dibit) as shown in Table E.2.2.1.2 -1. In this table, the leftmost bit of the dibit shall be the older bit; i.e., fetched from the interleaver before the rightmost bit.

Table E.2.2.1.2 -1: Transcoding for 360 kbps

Dibit	Symbol
00	0
01	2
11	4
10	6

E.2.2.1.3 8PSK Symbol Mapping

For the 720 kbps user data rate transcoding shall be achieved by linking one symbol to a set of three consecutive data bits (tribit) as shown in Table E.2.2.1.3 -1. In this table, the leftmost bit of the tribit shall be the oldest bit; i.e., fetched from the interleaver before the other two and the rightmost bit is the most recent bit.

Table E.2.2.1.3 -1: Transcoding for 720 kbps

Tribit	Symbol
000	1
001	0
010	2
011	3
100	6
101	7
110	5
111	4

E.2.2.1.4 QAM Data Symbols

For the QAM constellations, no distinction is made between the number formed directly from the data bits and the symbol number. Each set of 4 bits (16QAM), 5 bits (32QAM) or 6 bits (64QAM) is mapped directly to a QAM symbol. For example, the four bit grouping 0111 would map to symbol 7 in the 16QAM constellation while the 6 bit grouping 100011 would map to symbol 35 in the 64QAM constellation. Again, in each case the leftmost bit shall be the oldest bit, i.e. fetched from the interleaver before the other bits, and the rightmost bit is the most recent bit.

The mapping of bits to symbols for the QAM constellations has been selected to minimize the number of bit errors incurred when errors involve adjacent signaling points in the constellation.

E.2.2.1.5 The 16QAM Constellation

The constellation points which shall be used for 16QAM are shown in Figure E.2.2.1.5 -1 and specified in terms of their In-phase and Quadrature components in Table E.2.2.1.5-1.

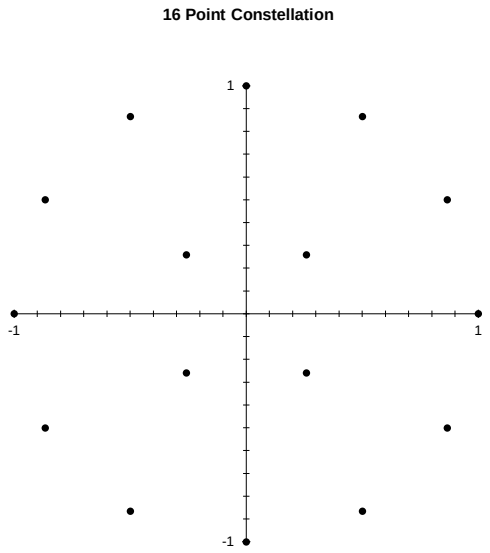


Figure E.2.2.1.5-1: 16QAM Signalling Constellation

Table E.2.2.1.5-1: In-phase and Quadrature Components of Each 16QAM Symbol

Symbol Number	In-Phase	Quadrature
0	0.866025	0.500000
1	0.500000	0.866025
2	1.000000	0.000000
3	0.258819	0.258819
4	-0.500000	0.866025
5	0.000000	1.000000
6	-0.866025	0.500000
7	-0.258819	0.258819
8	0.500000	-0.866025
9	0.000000	-1.000000
10	0.866025	-0.500000
11	0.258819	-0.258819
12	-0.866025	-0.500000
13	-0.500000	-0.866025
14	-1.000000	0.000000
15	-0.258819	-0.258819

E.2.2.1.6 The 32QAM Constellation

The constellation points which shall be used for 32QAM are shown in Figure E.2.2.1.6 -1 and specified in terms of their In-phase and Quadrature components in Table E.2.2.1.6-1.

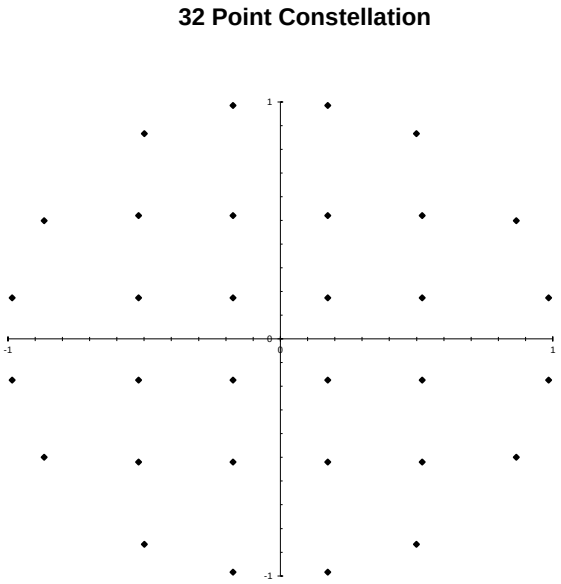


Figure E.2.2.1.6-1: 32QAM Signalling Constellation

Table E.2.2.1.6-1: In-phase and Quadrature Components of Each 32QAM Symbol

Symbol Number	In-Phase	Quadrature	Symbol Number	In-Phase	Quadrature
0	0.866380	0.499386	16	0.866380	-0.499386
1	0.984849	0.173415	17	0.984849	-0.173415
2	0.499386	0.866380	18	0.499386	-0.866380
3	0.173415	0.984849	19	0.173415	-0.984849
4	0.520246	0.520246	20	0.520246	-0.520246
5	0.520246	0.173415	21	0.520246	-0.173415
6	0.173415	0.520246	22	0.173415	-0.520246
7	0.173415	0.173415	23	0.173415	-0.173415
8	-0.866380	0.499386	24	-0.866380	-0.499386
9	-0.984849	0.173415	25	-0.984849	-0.173415
10	-0.499386	0.866380	26	-0.499386	-0.866380
11	-0.173415	0.984849	27	-0.173415	-0.984849
12	-0.520246	0.520246	28	-0.520246	-0.520246
13	-0.520246	0.173415	29	-0.520246	-0.173415

14	-0.173415	0.520246	30	-0.173415	-0.520246
15	-0.173415	0.173415	31	-0.173415	-0.173415

E.2.2.1.7 The 64QAM Constellation

The constellation points which shall be used for the 64QAM modulation are shown in Figure E.2.2.1.7 -1 and specified in terms of their In-phase and Quadrature components in Table E.2.2.1.7 -1.

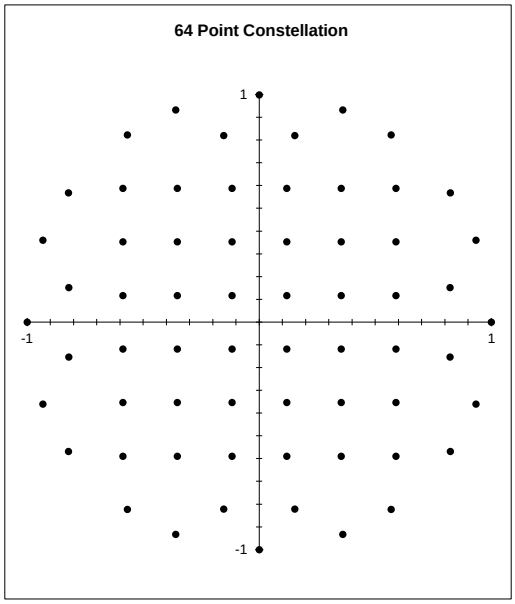


Figure E.2.2.1.7 -1: 64QAM Signalling Constellation

Table E.2.2.1.7 -1: In-phase and Quadrature Components of Each 64QAM Symbol

Symbol Number	In-Phase	Quadrature	Symbol Number	In-Phase	Quadrature
0	1.000000	0.000000	32	0.000000	1.000000
1	0.822878	0.568218	33	-0.822878	0.568218
2	0.821137	0.152996	34	-0.821137	0.152996
3	0.932897	0.360142	35	-0.932897	0.360142
4	0.000000	-1.000000	36	-1.000000	0.000000
5	0.822878	-0.568218	37	-0.822878	-0.568218
6	0.821137	-0.152996	38	-0.821137	-0.152996
7	0.932897	-0.360142	39	-0.932897	-0.360142
8	0.568218	0.822878	40	-0.568218	0.822878
9	0.588429	0.588429	41	-0.588429	0.588429
10	0.588429	0.117686	42	-0.588429	0.117686
11	0.588429	0.353057	43	-0.588429	0.353057
12	0.568218	-0.822878	44	-0.568218	-0.822878
13	0.588429	-0.588429	45	-0.588429	-0.588429
14	0.588429	-0.117686	46	-0.588429	-0.117686
15	0.588429	-0.353057	47	-0.588429	-0.353057
16	0.152996	0.821137	48	-0.152996	0.821137
17	0.117686	0.588429	49	-0.117686	0.588429
18	0.117686	0.117686	50	-0.117686	0.117686
19	0.117686	0.353057	51	-0.117686	0.353057
20	0.152996	-0.821137	52	-0.152996	-0.821137
21	0.117686	-0.588429	53	-0.117686	-0.588429
22	0.117686	-0.117686	54	-0.117686	-0.117686
23	0.117686	-0.353057	55	-0.117686	-0.353057
24	0.360142	0.932897	56	-0.360142	0.932897
25	0.353057	0.588429	57	-0.353057	0.588429
26	0.353057	0.117686	58	-0.353057	0.117686
27	0.353057	0.353057	59	-0.353057	0.353057
28	0.360142	-0.932897	60	-0.360142	-0.932897
29	0.353057	-0.588429	61	-0.353057	-0.588429
30	0.353057	-0.117686	62	-0.353057	-0.117686
31	0.353057	-0.353057	63	-0.353057	-0.353057

E.2.3 Data Scrambling

Data symbols for the 8PSK symbol constellation (180 kbps, 360 kbps, and 720 kbps) shall be scrambled by modulo 8 addition with a scrambling sequence. The data symbols for the 16QAM, 32QAM and 64QAM constellations (960 kbps, 1280 kbps, 1600 kbps, and 1920 kbps) shall be scrambled by using an exclusive or (XOR)

operation. Sequentially, the data bits forming each symbol (4 for 16QAM, 5 for 32QAM, and 6 for 64QAM) shall be XOR'd with an equal number of bits from the scrambling sequence. The scrambling sequence generator polynomial shall be $x^9 + x^4 + 1$ and the generator shall be initialized to 1 at the start of each data frame. A block diagram of the scrambling sequence generator is shown in Figure E.2.3 -1.

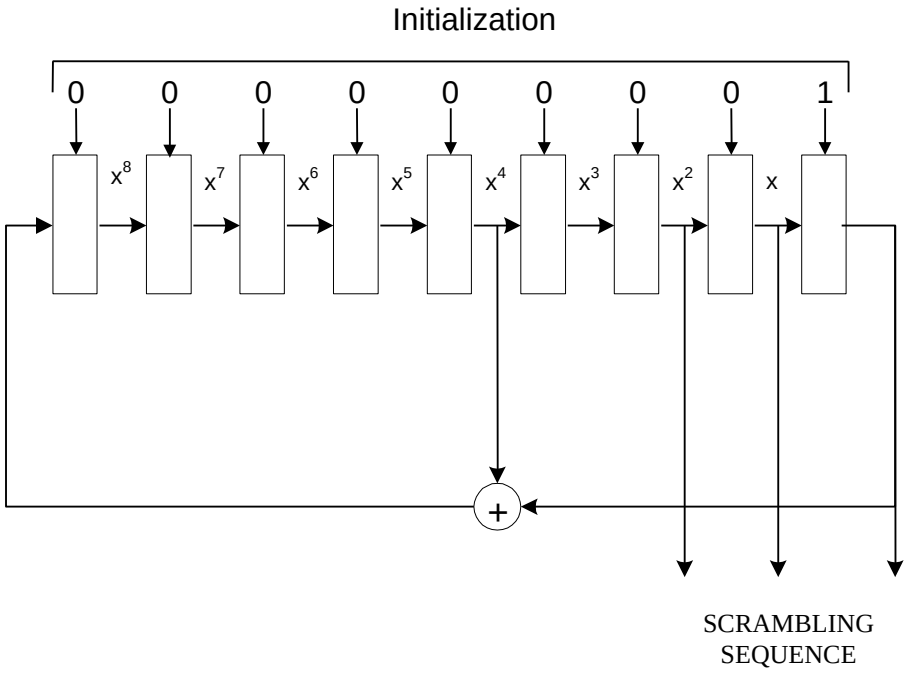


Figure E.2.3 -1: Scrambling Sequence Generator Illustrating Scrambling Generator for 8PSK Symbols

For 8PSK symbols (180, 360, and 720 kbps), the scrambling shall be carried out taking the modulo 8 sum of the numerical value of the binary triplet consisting of the last (rightmost) three bits in the shift register, and the symbol number (transcoded value). For example, if the last three bits in the scrambling sequence shift register were 010 which has a numerical value equal 2, and the symbol number before scrambling was 6, symbol 0 would be transmitted since: (6+2) Modulo 8 = 0. For 16QAM symbols, scrambling shall be carried out by XORing the 4 bit number consisting of the last (rightmost) four bits in the shift register with the symbol number. For example, if the last 4 bits in the scrambling sequence shift register were 0101 and the 16QAM symbol number before scrambling was 3 (i.e. 0011), symbol 6 (0110) would be transmitted. For 32QAM symbols, scrambling shall be carried out by XORing the 5 bit number formed by the last (rightmost) five bits in the shift register with the symbol number. For 64QAM symbols, scrambling shall be carried out by XORing the 6 bit number formed by the last (rightmost) six bits in the shift register with the symbol number.

After each data symbol is scrambled, the generator shall be iterated (shifted) the required number of times to produce all new bits for use in scrambling the next symbol (i.e., 3 iterations for 8PSK, 4 iterations for 16QAM, 5 iterations for 32QAM and 6 iterations for 64QAM). Since the generator is iterated after the bits are used, the first data symbol of every data frame shall, therefore, be scrambled by the appropriate number of bits from the initialization value of 00000001.

The length of the scrambling sequence is 511 bits. For a 576 symbol data block with 6 bits per symbol, this means that the scrambling sequence will be repeated just slightly less than 7 times.

E.3. Frame structure

The frame structure that shall be used for all data rates is described in the sections below.

E.3.1 Synchronization and Reinserted Preambles

The synchronization preamble is used for rapid initial synchronization. The reinserted preamble is used to facilitate acquisition of an ongoing transmission (acquisition on data).

E.3.2 Frank-Heimiller and Barker Sequences

All data rate frames shall use combinations of Frank-Heimiller (length 16) and Barker (length 13) sequences to form the preamble portions of the transmission. The Barker sequence shall be used to encode the data rate and interleaver properties of the transmission and to mark interleaver boundary points.

The Frank-Heimiller sequence used for this waveform is expressed as a series of 8PSK symbols as follows:

0, 0, 0, 0, 0, 2, 4, 6, 0, 4, 0, 4, 0, 6, 4, 2

Depending on the particular data rate, each preamble shall use a cyclic repetition of a portion of this sequence.

The 13 symbol Barker sequence shall be used to encode one or two bits of data for each sequence of 13 8PSK symbols. The base Barker sequence, expressed as a series of 8PSK symbols is shown below. When the Barker sequence is phase modulated all 13 symbols are phase shifted by the same degree. Both BPSK and QPSK modulations are employed to encode data on Barker sequences within this specification.

0, 4, 0, 4, 0, 0, 4, 4, 0, 0, 0, 0, 0

The sections below detail the 8PSK symbols that comprise each Barker sequence as required for particular encoded data value.

E.3.3 180, 360, 720, and 960 kbps Data Rates

The initial 600 symbol preamble shall be followed by 160 frames of alternating data and known symbols. Each data frame shall consist of a data block followed by a mini-probe consisting of 23 symbols of known data. The data block shall consist of 216 data symbols.

After 100 data frames, the reinserted preamble shall be transmitted, with a length of 123 symbols. This is done to facilitate late acquisition, Doppler shift removal, and sync adjustment.

It should be noted that the last data block in of the 100 data frames is not followed by a 23 symbol mini-probe segment. Instead, the known data portion of the 123 symbol preamble immediately follows this final data block.

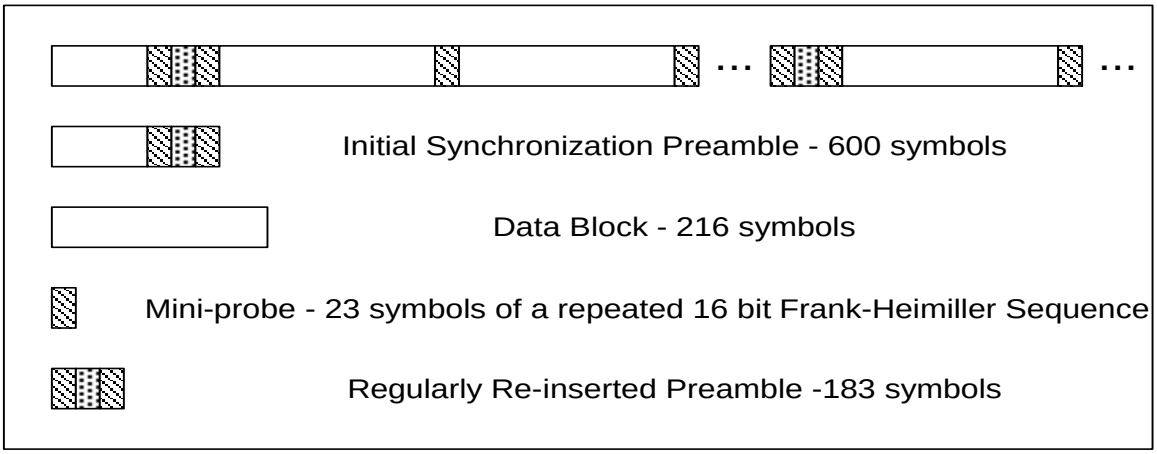


Figure E.3.3 -1: Frame Structure for Data Rates of 180, 360, 720 and 960 kbps

E.3.3.1 Synchronization Preamble

The mandatory portion of the preamble, which shall always be transmitted, and shall always appear at the beginning of an interleaver boundary, shall consist of 600 symbols. The first 417 symbols are intended exclusively for synchronisation and Doppler offset removal purposes while the final 183 symbols, which are common with the reinserted preamble, also carry information regarding the data rate and interleaver settings. Expressed as a sequence of symbols, using the symbol numbers given in Table E.2.1 -1 the second section of the synchronisation preamble shall be as follows:

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

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

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

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

The remaining 183 symbols of the synchronization preamble shall be the same as the reinserted preamble, as defined in the section below.

E.3.3.2 Reinserted Preamble

The reinserted preamble shall be defined for the 180, 360, 720, and 960 kbps data rates using a series of Frank-Heimiller and Barker sequences. The interleaver length and data rate shall be encoded in the preamble using the Barker sequences, as shown in Table E.3.3.2 -1.

Table E.3.3.2 -1: Reinserted Preamble: 180, 360, 720, and 960 kbps

Number of Symbols	Sequence	Data
30	Frank	0, 0, 0, 0, 0, 2, 4, 6, 0, 4, 0, 4, 0, 6, 4, 2, 0, 0, 0, 0, 0, 2, 4, 6, 0, 4, 0, 4, 0, 6
13	Barker	180 or 720 kbps: 0, 4, 0, 4, 0, 0, 4, 4, 0, 0, 0, 0, 0 360 or 960 kbps: 4, 0, 4, 0, 4, 4, 0, 0, 4, 4, 4, 4, 4
13	Barker	180 or 360 kbps: 0, 4, 0, 4, 0, 0, 4, 4, 0, 0, 0, 0, 0 720 or 960 kbps: 4, 0, 4, 0, 4, 4, 0, 0, 4, 4, 4, 4, 4
29	Frank	0, 0, 0, 0, 0, 2, 4, 6, 0, 4, 0, 4, 0, 6, 4, 2, 0, 0, 0, 0, 0, 2, 4, 6, 0, 4, 0, 4, 0
13	Barker	Interleaver boundary: 4, 0, 4, 0, 4, 4, 0, 0, 4, 4, 4, 4, 4 Not interleaver boundary: 0, 4, 0, 4, 0, 0, 4, 4, 0, 0, 0, 0, 0
29	Frank	0, 0, 0, 0, 0, 2, 4, 6, 0, 4, 0, 4, 0, 6, 4, 2, 0, 0, 0, 0, 0, 2, 4, 6, 0, 4, 0, 4, 0
13	Barker	Very Short or Long interleaver: 0, 4, 0, 4, 0, 0, 4, 4, 0, 0, 0, 0, 0

		Short or Very Long interleaver: 4, 0, 4, 0, 4, 4, 0, 0, 4, 4, 4, 4, 4
13	Barker	Very Short or Short interleaver: 0, 4, 0, 4, 0, 0, 4, 4, 0, 0, 0, 0, 0 Long or Very Long interleaver: 4, 0, 4, 0, 4, 4, 0, 0, 4, 4, 4, 4, 4
30	Frank	0, 0, 0, 0, 0, 2, 4, 6, 0, 4, 0, 4, 0, 6, 4, 2, 0, 0, 0, 0, 0, 2, 4, 6, 0, 4, 0, 4, 0, 6

There shall be a total of 183 symbols formed using these sequences.

E.3.4 1280, 1600, and 1920 kbps Data Rates

The initial 600 symbol preamble shall be followed by 64 frames of alternating data and known symbols. Each data frame shall consist of a data block followed by a mini-probe consisting of 23 symbols of known data. The data block shall consist of 576 data symbols.

After 64 data frames, the reinserted preamble shall be transmitted, with a length of 87 symbols. This is done to facilitate late acquisition, Doppler shift removal, and sync adjustment.

It should be noted that the last data block in of the 64 data frames is not followed by a 23 symbol mini-probe segment. Instead, the known data portion of the 87 symbol preamble immediately follows this final data block.

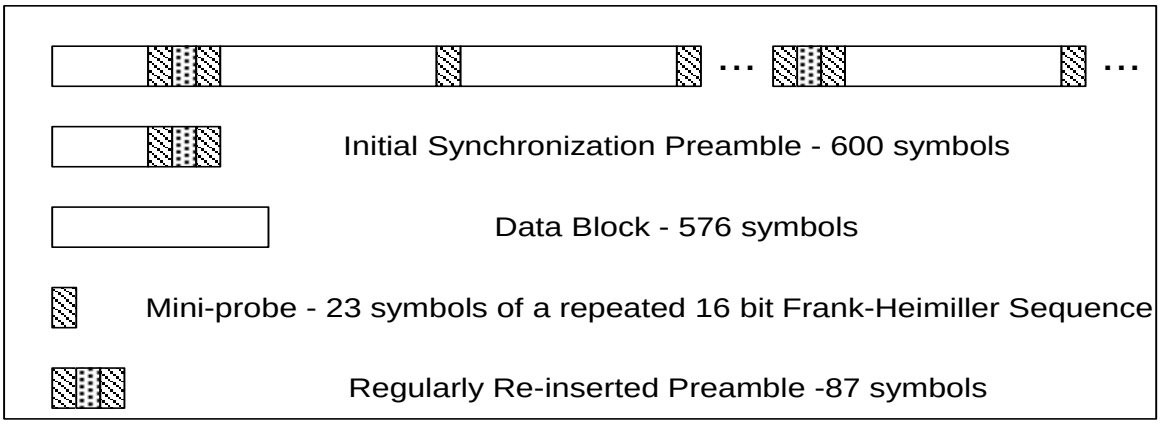


Figure E.3.4 -1: Frame Structure for Data Rates of 1280, 1600, and 1920 kbps

E.3.4.1 Synchronization Preamble

The mandatory portion of the preamble, which shall always be transmitted, and shall always appear at the beginning of an interleaver boundary, shall consist of 600 symbols. The first 513 symbols are intended exclusively for synchronisation and Doppler offset removal purposes while the final 87 symbols, which are common with the reinserted preamble, also carry information regarding the data rate and interleaver settings. Expressed as a sequence of symbols, using the symbol numbers given in Table E.2.1 -1 the second section of the synchronisation preamble shall be as follows:

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

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

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

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

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

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

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

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

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

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

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

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

4,0,4,0,0,0,0,0,0,4,4,4,4,4,4,4,0,4,0,4,0,4,0,0,4,4,0,0,4,4,4,0,

4,4,4,0,4,0,0,4,0,4,4,0,0,0,4,4,0,4,4,4,4,0,4,4,0,4,0,4,4,0,4,4,

0,0,4,0,0,4,0,0,0,4,4,4,0,0,0,0,4,0,4,4,4,4,4,0,0,4,0,4,0,4,4,4,

0,0,4,4,0,4,0,0,0,4,0,0,4,4,4,4,0,0,0,4,0,4,0,0,0,0,4,4,0,0,0,0,

0

It should be noted that the last 417 symbols of this pseudo random sequence are common with the 417 symbols of the pseudo random sequence employed for data rates less than 1280 kbps. The remaining symbols of the synchronization preamble shall be the same as the reinserted preamble, as defined in the section below.

E.3.4.2 Reinserted Preamble

The reinserted preamble shall be defined for the 1280, 1600, and 1920 kbps data rates using a series of Frank-Heimiller and Barker sequences. The interleaver length and data rate shall be encoded in the preamble using the Barker sequences, as shown in Table E.3.4.2 -1.

Table E.3.4.2 -1: Reinserted Preamble: 1280, 1600, 1920 kbps

Number of Symbols	Sequence	Data
24	Frank	0, 0, 0, 0, 0, 2, 4, 6, 0, 4, 0, 4, 0, 6, 4, 2, 0, 0, 0, 0, 0, 2, 4, 6
13	Barker	1280 kbps: 0, 4, 0, 4, 0, 0, 4, 4, 0, 0, 0, 0, 0 1600 kbps: 2, 6, 2, 6, 2, 2, 6, 6, 2, 2, 2, 2, 2 1920 kbps: 6, 2, 6, 2, 6, 6, 2, 2, 6, 6, 6, 6, 6
13	Barker	Interleaver boundary: 4, 0, 4, 0, 4, 4, 0, 0, 4, 4, 4, 4, 4 Not interleaver boundary: 0, 4, 0, 4, 0, 0, 4, 4, 0, 0, 0, 0, 0
13	Barker	Very Short Interleaver: 0, 4, 0, 4, 0, 0, 4, 4, 0, 0, 0, 0, 0 Short Interleaver: 2, 6, 2, 6, 2, 2, 6, 6, 2, 2, 2, 2, 2 Long Interleaver: 6, 2, 6, 2, 6, 6, 2, 2, 6, 6, 6, 6, 6 Very Long Interleaver: 4, 0, 4, 0, 4, 4, 0, 0, 4, 4, 4, 4, 4
24	Frank	0, 0, 0, 0, 0, 2, 4, 6, 0, 4, 0, 4, 0, 6, 4, 2, 0, 0, 0, 0, 0, 2, 4, 6

There is a total of 87 symbols formed using these sequences.

E.3.5 Mini-Probes

For all data rates, mini-probes 23 symbols in length shall be inserted following every data block, with the exception of the last data block prior to the reinserted preamble. Using the 8PSK symbol mapping, each mini-probe shall be based on a 16 bit Frank-Heimiller sequence. The sequence that shall be used, specified in terms of the 8PSK symbol numbers, is given by:

0, 0, 0, 0, 0, 2, 4, 6, 0, 4, 0, 4, 0, 6, 4, 2, 0, 0, 0, 0, 0, 2, 4

E.4.Coding and interleaving

The interleaver used shall be a block interleaver. Each block of input data shall also be encoded using a block encoding technique with a code block size equal to the size of the block interleaver. Thus, the input data bits will be sent as successive blocks of bits that span the duration of the interleaver length selected.

Table E.4-1 shows the interleaver length in milliseconds for each combination of data rate and interleaver type (very short, short, long, and very long).

Table E.4-1: Interleaver Length in Milliseconds as a Function of Data Rate and Interleaver Type

Data Rate (kbps)	Interleaver Length			
	Very Short	Short	Long	Very Long
	Input Data Block Size (ms)			
180	0.6	12	96	768
360	0.6	12	96	768
720	0.6	12	96	768
960	0.6	12	96	768
1280	1.5	12	96	768
1600	1.5	12	96	768
1920	1.5	12	96	768

Table E.4-2 shows the number of input data bits per block as function of both data rate and interleaver length. Note that an “input data block” should not be confused with the 216 or 576 symbol data block that is part of a data frame in the waveform format. The bits from an input data block, when expanded by the application of the forward error correction coding, are sufficient to fill the specified interleaver.

Table E.4-2: Input Data Block Size in Bits as a Function of Data Rate and Interleaver Length

Data Rate (kbps)	Interleaver Length			
	Very Short	Short	Long	Very Long
	Input Data Block Size (bits)			
180	108	2160	17280	138240
360	216	432	3456	276480
720	432	864	69120	552960
960	576	1152	92160	727280
1280	1920	15360	122880	983040
1600	2400	19200	153600	1228800
1920	2880	23040	184320	1474560

E.4.1 Block Boundary Alignment

Each code block shall be interleaved within a single block interleaver of the same size. The boundaries of these blocks shall be aligned such that the beginning of the first data frame following each reinserted preamble shall coincide with an interleaver boundary. Thus for an interleaver length of 8 frames, the first eight data frames following a reinserted preamble will contain all of the encoded bits for a single input data block. The first data symbol from the first data frame in each interleaver set shall have as its most significant bit (MSB) the first bit fetched from the interleaver.

E.4.2 Block Encoding

The full-tail-biting and puncturing techniques shall be used with a rate 1/2 convolutional code to produce a block code that is the same length as the interleaver. For those data rates where the code is punctured to rate 5/6 or rate 2/3, the punctured block shall still fit exactly within the interleaver.

E.4.3 Rate ½ Convolutional Code

A constraint length 7, rate 1/2 convolutional code shall be used prior to puncturing. Figure E.4.3 -1 is a pictorial representation of the encoder. The two generator polynomials used shall be:

$T_1 = x^6 + x^4 + x^3 + x + 1$
$T_2 = x^6 + x^5 + x^4 + x^3 + 1$

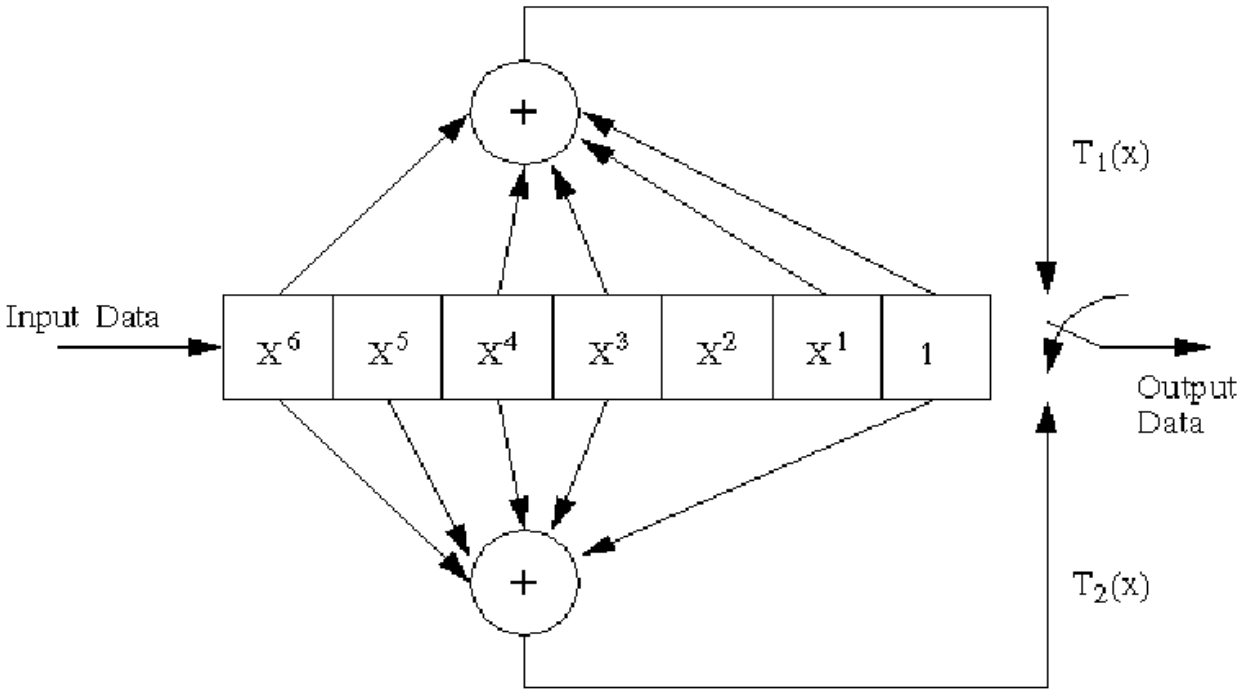


Figure E.4.3 -1: Constraint Length 7, Rate ½ Convolutional Encoder

The two summing nodes in the figure represent modulo 2 addition. For each bit input to the encoder, two bits are taken from the encoder, with the upper output bit, $T_1(x)$, taken first.

E.4.4 Full-Tail-Biting Encoding

To begin encoding each block of input data, the encoder shall be preloaded by shifting in the first eight input data bits without taking any output bits. These eight input bits shall be temporarily saved so that they can be used to "flush" the encoder. The first two coded output bits shall be taken after the ninth bit has been shifted in, and shall be defined to be the first two bits of the resulting block code. After the last input data bit has been encoded, the first eight "saved" data bits shall be encoded. Note that the encoder shift register should not be changed before encoding these saved bits; i.e., it should be filled with the last nine input data bits. The eight "saved" data bits are encoded by shifting them into the encoder one at a time, beginning with the earliest of the eight. The encoding thus continues by taking the two resulting coded output bits as each of the saved eight bits is shifted in. These encoded bits shall be the final bits of the resulting (unpunctured) block code. Prior to puncturing, the resulting block code

will have exactly twice as many bits as the input information bits. Puncturing of the rate ½ code to the required rate $^5/_6$ shall be done prior to sending bits to the interleaver.

E.4.5 Puncturing to Rate $^5/_6$

In order to obtain a rate $^5/_6$ code from the rate ½ code used, the output of the encoder must be punctured by not transmitting 4 bits out of every 10. Puncturing shall be performed by using a puncturing mask of 1 1 1 0 0 1 1 0 0 1, applied to the bits output from the encoder. In this notation a 1 indicates that the bit is retained and a 0 indicates that the bit is not transmitted. For an encoder generated sequence of:

T1(k), T2(k), T1(k+1), T2(k+1), T1(k+2), T2(k+2) . . .

the transmitted sequence shall be:

T1(k), T2(k), T1(k+1), T2(k+2) . . .

Defining T1(0), T2(0) to be the first two bits of the block code generated as defined in Paragraph E.4.2 , then the value of k in the above sequences shall be an integral multiple of 5. The block code shall be punctured in this manner before being input to the interleaver.

E.4.6 Puncturing to Rate ⅔

In order to obtain a rate ⅔ code from the rate ½ code used, the output of the encoder must be punctured by not transmitting 1 bit out of every 4. Puncturing shall be performed by using a puncturing mask of 1 1 1 0, applied to the bits output from the encoder. In this notation a 1 indicates that the bit is retained and a 0 indicates that the bit is not transmitted. For an encoder generated sequence of:

T1(k), T2(k), T1(k+1), T2(k+1), T1(k+2), . . .

the transmitted sequence shall be:

T1(k), T2(k), T1(k+1), T1(k+2), . . .

Defining T1(0), T2(0) to be the first two bits of the block code generated as defined in Paragraph E.4.2 , then the value of k in the above sequences shall be an integral multiple of 2. The block code shall be punctured in this manner before being input to the interleaver.

E.4.7 Block Interleaver Structure

The block interleaver used is designed to separate neighbouring bits in the punctured block code as far as possible over the span of the interleaver with the largest separations resulting for the bits that were originally closest to each other. Because of

the 28 different combinations of data rates and interleaver lengths, a flexible interleaver structure is needed.

E.4.7.1 Interleaver Size in Bits

The interleaver shall consist of a single dimension array, numbered from 0 to its size in bits –1. The array size shall depend on both the data rate and interleaver length selected as shown in Table E.4.7.1 -1.

Table E.4.7.1 -1: Interleaver Size in Bits as a Function of Data Rate and Interleaver Length

Data Rate (kbps)	Interleaver Length			
	Very Short	Short	Long	Very Long
	Interleaver Size in Bits			
180	216	4320	34560	276480
360	432	8640	69120	552960
720	648	12960	103680	829440
960	864	17280	138240	1105920
1280	2304	18432	147456	1179648
1600	2880	23040	184320	1474560
1920	3456	27648	221184	1769472

E.4.7.2 Interleaver Load

The punctured block code bits shall be loaded into the interleaver array beginning with location 0. The location for loading each successive bit shall be obtained from the previous location by incrementing by the “Interleaver Increment Value” specified in Table E.4.7.2 -1, modulo the “Interleaver Size in Bits.”

Defining the first punctured block code bit to be B(0), then the load location for B(n) is given by:

Load Location = (n * Interleaver Increment Value) Modulo (Interleaver Size in Bits)

Thus for 180 kbps, with Very Short interleaver (216 bit size with an increment of 41), the first 8 interleaver load locations are: 0, 41, 82, 123, 164, 205, 30, and 71.

Table E.4.7.2 -1: Interleaver Increment Value as a Function of Data Rate and Interleaver Length

Data Rate (kbps)	Interleaver Length			
	Very Short	Short	Long	Very Long
	Interleaver Increment Value			
180	41	1679	5567	49771
360	65	2029	10459	95803
720	97	3049	16333	155029
960	161	4081	20021	161261
1280	337	5321	22493	159613
1600	451	6631	27191	198031
1920	505	7993	67789	237637

E.4.7.3 Interleaver Fetch

The fetching sequence for all data rates and interleaver lengths shall start with location 0 of the interleaver array and increment the fetch location by 1. This is a simple linear fetch from beginning to end of the interleaver array.

E.5.Operational features and message protocols

E.5.1 User Interfaces

E.5.1.1 Synchronous Data Communications Equipment (DCE) Interface

The modem shall implement a synchronous DCE interface that is compatible with EIA-232, EIA-422 or Mil-Std-188-114A.

E.5.1.2 Conventional Asynchronous Interface (Optional)

The modem may be capable of interfacing with an asynchronous DTE; this is not required behaviour. Conventional RS-232 asynchronous serial interfaces will not support all data rates defined by this standard.

E.5.1.3 Ethernet Interface (Optional)

The modem shall provide an Ethernet interface. The bytes shall be aligned with the Input Data Block boundary following an initial or reinserted preamble.

E.5.2 Onset of Transmission

The modem shall begin a transmission no later than 25 ms after it has received an entire input data block (enough bits to fill a coded and interleaved block), or upon receipt of the last input data bit, whichever occurs first. The latter would only occur when the message is shorter than one interleaver block. A transmission shall be defined as beginning with the keying of the radio, followed by the output of the preamble waveform after the configured pre-key delay, if any.

E.5.3 End of Message

The use of an end-of-message (EOM) in the transmit waveform shall be a configurable option. When the use of an EOM has been selected, a 32-bit EOM pattern shall be appended after the last input data bit of the message. The EOM, expressed in hexadecimal notation is 4B65A5B2, where the left most bit is sent first. If the last bit of the EOM does not fill out an input data block, the remaining bits in the input data block shall be set to zero before encoding and interleaving the block.

If the use of an EOM has been inhibited and the last input data bit does not fill out an input data block, the remaining bits in the input data block shall be set to zero before encoding and interleaving the block. It is anticipated that the use of an EOM will only be inhibited when an ARQ data protocol uses ARQ blocks which completely fill (or nearly so) the selected input data block size (interleaver block). Without this feature, the use of an EOM would require the transmission of an additional interleaver block under these circumstances.

E.5.4 Termination of a Transmission

Upon receipt of a radio silence (or equivalent) command, the modem shall immediately un-key the radio and terminate the transmit waveform.

In normal operation, the modem shall terminate a transmission only after the transmission of the final data frame, including a mini-probe, associated with the final interleaver block. Note that a data frame consists of a 216 or 576 symbol data block followed by a mini-probe. The entire final mini-probe shall be transmitted before the modem unkeys.

E.5.5 Termination of Receive Data Processing

There are a number of events which shall cause the modem to cease processing the received signal to recover data, and return to the acquisition mode. These are necessary because a modem is not able to acquire a new transmission while it is attempting to demodulate and decode data.

E.5.5.1 Detection of EOM

The modem shall always scan all of the decoded bits for the 32-bit EOM pattern defined in Paragraph E.5.3 . Upon detection of the EOM the modem shall return to the acquisition mode. The modem shall continue to deliver decoded bits to the user (DTE) until the final bit immediately preceding the EOM has been delivered.

E.5.5.2 Command to Return to Acquisition

Upon receipt of a command to terminate reception, the modem shall return to the acquisition mode and terminate the delivery of decoded bits to the user (DTE).

E.5.5.3 Receipt of a Specified Number of Data Blocks

The maximum message duration measured in number of Input Data Blocks (interleaver blocks) shall be a configurable parameter. Setting this parameter to zero shall specify that an unlimited number may be received. Once the modem has decoded and delivered to the user (DTE), the number of bits corresponding to the configured maximum message duration, the modem shall return to the acquisition mode and terminate the delivery of decoded bits to the user (DTE). Note that for a given interleaver length, this parameter also specifies the maximum message duration in time, independent of the bit rate. Note that this parameter is the maximum duration and that the transmit end always has the option of using an EOM for shorter transmissions.

Operation with a specified number of input data blocks may be used by an ARQ or TDMA protocol where the size of the packet is fixed, or occasionally changed to accommodate changing propagation conditions. In this case we anticipate that this parameter (maximum message duration) will be sent to the receiving end of the link as part of the ARQ protocol or specified in the TDMA protocols. It would then have to be sent to the receiving modem through the remote control interface or be preconfigured by an operator since it is not embedded in the waveform itself as the data rate and interleaver length parameters are.

E.5.5.4 Initiation of a Transmission

If, If, and only if, the Modem is configured to operate in half-duplex mode with transmit override, the initiation of a transmission by the user (DTE) shall cause the modem to terminate the receive processing and the delivery of decoded bits to the user (DTE).

E.5.6 TDMA Compatibility

E.5.6.1 Jitter

The total uncertainty in the time between the receipt of Request to Send (RTS) and the onset of the transmission (defined above) shall be less than 20 ms.

E.5.6.2 Minimum Time Between Back-to-Back Receptions

The modem shall be capable of receiving a subsequent transmission within 10 ms of the end of an initial transmission if the modem is configured to ignore the autobaud information in the waveform and the same data rate and interleaver settings are used for the initial and subsequent transmissions. If the modem is configured to accept autobaud data rate and interleaver settings, the minimum time between back to back receptions shall be 1.2 * the interleaver length.

E.5.6.3 Associated Communications (Radio) Equipment

The radio must provide a passband of at least 360 kHz with a ripple of not more than +/- 2 dB across the passband. The radio must support linear modulations and must support a signal to noise and distortion (SINAD) of not less than 30 dB for the 64 QAM modulation defined herein.

E.6.Performance Requirements

The minimum performance requirements for the high data rate mode are specified in this section. These requirements are not exhaustive, but are intended to ensure that equipment contains a high-quality implementation of the standard.

E.6.1 Bit Error Rate (BER) Performance

BER performance shall be measured with the channel simulator programmed to simulate an additive white Gaussian noise (AWGN) channel. The AWGN channel shall consist of a single, non-fading path. Each condition shall be measured for at least 15 minutes for 1.0E-5 BER using the shortest interleaver setting. Table E.6.1 -1 shows the Signal to Noise Ratio (SNR), measured in 500 kHz, for which the modem must achieve a bit error rate of 1e-5 or better.

Table E.6.1 -1: Minimum Performance Specifications for Achieving a Bit Error Rate of 1e-5 in an AWGN Channel

Modem Data Rate (kbps)	SNR (dB in 500 kHz) for 1e-5 BER
180	4
360	7
720	12
960	14
1280	17
1600	20
1920	22

E.6.2 Doppler Shift Test

The modem shall acquire and maintain synchronisation for at least 5 minutes with test signals with high SNR and Doppler shifts of +800 Hz and –800 Hz.

E.6.3 Doppler Sweep Performance

The AWGN BER test at the highest rate shall be repeated with a test signal having a frequency offset that continuously varies at a rate of 40 Hz/s between the limits of +800 Hz and -800 Hz, such that a plot of frequency offset versus time describes a periodic “triangle” waveform having a period of 160 (3200/40) seconds. During a test of duration 1 hour, the modem shall achieve a BER of 1.0E-5 or less at a high SNR.

AComP-4691(A)(1)