P_MUL - A PROTOCOL FOR RELIABLE MULTICAST IN BANDWIDTH CONSTRAINED AND DELAYED ACKNOWLEDGEMENT (EMCON) ENVIRONMENTS

ACP 142(A)

COMBINED COMMUNICATIONS-ELECTRONICS BOARD (CCEB)

OCTOBER 2008

FOREWORD

1. The Combined Communications-Electronics Board (CCEB) is comprised of the five member nations, Australia, Canada, New Zealand, United Kingdom and United States and is the Sponsoring Authority for all Allied Communications Publications (ACPs). ACPs are raised and issued under common agreement between the member nations.

2. ACP 142(A), P_MUL - A PROTOCOL FOR RELIABLE MULTICAST IN BANDWIDTH CONSTRAINED AND DELAYED ACKNOWLEDGEMENT (EMCON) ENVIRONMENTS, is an UNCLASSIFIED CCEB publication.

3. This publication contains Allied military information for official purposes only.

4. It is permitted to copy or make extracts from this publication.

5. This ACP is to be maintained and amended in accordance with the provisions of the current version of ACP 198.

THE COMBINED COMMUNICATION-ELECTRONICS BOARD LETTER OF PROMULGATION FOR ACP 142(A)

1. The purpose of this Combined Communication Electronics Board (CCEB) Letter of Promulgation is to implement ACP 142(A) within the Armed Forces of the CCEB Nations. ACP 142(A), P_MUL - A PROTOCOL FOR RELIABLE MULTICAST MESSAGING IN BANDWIDTH CONSTRAINED AND DELAYED ACKNOWLEDGEMENT (EMCON) ENVIRONMENTS, is an UNCLASSIFIED publication developed for Allied use and, under the direction of the CCEB Principals. It is promulgated for guidance, information, and use by the Armed Forces and other users of military communications facilities.

2. ACP 142(A) is effective on receipt for CCEB Nations and when directed by the NATO Military Committee (NAMILCOM) for NATO nations and Strategic Commands.

EFFECTIVE STATUS

Publication	Effective for	Date	Authority
ACP 142(A)	CCEB	On Receipt	LOP

3. All proposed amendments to the publication are to be forwarded to the national coordinating authorities of the CCEB or NAMILCOM.

For the CCEB Principals

Paul Foster

P. FOSTER
Major, CF
CCEB Permanent Secretary

RECORD OF MESSAGE CORRECTIONS

Identification of Change or Correction and Date, time group		Date Entered	By Whom Entered (Signature; rank, grade or rate; name of command)
Change	Correction		

TABLE OF CONTENTS

TABLE OF FIGURES

CHAPTER 1

INTRODUCTION

BACKGROUND

101. Version 1 of this Allied Communication Publication (ACP) was developed and coordinated by the Tactical Communication Task Force (TCTF) of the Allied Information Services (AIS) International Subject Matter Experts (ISME) working group under the auspices of the Combined Communications Electronics Board (CCEB). The initial mandate of this Task Force was to facilitate the interoperability of allied multicast information transfers in bandwidth-constrained and delayed-acknowledgement environments.

102. ACP 142(A) was developed as a joint effort between the NATO Core Enterprise Services Working Group (CESWG) under the Information Systems Sub-Committee with the governance of the NATO C3 Board and, the Messaging Task Force (MTF) under the auspices of the CCEB.

103. The function of ACP 142(A), is to provide a protocol definition for reliable multicast information transfer in bandwidth-constrained and delayed-acknowledgement environments to support efficient allied information transfer. This document also provides examples illustrating various functional aspects of the protocol as well as some guidance on its use and implementation.

104. The protocol described in this document, P_MUL, is the result of a development effort under the multi-national Communications System Networks Interoperability (CSNI) project and enhancements to this protocol under various Combined Warrior Interoperability Demonstration (CWID) experiments. Every effort has been made to make this protocol simple; transparent to the application it supports, as easy as possible to implement and platform-independent.

EVOLUTION

105. Version 1 of this ACP provides the basic specification of the P_MUL protocol. The current specification of this protocol provides a new allied capability, namely reliable multicast information transfer in delayed-acknowledgement environments, in addition to providing a mechanism for more efficient use of bandwidth.

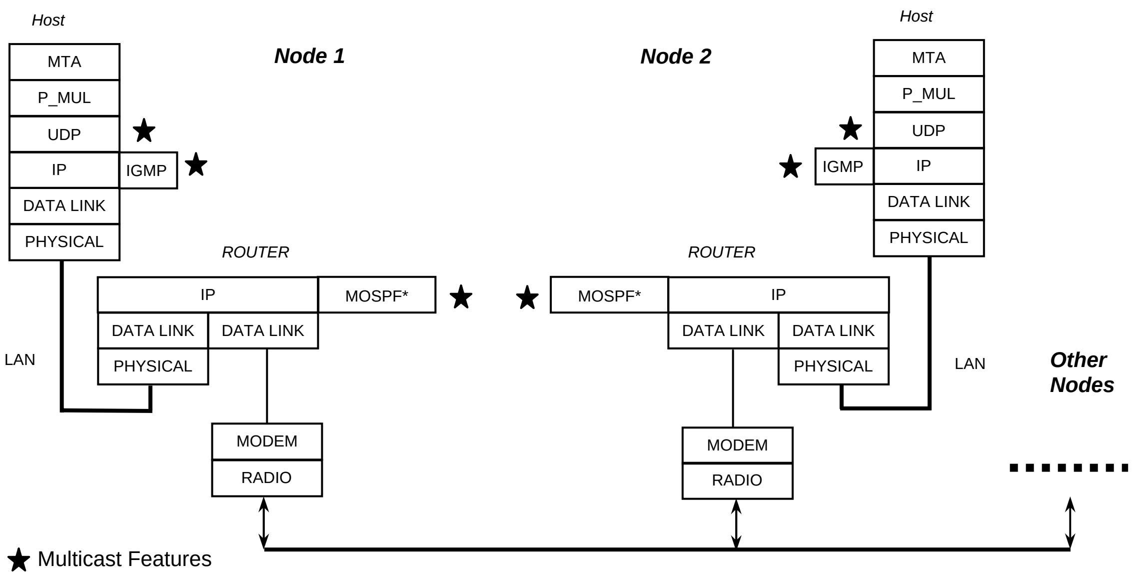
106. ACP 142(A) includes significant improvements to this protocol such as congestion control, priority based pre-emption, adaptability to changes in communications conditions, mechanism to avoid acknowledgement implosion, enabling the use of Forward Error Correction (FEC) and making the protocol less vulnerable to loss of control packets. These amendments will further improve the operational gains associated with the use of P_MUL.

107. Efforts to standardise this protocol in other forums such as NATO and the Internet Engineering Task Force (IETF) is encouraged and supported to the practical extent possible.

SCOPE

108. The objective of ACP 142(A) is to specify and standardise the P_MUL protocol. This document defines the various states of the P_MUL transmitters and receivers when utilising the P_MUL protocol. Within the P_MUL application there are many configurable parameters that can be used to optimise P_MUL for an operational environment. This document does not mandate a particular setting for any parameter nor dictate policy for any nation for conducting a multicast service. Operational guidance is provided within this document to provide examples of how some particular parameters affect the performance of the protocol.

109. P_MUL, as a reliable multicast protocol, requires an underlying connectionless network infrastructure with multicast routing functionality. Current P_MUL implementations, as an example depicted in Figure 1-1, use the User Diagram Protocol (UDP) [UDP80] and Internet Protocol (IP). However its use is not limited to this protocol stack.



* Note: other multicast protocols such as DVMRP or PIM could also be employed

Figure 1-1: An implementation of P_MUL on a UDP/IP Stack

110. This document is the standard that ensures allied interoperability. Vendors are allowed to develop products in accordance with this standard. Note the implementation of dynamic multicasting group formation (Chapter 4) is an optional capability in this edition of ACP 142.

OVERVIEW

111. Within this ACP defining the P_MUL protocol the word "message" is to be understood as a generic data element.

112. This protocol takes advantage of a multicast communication service to transfer messages between different nodes on a single multicast network under both normal (which

means duplex, half or full, oriented communication conditions) and under Emission Control (EMCON) conditions. EMCON or "Radio Silence" condition means that a receiving node is able to receive messages but cannot acknowledge received messages for a relatively long time (e.g. hours or days).

113. Figure 1-2 illustrates a simple multicast scenario, where the same message is transmitted from transmitting node, Node 0, to receiving nodes, Node 1, Node 2 and Node 3. Using a multicast communication service instead of a unicast communication service, only one message is transmitted from Node 0 to the Router, instead of three as required by unicast. This saves transmission of two messages and consequently conserves network bandwidth. Depending on the transmission rate (in some radio networks this is less than 9.6 Kb/s), this saving can be significant. Clearly, the conservation in bandwidth increases with increasing numbers of receiving nodes.

114. The P_MUL protocol may be understood as a transport layer protocol. P_MUL utilises lower layer protocols to transmit its PDUs (Protocol Data Units) over a multicast network.

115. Considering the fact that nodes under EMCON are not allowed to acknowledge messages, they are unable to use a reliable transport protocol like Xpress Transport Protocol (XTP) [XTP95] for the transmission of messages. Therefore, P_MUL makes use of a connectionless transport protocol, such as UDP.

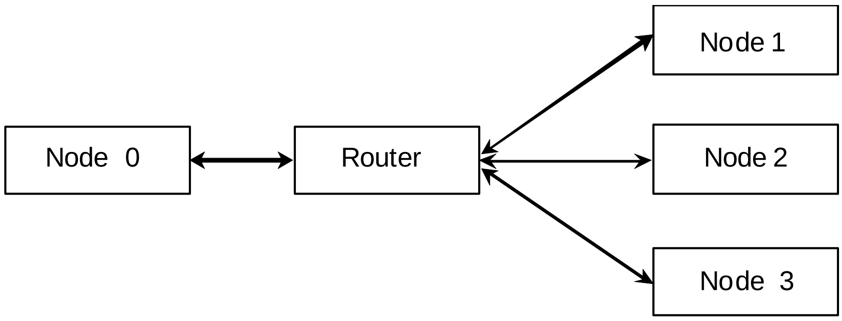


Figure 1-2: Typical Messaging or File Distribution Configuration

116. Although P_MUL is based on a connectionless transport protocol, it provides users with reliable connection-oriented multicast services. It enables the receivers to receive messages while being under EMCON restrictions. It ensures that the transmitter is informed about the timely completeness of the transmission of the messages after the receivers leave the EMCON status and, if required, enables the re-transmission of any messages that were not properly received.

117. The multicast service provided by P_MUL is to some extent "self-contained". It makes use of standard features of multicast routing protocols such as DVMRP, MOSPF and PIM, and can be used with any of them. P_MUL requires that the router be able to support static multicast routing. It also requires that the router or the sub-network interface device be able to support a simplex link to operate in EMCON.

118. It is envisaged that P_MUL will be deployed in a network environment consisting of a few nodes to hundreds of nodes.

119. P_MUL works for fixed multicast groups and can also work for dynamically formed ad hoc multicast groups. For fixed multicast groups each node has knowledge about the

group memberships of one or more multicast groups. Dynamically formed ad hoc multicast groups are used for transmitting single messages.

STRUCTURE OF DOCUMENT

120. Chapter 1 contains an overview of the P_MUL protocol, Definitions and Abbreviations. Chapter 2 describes the structure of the P_MUL Protocol Data Units. Chapter 3 describes the P_MUL messaging processes. Chapter 4 describes the procedures to dynamically form multicast groups. Chapter 5 contains references. Annex A contains examples illustrating the operation of the protocol. Annex B contains a table of predefined protocol parameters, recommended multicast addresses, UDP port numbers, and the checksum algorithm. Annex C gives an operational guidance.

DEFINITIONS

121. The definitions of a set of related, abbreviations and commonly used terminology are provided in the Glossary of Terms.

CONVENTIONS

122. Fields and variable names are made from a string of self-explanatory words connected by underscore(s). Each non-connection word is printed in lower case with the first letter in upper case. The connection words and prepositions are printed in lower cases while abbreviations are printed in upper cases. Examples are Announce_PDU, Message_ID and Expiry_Time. Constants and pre-defined values are made from self-explanatory words all in upper cases, such as ACK_PDU_TIME.

CHAPTER 2

P_MUL PROTOCOL DATA UNITS

GENERAL

201. This chapter describes the protocol data units employed in P_MUL. All integers are transmitted in big endian format.

202. There are two groups of protocol data units (PDUs). The first group consists of PDUs required for the data transfer while the second group consists of PDUs used only for dynamic configuration of multicast groups. If dynamic allocation of multicast groups is not required, then only the first group of PDUs is needed (para 205 to 214).

203. As the P_MUL implementing processes have to track the states of the message transfer (connections) between the transmitting node and all receiving nodes, all PDUs are transmitted using the UDP.

204. Normally, all PDUs are delivered by the transmitting node using the multicast communication service provided in the lower layer. Only in the case where there is only one receiving node, not in EMCON, would the transmitting node switch into unicast addressing mode. All PDUs sent from receiving nodes back to the transmitting node are transmitted in unicast mode.

PDUS FOR DATA TRANSFER

205. The number of PDUs used by P_MUL for data transfer is determined mainly by the requirement that the protocol be able to support communication under EMCON or "Radio Silence" conditions, i.e simplex communication. Under these conditions, one or more nodes are able to receive messages but are not permitted to provide acknowledgements. Such a restriction can exist for long periods of time, e.g. hours or days. P_MUL uses four different PDUs for the data transfer as follows:

a. Transmitter identifies message addresses (Address_PDU)

b. Transmission of message fragments (Data_PDU)

c. Receiver acknowledges message reception (ACK_PDU)

d. Transmitter terminates the transmission of a specific message (Discard_Message_PDU).

ADDRESS_PDU

206. The P_MUL transmitter generates an Address_PDU to announce to the intended recipients of a message transmission and provide the Message_ID. This PDU and the ACK_PDU (para 212 to 214) effect re-transmission control of P_MUL packets. The total list of Destination_Entries has to be a sorted list in increasing order concerning the element Destination_ID of each Destination_Entry. As P_MUL has to observe a maximum PDU size and as the number of Destination_Entries has no maximum value, it is essential that the total address information be able to be split into more than one Address_PDU. To distinguish

between the first, middle and last Address_PDU the MAP field is used. The structure of Address_PDU is depicted in Figure 2-1.

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
```

Length_of_PDU	Priority	MAP	PDU_Type
Total_Number_of_PDUs	Checksum		
Source_ID			
Message_ID (MSID)			
Expiry_Time			
Count_of_Destination_Entries	Length_of_Reserved_Field		
List of Destination_Entries (variable length)			

One Destination_Entry

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
```

Destination_ID
Message_Sequence_Number
Reserved_Field (variable length)

Figure 2-1: Structure of an Address_PDU

207. Description of fields:

Length_of_PDU: The field (2 octets) indicates the total number of octets within this PDU.

Priority: This 1-octet field contains the priority of the message. Priority is defined by the application. A value of 0 denotes the highest priority.

MAP: This 2-bit field specifies whether this Address_PDU is first, middle or last.

The high order bit is set to

0: This is the first one of a set of Address_PDUs.

1: This is NOT the first one of a set of Address_PDUs.

The low order bit is set to

0: This is the last one of a set of Address_PDUs.

1: This is NOT the last one of a set of Address_PDUs.

When both bits are set to 0, it means there is only on Address_PDU.

PDU_Type: This 6-bit field specifies the type of the actual PDU. PDU_Type x '02 denotes an Address_PDU and PDU_Type x '12 denotes an Extra_Address_PDU.

Total_Number_of_PDUs:

These 2 octets hold the total number of Data_PDUs of the message.

Checksum: The checksum is calculated over the entire PDU except the checksum field. The checksum algorithm is:

The checksum field is the 16 bit one's complement of the one's complement sum of all 16 bit words in the header. For purposes of computing the checksum, the value of the checksum field is zero.

Source_ID: This field holds the ID of the sender of this Address_PDU. Source_ID is a unique identifier within the scope of the nodes supporting P_MUL (e.g. the Internet version 4 address of the specific multicast interface of the transmitting node).

Message_ID: MSID is a unique identifier created within the scope of Source_ID by the transmitter.

Expiry_Time: This is the expiry time of the message. This entry allows transmitters and receivers of a message to decide when a message entry is outdated and can be removed from a processing list. This value is set as the number of seconds since 00:00:00 1.1.1970 – Unix Time.

Count_of_Destination_Entries:

These two octets specify the number of destination entries.

Length_of_Reserved_Field:

These 2 octets specify the length of the Reserved_Field. This field shall be set to 0 when not used.

List of Destination_Entries:

This field is an array of destination entries. Its dimension is specified by Count_of_Destination_Entries.

Destination_ID: This field holds a unique identifier identifying a receiving node on the actual multicast network (e.g. the Internet version 4 address of the receiving node). Destination_ID is a unique identifier within the scope of the nodes supporting P_MUL.

Message_Sequence_Number:

This entry holds a message sequence number, which is unique for the sender/receiver pair denoted by Source_ID and Destination_ID. This sequence number is generated by the transmitter consecutively with no omissions and is used by receivers to detect message loss.

Reserved Field: This field is provided for future expansion.

DATA_PDU

208. The P_MUL transmitter generates the Data_PDU to pass each of the message fragments to the intended recipients. This PDU holds the unique identifier of the message, the position of this Data_PDU within the ordered set of all Data_PDUs and a part of the total message. The structure of the Data_PDU is depicted in Figure 2-2.

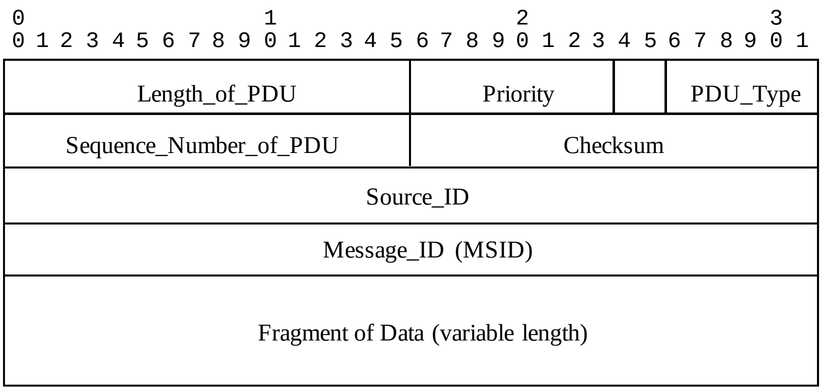


Figure 2-2: Structure of a Data_PDU

209. Description of fields:

Length_of_PDU: The field (2 octets) indicates the total number of octets within this PDU.

Priority: This 1-octet field contains the priority of the message. Priority is defined by the application. A value of 0 denotes the highest priority.

The two bits before the PDU_TYPE field shall be set to 0.

PDU_Type: This 6-bit field specifies the type of the actual PDU. PDU_Type x '00 denotes a Data_PDU.

Sequence_Number_of_PDU: This value specifies the order of the message fragment within the original message, starting from 1.

Checksum: The checksum is calculated over the entire PDU except the checksum field. The checksum algorithm is:

The checksum field is the 16 bit one's complement of the one's complement sum of all 16 bit words in the header. For purposes of computing the checksum, the value of the checksum field is zero.

Source_ID: This field holds the ID of the sender of this Data_PDU and is equivalent to the Source_ID within the corresponding Address_PDU. Source_ID is a unique identifier within the scope of the nodes supporting P_MUL (e.g. the Internet version 4 address of the specific multicast interface of the transmitting node).

Message_ID: MSID is a unique identifier created within the scope of Source_ID by the transmitter. Its value is equivalent to the value of Message_ID of the corresponding Address_PDU.

Fragment of Data: This field of the Data_PDU holds the message or a fragment of the message.

DISCARD_MESSAGE_PDU

210. The Discard_PDU generated by a P_MUL transmitter is used to inform the receiving nodes that the transfer of a specific message has been terminated and no further PDUs of this message will be transmitted. Such situations can arise in the event of hardware error or message obsolescence. PDUs already received are to be discarded by the receiving node. Under these circumstances the transmitting node shall generate a Non-Delivery Report. The structure of the Discard_Message_PDU is depicted in Figure 2-3.

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
```

Length_of_PDU	Priority		PDU_Type
unused	Checksum		
Source_ID			
Message_ID (MSID)			

Figure 2-3: Structure of a Discard_Message_PDU

211. Description of fields:

Length_of_PDU: The field (2 octets) indicates the total number of octets within this PDU.

Priority: This 1-octet field contains the priority of the message. Priority is defined by the application. A value of 0 denotes the highest priority.

The two bits before the PDU_TYPE field shall be set to 0.

PDU_Type: This 6-bit field specifies the type of the actual PDU. PDU_Type x '03 denotes a Discard_Message_PDU.

The sixteen bits before the Checksum field shall be set to 0.

Checksum: The checksum is calculated over the entire PDU except the checksum field. The checksum algorithm is:

The checksum field is the 16 bit one's complement of the one's complement sum of all 16 bit words in the header. For purposes of computing the checksum, the value of the checksum field is zero.

Source_ID: This field holds the ID of the sender of the Discard_Message_PDU and is equivalent to the Source_ID within the corresponding Address_PDU. Source_ID is a unique identifier within the scope of the nodes supporting P_MUL (e.g. the Internet version 4 address of the specific multicast interface of the transmitting node).

Message_ID: MSID is a unique identifier created within the scope of Source_ID by the transmitter. Its value is equivalent to the value of Message_ID of the corresponding Address_PDU.

ACK_PDU

212. This PDU is generated by a receiving node identified by the Source_ID_of_ACK_Sender and is used to inform the transmitting node of the status of one or more messages received. This information is composed as one or more entries of the list of ACK_Info_Entries. Each of these entries holds a message identifier (Source_ID and Message_ID) and a list of Missing_Data_PDU_Seq_Numbers, which may contain a list of those Data_PDUs not yet received. If this list is empty, the message identified by Source ID and Message ID has been correctly received. Figure 2-4 depicts the structure of an ACK_PDU.

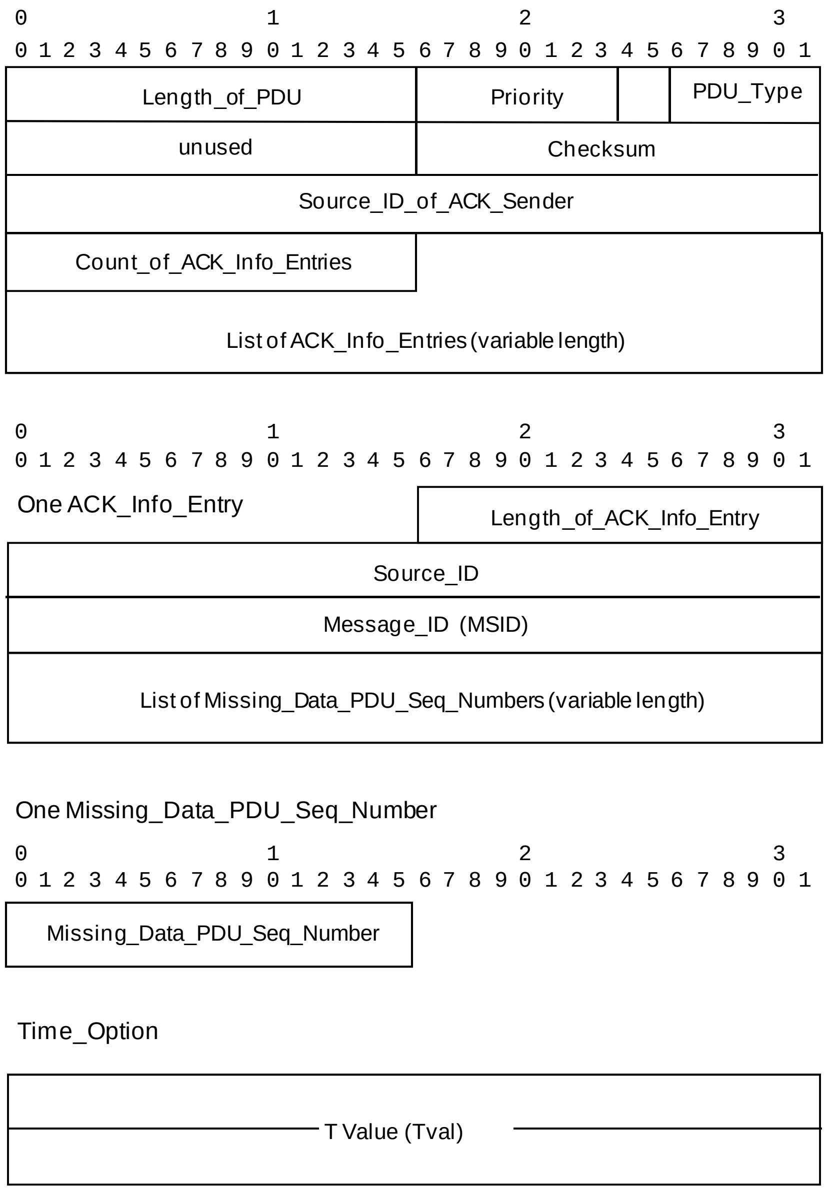


Figure 2-4: Structure of an ACK_PDU

213. Description of fields:

Length_of_PDU: The field (2 octets) indicates the total number of octets within this PDU.

Priority: This 1-octet field contains the priority of the message. Priority is defined by the application. A value of 0 denotes the highest priority.

The two bits before the PDU_TYPE field shall be set to 0.

PDU_Type: This 6-bit field specifies the type of the actual PDU. PDU_Type x '01 denotes an ACK_PDU.

The sixteen bits before the Checksum field shall be set to 0.

Checksum: The checksum is calculated over the entire PDU except the checksum field. The checksum algorithm is:

The checksum field is the 16 bit one's complement of the one's complement sum of all 16 bit words in the header. For purposes of computing the checksum, the value of the checksum field is zero.

Source_ID_of_ACK_Sender:

This field holds the ID of the sender of the ACK_PDU and is equivalent to one of the Destination_IDs described for the Address_PDU. This ID must be unique for the actual multicast network.

Count_of_ACK_Info_Entries:

This field contains the number of ACK_Info_Entries. If there is only one active sender, there will be only one ACK_Info_Entry, although there may be more than one transmitting node on a given multicast network.

List of ACK_Info_Entries:

This entry is an array of ACK_Info_Entries for one or more transmitting nodes of the multicast network. Its dimension is specified by Count_of_ACK_Info_Entries.

ACK_Info_Entry: This field contains the Source_ID of the transmitting node, the Message_ID and a variable length list of the sequence numbers of the Data_PDUs that have not yet been received.

Length_of_ACK_Info_Entry:

This field holds the octet length of each ACK_Info_Entry. It equals 2N+10 where N is the actual number of entries in the list of Missing_Data_PDU_Seq_Numbers of an ACK_Info_Entry. When it equals to 10, the ACK_Info_Entry acknowledges a complete message. Its maximum value is 2MM+10 for an intermediate-list, and 2MM+12 for an end-list (see “List of Missing_Data_PDU_Seq_Numbers” for definitions), where MM is a receiver-configurable parameter (see para 355).

Source_ID: This field holds the Identifier of the transmitting node. Its value is equivalent to the value "Source_ID" of the corresponding Address_PDU.

Message_ID: MSID is a unique identifier created by the sender within the scope of Source_ID by the transmitter. Its value is equivalent to the value of Message_ID of the corresponding Address_PDU.

List of Missing_Data_PDU_Seq_Numbers:

This field contains the list of sequence numbers of those Data_PDUs expected by the receiving node but found to be lost during the transmission.

It can be either an intermediate-list or an end-list in terms of a message transmission or re-transmission. An intermediate-list consists of MM entries and may be used only when a message has more than MM missing Data_PDUs. An intermediate-list is transmitted as soon as a receiving node realises that MM Data_PDUs are lost in a message, before reaching the end of the message transmission or re-transmission. An end-list is transmitted after receiving the last Data_PDU in a message transmission, or the highest numbered Data_PDU expected by the receiver in a message re-transmission. An end-list contains a variable number of entries, up to MM+1. A special form of an end-list is the empty list, known as the message completion acknowledgement, to report that all Data_PDUs in the message has been received correctly.

In an intermediate-list, the sequence numbers, except zeros (see below), are listed in a monotonically increasing order. In an end-list, the sequence numbers are also listed in a monotonically increasing order, until the highest number is reached, which is then followed by the (repeating) lowest missing Data_PDU number to mark the end. In this case if the list has less than MM entries, and if more than one ACK_PDU is required, other previously listed (repeating numbers should also be concatenated, from either end. If more than one ACK_PDU is required, then across the lists the numbers are listed in piecewise monotonically increasing order, except at the "boundaries" where the highest number meets the lowest one. The transmitting node interprets that any Data_PDU(s) between two adjacent ones in the lists are received correctly by this receiving node, modulo the total Data_PDU number in a message. When only one packet has been lost in a message, the list contains the lost sequence number twice, since it is both the highest and the lowest number. For efficiency, a zero may be used to represent a block of consecutive numbers between two listed numbers if no confusion is caused (e.g. 2,3,4,5 may be represented by 2,0,5). (See para 358 for more details)

Time_Option: The Tval field of 8 bytes may be used for reporting the time elapsed (in units of 100ms) from the receiver received the Address_PDU to the receiver sent back the ACK_PDU. The sender may use this time to estimate the time it has taken to transmit the Data_PDUs and to adaptively adjust the parameter “PDU_DELAY” used for congestion control. No synchronization of clocks is required.

214. Normally one ACK_PDU reports the receiving status of one MSID from one Source_ID only. However, one ACK_PDU may contain the receiving status of more than one MSID from more than one Source_ID, especially at the end of an EMCON period.

FORWARD ERROR CORRECTION (FEC)

215. Forward Error Correction and erasure decoding allow the receiver to recover the message more efficiently. FEC is accomplished by generating redundancy to the information using a predetermined algorithm (not defined in this specification). These redundancy packets are used to replace the repetition of data packets when required. The receiver can recover a message once collected sufficient data and redundancy packets. FEC may improve the throughput of both reliable communications with acknowledgements and unreliable communications for EMCON recipients.

216. FEC is applied on message basis, and FEC scheme may vary from message to message. Two new types of FEC related address PDUs are introduced for backward compatibility: FEC_Address_PDU and Extra_FEC_Address_PDU. The structures are depicted below:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
```

Length_of_PDU		Priority	MAP / PDU_Type
Total_Number_of_PDUs		Checksum	
Source_ID			
Message_ID (MSID)			
Expiry_Time			
FEC_Para_Length	FEC ID	FEC Parameters (variable length)	
Count_of_Destination_Entries		Length_of_Reserved_Field	
List of Destination_Entries (variable length)			

One Destination_Entry is

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
```

Destination_ID
Message_Sequence_Number
Reserved_Field (variable length)

Figure 2-5: Structure of a FEC_Address_PDU or an Extra_FEC_Address_PDU

PDU_Type:

This 6-bit field specifies the type of the actual PDU.

PDU_Type x’08’ denotes a FEC_Address_PDU, and PDU_Type x’18’ denotes an Extra_FEC_Address_PDU.

FEC_Para_Length:

This field describes the length of the FEC_Parameters in two-octet units.

FEC_ID:

This field describes the FEC algorithm to be used for the encoding by the transmitter and for the decoding by the receivers. A separate formal definition is required for each FEC scheme with a unique FEC_ID.

FEC_ID = x'00 means no FEC is used.

FEC_Parameters:

In most cases the decoding algorithms need parameter information for correct decoding. This information has to be sent within the field FEC_Parameters. How this field is structured is up to the description of the special FEC algorithm identified by FEC_ID.

Example:

For a systematic R-S erasure code implementation the FEC_Parameter may hold one number only, that is, the number of source packets. The number of encoding packets is identical to the value of Total_Number_of_PDUs and therefore not really necessary.

PDUS TO DYNAMICALLY CREATE MULTICAST GROUPS

217. When a set of nodes intends to deploy multicast message transfers, all nodes have to be connected to a multicast-supporting network. This set consists of a subject "T_Nodes" of potentially transmitting and a subset "R_Nodes" of potentially receiving nodes. The two subsets normally have a non-empty intersection.

218. The communication for this group management scheme is based on a globally known multicast address "GG", which all members of T_Nodes and R_Nodes join. GG is a multicast address used for group management purposes.

219. To allow different communication channels for T_Nodes and R_Nodes it is assumed that there are two well-known ports "TPORT" and "RPORT". All transmitting nodes within T_Nodes read PDUs addressed to GG and TPORT, while all receiving nodes within R Nodes read PDUs addressed to GG and RPORT. (Note: The data transmission itself is carried via a dynamically installed multicast group and two additional ports "DPORT" (Data_Port) and "APORT" (Acknowledgement_Port)).

220. For dynamic installation of a multicast group the following 4 PDUs are defined:

a. Requesting a Multicast Group (Request_PDU)

b. Rejecting a Multicast Group (Request_PDU)

c. Releasing a Multicast Group (Release_PDU)

d. Announcing a Multicast Group (Announce_PDU)

221. The first three PDU types are used only for communications between the transmitter within a transmitting node and all other members of T_Nodes being addressed by the global

group GG with TPORT. The fourth PDU type is issued by a node of T_Nodes, to install a new multicast group and to inform all the affected receiving nodes to join the new group. This PDU is sent to all nodes within the set R_Nodes, being addressed by the global group GG with port RPORT. This PDU is used only after it has been agreed upon among the members of T_Nodes, and the transmitting node is deemed to own that particular multicast group. After the completion of message transfer, the owner of the multicast group must release the multicast group.

REQUEST_PDU, REJECT_PDU AND RELEASE_PDU

222. These three PDU types (Request_PDU, Reject_PDU, and Release_PDU) have the same simple structure. They contain only a few information fields and differ only with respect to a field "PDU_Type". The general structure of these PDUs is depicted in Figure 2-6.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
```

Length_of_PDU	unused		PDU_Type
unused	Checksum		
Source_ID			
Message_ID (MSID)			
Multicast_Group_Address			

Figure 2-6: General Structure of Request_PDU, Reject_PDU and Release_PDU

223. Description of fields

Length_of_PDU: The field (2 octets) indicates the total number of octets within the PDU.

PDU_Type: This 6-bit field specifies the type of the actual PDU. PDU_Type x '05 for a Request_PDU, or x '06 for a Reject_PDU or x '07 for a Release_PDU.

The sixteen bits before the Checksum field shall be set to 0.

Checksum: The checksum is calculated over the entire PDU except the checksum field. The checksum algorithm is:

The checksum field is the 16 bit one's complement of the one's complement sum of all 16 bit words in the header. For purposes of computing the checksum, the value of the checksum field is zero.

Source_ID: This field holds the ID of the sender of this PDU. Source_ID is a unique identifier within the scope of the nodes supporting P_MUL (e.g. the Internet version 4 address of the specific multicast interface of the transmitting node).

Message_ID: MSID is a unique identifier created by the transmitter within the scope of Source_ID.

Multicast_Group_Address:

This field holds the address of a multicast group, to be requested, rejected, or released.

224. Request_PDU: The Request_PDU is distributed by a node wishing to initiate a new multicast group within T_Nodes, using group GG and port TPORT.

225. Reject_PDU: The Reject_PDU is used by a member of T_Nodes, which already "owns" the multicast group. The Reject_PDU is sent to the requesting node in unicast mode.

226. Release_PDU: The Release_PDU is used in the following two situations:

(a) After the sender has received a Reject_PDU,

(b) After a transmission has finished.

This PDU type is used to inform those members of T_Nodes, which senders have relinquished particular multicast addresses.

ANNOUNCEMENT OF A MULTICAST GROUP USING THE ANNOUNCE_PDU

227. The Announce_PDU sends a list of nodes, which are to receive a specific message to the nodes of R_Nodes. As P_MUL has to observe a maximum PDU size and as the number of Destination_IDs has no maximum limit, it is essential that the total list of Destination_IDs be able to be split more than one Announce_PDU. To distinguish between the first, middle, or last Announce_PDU, the MAP field ("More Address_PDUs") is used. The structure of the Announce_PDU is depicted in Figure 2-6.

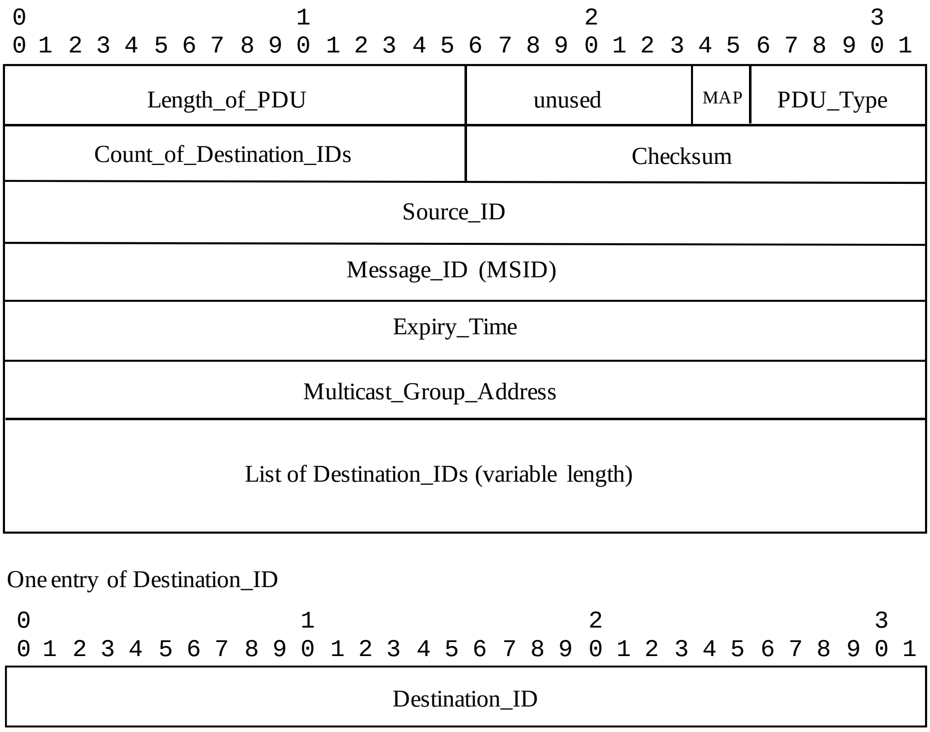


Figure 2-7: Structure of an Announce_PDU

228. Description of fields:

Length_of_PDU: The field (2 octets) indicates the total number of octets within this PDU.

MAP: This 2-bit field specifies whether this Announce_PDU is first, middle or last.

The high order bit is set to '0: this is the first one of a set of Announce_PDUs. '1: This is NOT the first one of a set of Announce_PDUs.

The low order bit is set to

'0: this is the last one of a set of Announce_PDUs.

'1: This is NOT the last one of a set of Announce_PDUs.

When both bits are set to '0, it means there is only one Announce_PDU.

PDU_Type: This 6-bit field specifies the type of the actual PDU. PDU_Type x '04 denotes an Announce_PDU.

Count_of_Destination_IDs:

These 2 octets hold the total number of Destination_IDs within the list of Destination IDs.

Checksum: The checksum is calculated over the entire PDU except the checksum field. The checksum algorithm is:

The checksum field is the 16 bit one's complement of the one's complement sum of all 16 bit words in the header. For purposes of computing the checksum, the value of the checksum field is zero.

Source_ID: This field holds the ID of the sender of this PDU. Source_ID is a unique identifier within the scope of the nodes supporting P_MUL (e.g. the Internet version 4 address of the specific multicast interface of the transmitting node).

Message_ID: MSID is a unique identifier created within the scope of Source_ID by the transmitter.

Expiry_Time: This is the expiry time of the message. This entry allows transmitters and receivers of a message to decide when a message entry is outdated and can be removed from a processing list. This value is set as the number of seconds since 00:00:00 1.1.1970 – Unix Time.

Multicast_Group_Address:

This 4-octet field holds the address of the multicast group announced for a message transfer denoted by Source_ID and MSID.

List of Destination_IDs:

This variable length field contains a list of Destination_IDs of the intended receiving nodes for the message denoted by source_ID and MSID. A Destination_ID may hold the Internet version 4 address of the receiving node.

CHAPTER 3

MESSAGING PROCEDURES

GENERAL

301. This chapter describes the messaging procedures employed between a transmitting node and one or more receiving nodes either in the non-EMCON or EMCON mode of operation, using the P_MUL multicast protocol. These procedures are implemented for every message transmitted. The procedures are divided into three main areas, namely:

- Transmission and re-transmission (see paras 302-325)
- Reception (see paras 326-343)
- Acknowledgement (see paras 344-369)

A set of state diagrams depicting these procedures is provided for information at Annex C.

TRANSMISSION AND RE-TRANSMISSION OF A MESSAGE

302. A node shall initiate the transmission of a message by transmitting an Address_PDU containing a list of nodes that are to receive the message. The transmitting node shall be notified by a "Higher Management Function" about the operation mode of each receiving node, that is, which receiving nodes are in the EMCON mode, and which are in the non-EMCON mode. Based on this information, the transmitting node shall create a list of non-EMCON receiving nodes, from which it will expect to receive ACK_PDUs. If FEC is used, the FEC_Address_PDU (see para 215 & 216) shall be transmitted instead of the Address_PDU.

303. After transmitting the Address_PDU, the transmitting node will transmit the Data_PDU(s), with the Message_ID field set equal to the Message_ID field of the associated Address_PDU. After transmitting the last Data_PDU of a message, the transmitting node shall initialise a "Transmitter Expiry_Time Timer". If one or more of the receiving nodes is in the EMCON mode, the transmitting node shall initialise an "EMCON Re-transmission counter" (EMCON_RTC). The transmitting node shall then enter the non-EMCON or EMCON re-transmission mode (see paras 313and 323) according to the higher management function.

304. After the last Data_PDU is transmitted, an Extra_Address_PDU MAY be transmitted in order to increase the chances of getting the Address_PDU through over low quality channels without having to wait for re-transmission. If FEC is used, the Extra_FEC_Address_PDU (para 216) shall be transmitted instead of the Extra_Address_PDU.

305. The transmitting node may delay the transmission of each Data_PDU according to the PDU_DELAY parameter in order to avoid congestion.

306. The transmission and re-transmission of a message is determined from the data’s original priority and then timestamp (T Value). This allows transmission and re-transmission of high priority data before lower priority data. If data with higher priority arrives for

transmission during an ongoing transmission or an ACK_PDU arrives indicating the need for re-transmission of Data_PDUs with higher priority, the ongoing transmission shall be pre-empted and the higher priority data shall be given precedence.

TRANSMITTER EXPIRY_TIME TIMER

307. This timer indicates the time remaining before the contents of the transmitted messages are considered invalid. It is initialised in accordance with the value transmitted in the Expiry_Time field of the Address_PDU.

308. If one or more of the receiving nodes have not acknowledge the receipt of the complete message when this timer expires the transmitting node will transmit a Discard_Message_PDU with the Message_ID set equal to the Message_ID field in the associated Address_PDU and Data_PDUs.

309. If, after transmitting a Discard_Message_PDU, the transmitting node receives an ACK_PDU indicating partial reception of the message, the transmitting node shall re-transmit the Discard_Message_PDU.

310. If, after having transmitted a Discard_Message_PDU, the transmitting node received an ACK_PDU indicating receipt of the complete message, the transmitting node shall disregard the ACK_PDU.

311. If all the receiving nodes addressed in the list of Destination_Entries of the Address_PDU acknowledge receipt of the complete message before this timer expires, the timer shall be stopped.

EMCON RE-TRANSMISSION COUNTER

312. This counter (EMCON_RTC) indicates the maximum number of times the complete message may be re-transmitted while in the EMCON Re-transmission mode.

NON-EMCON RE-TRANSMISSION

313. The transmitting node may enter this mode when one or more of the receiving nodes are in the non-EMCON mode. On entering this mode the transmitting node shall either:

a. Having transmitted the last Data_PDU, initialize the "Re-transmission timer", or

b. Having left the "EMCON Re-transmission" mode, re-transmits all the indicated missing Data_PDUs, and then initialize the "Re-transmission timer".

314. Every non-EMCON re-transmission session is started with a transmission of an updated Address_PDU, listing only those receiving nodes that still need to receive the consequent re-transmitted data.

RE-TRANSMISSION TIMER

315. The re-transmission timer (TRANSMISSION TIME) shall be initialized after the transmitting node has sent the last Data_PDU of the message.

316. When this timer is running, the transmitting node shall :

a. Collect all missing Data_PDU listed in received ACK_PDUs

b. Delete Destination_IDs from the list of Destination_IDs, for all complete ACK_PDUs received

317. If the transmitting node has received at least one ACK_PDU from all of those receiving nodes before this timer is triggered, or this timer expires the transmitting node shall either:

a. If all the ACK_PDUs indicate that the complete message has been received (i.e. the Destination_IDs list is empty), the transmitting node shall stop this timer, send an Address_PDU with an empty list of Destination_Entries and initializes the End Session timer. If at this stage there are receiving nodes in the EMCON mode, the transmitting node shall enter the EMCON Re-transmission node.

b. If at least one DATA_PDU is missing, the transmitting node shall transmit an updated Address_PDU, wait for a configurable period of time and then after this delay resend all missing Data_PDUs.

318. When this timer expires, it shall be reset with a value greater than the previous one by a configurable BACK_OFF_FACTOR.

RECEIPT OF MESSAGE COMPLETE ACK_PDU

319. See paragraphs 315to 318.

END SESSION TIMER

320. This timer shall be initialized after the first Address_PDU containing an empty list of Destination_Entries has been sent. This timer shall be stopped when transmitter Expiry Time timer expires or stops.

321. If the transmitting node has not received any ACK_PDU after transmitting an Address_PDU containing an empty list of Destination_Entries and this timer expires then this timer is not restarted. If the particular multicast group was one that was dynamically created, the transmission node shall transmit a Release_PDU to all nodes of that multicast group to release the multicast group.

322. If the transmitting node has received some ACK_PDU after transmitting an Address_PDU containing an empty list of Destination_Entries and this timer expires then the transmitting node shall re-transmit and Address_PDU containing an empty list of Destination_Entries and reset this timer.

EMCON RE-TRANSMISSION

323. The transmitting node may enter EMCON Re-transmission mode when any receiving nodes are in the EMCON mode. Having transmitted the last Data-PDU, the transmitting node shall initialise the "EMCON Re-transmission Timer" (EMCON_RTI).

EMCON RE-TRANSMISSION TIMER

324. If this timer expires, implying that since the last EMCON re-transmission all the receiving nodes in the EMCON mode have not entered the non-EMCON mode (by sending ACK_PDUs), and the "EMCON Re-transmission Counter" has not exceeded its maximum, the transmitting node shall re-transmit the Address_PDU and all Data_PDUs not already acknowledged. The transmitting node shall re-initialise this timer and increment the "EMCON Re-transmission Counter". The transmitting node shall wait until either:

a. All the receiving nodes in EMCON mode respond with an ACK_PDU at which point the transmitting node shall enter the "Non-EMCON Re-transmission" mode, or

b. The "Transmitter Expiry_Time timer" expires at which point the transmitting node shall transmit a Discard_Message_PDU.

325. If all of the receiving nodes in the EMCON mode respond (entering non-EMCON mode) with an ACK_PDU indicating partial or complete reception of the message before this timer expires, the transmitting node shall stop the timer and enter the "Non-EMCON Re-transmission" mode (see paras 313& 314).

RECEPTION OF A MESSAGE

326. A receiving node shall enter the "Reception of a Message" mode when it receives either an Address_PDU or a Data_PDU.

RECEIPT OF AN ADDRESS_PDU

327. On receipt of an Address_PDU the receiving node shall first check whether the Address_PDU with the same tuple "Source_ID, MSID" has already been received.

328. If such an Address_PDU has already been received the receiving node shall check whether it has previously sent a message complete ACK_PDU for this message.

a. If it has sent a message complete ACK_PDU, then it shall either:

(i) If its own ID is not in the list of Destination_Entries, it knows that its own ACK_PDU has been successfully received by the transmitting node. Consequently, the receiving node can release all information about this message and its membership of the dynamically created multicast group, and can then discard the Address_PDU, or

(ii) If its own ID exists in the list of Destination_Entries, re-transmit the message complete ACK_PDU, and discard the Address_PDU, or

b. If it has not sent a message complete, ACK_PDU, discard the Address_PDU and wait for remaining Data_PDUs. The receiving node shall check (see para 348) if it shall enter the “Acknowledgement of a Message” mode (see para 345 -347) or remain in the “Reception of a Message” mode (see para 326).

329. If the Address_PDU has not been previously received the receiving node shall either:

a. If its own ID is not in the list of Destination_Entries, the receiving node shall discard the Address_PDU, or

b. If its own ID is in the list of Destination_Entries, determine whether it has previously received any Data_PDUs associated with this Address_PDU.

(i) If there are no Data_PDUs associated with this Address_PDU, the receiving node shall create a message entry and wait transmission of associated Data-PDUs.

(ii) If there are Data_PDUs associated with this Address_PDU, the receiving node shall update the status of the Data_PDU entry (see paras 336 to 339) to a message entry. The receiving node shall stop the "Unidentified_Data_PDU_Validity_Timer" (see paras 340 to 341), and initialise the "Receiver Expiry_Time Timer". The receiving node shall determine whether to enter the "Acknowledgement of a Message" mode (see paras 345 to 347).

c. If its own ID is in the list of Destination_Entries, and if the receiving node is in non EMCON mode the receiving node may start the “Last_PDU” timer.

RECEIPT OF AN EXTRA_ADDRESS_PDU

330. On receipt of an Extra_Address_PDU, the receiving node shall check whether the Extra_Address_PDU is a duplicate of a previously received Address_PDU except for PDU-type field.

a. If the Extra_Address_PDU is a duplicate of a previously received Address_PDU, the receiving node shall discard the Extra_Address_PDU and, if in non EMCON mode, may enter in “ACKnowledgment of a message” (see para 344).

b. If the Extra_Address_PDU is not a duplicate of a previously received Address_PDU, the receiving node shall change the PDU_Type to Address_PDU and shall enter in “Receipt of an Address_PDU” (see paras 327 to 329).

RECEIVER EXPIRY_TIME TIMER

331. This timer indicates the time remaining before the contents of the received message are considered invalid. It is initialised in accordance with the value received in the Expiry_Time field of the Address_PDU.

332. If this timer expires before the receiving node has received all the Data_PDUs associated with a message, the receiving node shall discard the associated Data_PDUs and Address_PDU.

RECEIVER LAST_PDU TIMER

333. This timer indicates the time remaining before the receiving node expects the reception of the last Data_PDU (see paras 349 to 350), which will trigger the generation of an

ACK_PDU. If the last Data_PDU is lost, this timer will make sure that an ACK_PDU is generated.

334. This timer should be updated with a dynamic computed value each time a Data_PDU is received.

335. This timer shall be restarted when last Data_PDU is received.

RECEIPT OF A DATA_PDU

336. On receipt of a Data_PDU the receiving node shall first check whether the Data_PDU has already been received.

337. If the Data_PDU has already been received the receiving node shall either:

a. if the receiving node has not received a duplicate of the associated Address_PDU (see paras 327to 329) and it has previously sent a "message complete" ACK_PDU for this message, re-transmit the "message complete" ACK_PDU and discard the Data_PDU, or

b. otherwise, discard the Data_PDU. The receiving node shall check (see para 348) if it shall enter the “Acknowledgement of a Message” mode (see para 345 to 347) or remain in the “Reception of a Message” mode (see para 326).

338. If the Data_PDU has not been previously received, the receiving node shall check whether it has received the associated Address_PDU.

a. If the associated Address_PDU has been received and a message entry exists, the receiving node shall update the status of the message entry and determine whether to enter the "Acknowledgement of a Message" mode (see paras 345to 347). If the associated Address_PDU has been received but no message entry exists, the receiving node shall discard the Data_PDU.

b. If the associated Address_PDU has not yet been received the receiving node shall check whether there is a Data_PDU entry associated with the Source_ID and Message_ID contents of the received Data_PDU. If there is no Data_PDU entry associated with this Data_PDU, the receiving node shall create a Data_PDU entry and await transmission of the associated Address_PDU. In addition, the receiving node shall initialise a "Unidentified_Data_PDU_Validity_Timer". If there is a Data_PDU entry associated with this Data_PDU, the receiving node shall update the status of the Data_PDU entry and await transmission of the associated Address_PDU.

339. When an FEC scheme is used, on receipt of sufficient Data_PDUs, the decoding process starts and the message is to be recovered from those Data_PDUs using the agreed FEC algorithm. When FEC is used, the definition of "receiving all Data_PDUs" in the following paragraphs means sufficient Data_PDUs are received so that the message is able to be recovered.

UNIDENTIFIED_DATA_PDU_VALIDITY_TIMER

340. This timer indicates the time remaining before the Data_PDUs in a Data_PDU entry are no longer considered valid and can therefore be discarded.

341. If the timer expires before the receiving node receives either the Address_PDU or a Discard_Message_PDU associated with the Data_PDUs in the Data_PDU entry, the receiving node shall discard all Data_PDUs.

ENTRY TO "ACKNOWLEDGEMENT OF A MESSAGE"

342. Having updated the status of a message entry, the receiving nodes shall either:

a. If in non-EMCOM mode check (para 348) if the receiving mode shall enter in the “Acknowledgement of a Message” (para 344) or remain in the “Reception of a Message” mode (para 326) or,

b. If in EMCON mode, check whether all the Data_PDUs for the message have been received. If there are missing Data_PDUs the receiving node shall remain in the "Reception of a Message" mode. If all the Data_PDUs have been received, the message shall be tagged as complete and ready for acknowledgement when the non-EMCON mode is entered. The completed message can then be passed up to the "Higher Layer Application". The receiving node shall remain in the "Reception of a Message" mode.

RECEIPT OF A DISCARD_MESSAGE_PDU

343. If a receiving node receives a Discard_Message_PDU, it shall discard all the PDUs (i.e. data and address) associated with the message identified by the combination of the Source_ID and Message_ID fields in the Discard_Message_PDU, and stop any associated timers. If a special multicast group has been dynamically created for this specific message, the receiving node shall, delete all corresponding Data_PDUs and clear all related timers.

ACKNOWLEDGEMENT PROCEDURES

344. The "Acknowledgement of a Message" mode procedures shall be dependent on whether the receiving node received the messages in the non-EMCON mode (see paras 345 to 361), or **leaving** the EMCON mode (see paras 362 to 369).

ACKNOWLEDGEMENT OF A MESSAGE

345. The "Acknowledgement of a Message" mode can only be entered by a receiving node that is in the non-EMCON mode of operation.

346. To avoid the problem of acknowledgement implosion at the message-transmitting site, each transmission of an ACK_PDU is delayed by a randomly determined period of time.

347. ACK_PDUs shall be processed and transmitted in the order of the priority and timestamp (T Value) of the corresponding received message.

CRITERIAS FOR ENTERING THE "ACKNOWLEDGEMENT OF A MESSAGE" MODE

348. A receiving node, if operating in the non-EMCON mode, shall enter the "Acknowledgement of a Message" mode:

a. When the last Data_PDU of a message is received for the first time, or

b. When the expected highest numbered (previously missing) Data_PDU of a message is received, in a re-transmission session, or

c. When there are MM missing Data_PDUs (see paras 212 to 214 and 355 to 361), or

d. When an Address_PDU is received and a message complete ACK_PDU has already be sent, or

e. When the Last_PDU timer expires.

LAST DATA_PDU RECEIVED

349. In an initial transmission of a message, if the last Data_PDU of a message has been received, the receiving node shall determine whether there are any missing Data_PDUs in the message. In a re-transmission session, this happens once the highest numbered missing Data_PDU has been received.

350. If the Last_PDU timer has been started this timer shall be restarted

LAST_PDU TIMER EXPIRES

351. If the Last_PDU timer expires the receiving node shall check if there are any missing Data_PDU(s) (see paras 352 to 353).

MISSING DATA_PDUS

352. If there are missing Data_PDUs the receiving node shall either:

a. If no ACK_PDU associated with this message has been transmitted, transmit an ACK_PDU listing which Data_PDUs are missing, and return to the "Reception of a Message" mode (refer para 326), or

b. If an ACK_PDU associated with this message has been transmitted, return to the "Reception of a Message" (refer para 326) mode.

353. The Last_PDU timer shall be restarted.

RECEIVED COMPLETE MESSAGE

354. If there are no missing Data_PDUs, the receiving node shall transmit an ACK_PDU indicating that the message is complete, and initializes an ACK_PDU Timer (see paras 367 to 369). When the receiving node receives an Address_PDU in which the receiving node is absent, it shall stop the "Receiver Expiry_Time Timer" associated with this message and stop the Last_PDU timer. The complete message shall be passed up to the "Higher Layer

Application" when the "Receiver Expiry_Time Timer" is stopped or on completion of a transmission of the ACK_PDU.

MAXIMUM NUMBER OF MISSING DATA_PDUS (MM)

355. MM is a receiver-configurable parameter defined as the maximum number of missing Data_PDUs a receiving node should acknowledge (negatively) at a time. MM is also the maximum number of Missing_Data_PDU_Seq_Numbers an ACK_Info_Entry of an ACK_PDU can carry in an intermediate-list (see para 212). However, in an end-list (at the end of a message transmission or re-transmission), the maximum number of entries in one ACK_Info_Entry is MM+1.

356. A message may consist of more than MM Data_PDUs. Therefore a receiving node may miss more than MM Data_PDUs before receiving the last Data_PDU of a message. A re-transmission session may also consist of more than MM Data_PDUs and a receiving node (in non-EMCON mode) may miss more than MM of them before receiving the highest numbered Data_PDU as expected from this re-transmission session. The receiving node shall construct and transmit an ACK_Info_Entry of an ACK_PDU for every MM missing Data_PDUs that it misses, until the last Data_PDU of the message, or the highest numbered Data_PDU expected is received.

357. Once the expected last Data_PDU in the message has been received, the receiving node will transmit an ACK_PDU with an end-list, listing the remaining N missing Data_PDUs, where $0< N < MM$, followed by the (previously reported) lowest missing Data_PDU sequence number. If the total number in the list N+1 is less than MM+1, then the receiving node may fill in with previously reported MM-N numbers as a repeating measure for better reliability.

358. If the receiving node has sent a number of intermediate lists and since then has found no further missing Data_PDU when the last expected Data_PDU of the message is received (N = 0), it should mark the end of the overall list by sending another ACK_PDU. This ACK_PDU contains a list of the highest and lowest missing Data_PDU numbers as previously reported. Again, this list can be filed with MM-1 repeating missing Data_PDU sequence numbers to make up MM+1 for better reliability. In order to decrease the possibilities of misinterpretations of received Data_PDUs due to loss of ACK_PDUs (or NACKs), the last missing sequence number of the previous ACK_PDU (NACK) SHALL be repeated in the next ACK_PDU.

359. Example: Consider an example with MM at 3 and a client that has been sent 20 packets, and has not received packets 5, 7, 11, 13, 15, 16, 17. He shall then send the following four ACK_PDUs with sequence numbers as indicated below. It will then be easy for the transmitter to see if an ACK_PDU is lost.

- 5,7,11
- 11,13,15
- 15,16,17
- 17,5

360. When all of the Data_PDUs of a message have been received correctly, the receiving node will transmit a message completion ACK_PDU, with which the ACK_Info_Entry has

an empty list of Missing_Data_PDU_Seq_Numbers, and the Length_of_ACK_Info_Entry equals 10.

361. On completion of transmission of every ACK_PDU, the receiving node shall return to the "Reception of a Message" mode.

RECEIVING NODE LEAVING EMCON

362. As soon as a receiving node changes its operation mode from EMCON to non-EMCON, it shall acknowledge each message. Upon entering the non-EMCON mode it shall determine whether it has received the last Data_PDU of a message.

Note: The following three procedures are relevant only for those messages (complete or partial) received while in the EMCON mode. All new messages received before the change into the EMCON mode or after the change to the non-EMCON mode of operation are acknowledged in accordance with the procedures described in paras 345 to 347.

LAST DATA_PDU RECEIVED

363. If the last Data_PDU (and Address_PDU) of a message have been received or the receiver Last_PDU timer has expired, the receiving node shall determine whether there are any missing Data_PDU from the message.

MISSING DATA_PDUS

364. If any Data_PDUs are missing, the receiving node shall transmit an ACK_PDU listing the Data_PDUs, which are missing, and initialise an associated "ACK_PDU timer" of the open message (see paras 367 to 369). If there are N missing Data_PDUs and N > MM, the receiving node shall transmit at least [N/MM] ACK_PDUs in order to indicate the total number of missing Data_PDUs, where [x] means the smallest integer no less than x. No ACK_PDU shall indicate more than MM new missing Data_PDUs (see paras 355 to 361). The receiving node shall initialise the associated "ACK_PDU Timer" with each ACK_PDU transmitted.

365. On completion of transmission of the ACK_PDU(s) the receiving node shall then return to the "Reception of a Message" mode.

RECEIVED COMPLETE MESSAGE

366. If there is no missing Data_PDU, the message is tagged as complete (see para 342.b), and the receiving node shall transmit a message completion ACK-PDU (para 360) with an empty list of Missing_Data_PDU_Seq_Numbers and initialise the "ACK_PDU Timer". When the receiving node receives an Address_PDU in which the receiving node is absent it shall stop the "Receiver_Expiry_Time Timer". The complete message shall be passed up to the “Higher Layer Application” when the "Receiver_Expiry_Time Timer" is stopped or on completion of a transmission of the ACK_PDU.

ACK_PDU TIMER

367. There is one ACK_PDU Timer for each message that is not yet fully received. The "ACK_PDU Timer" shall be initialized for every ACK_PDU transmitted by a receiving node in the non-EMCON mode, having previously been in the EMCON mode.

368. If the receiving node receives a response to a transmitted ACK_PDU from the transmitting node in the form of the requested missing Data_PDUs or an Address_PDU, it shall stop the associated "ACK_PDU Timer".

369. If the receiving node does not receive a response to the transmitted ACK_PDU(s) from the transmitting node in the form of the requested missing Data_PDUs or an Address_PDU, and an "ACK_PDU Timer" expires, the receiving node shall re-transmit the associated ACK_PDU(s) and re-initialise the timer.

CHAPTER 4

MULTICAST GROUP FORMING PROCEDURES

GENERAL

401. Multicast group formation procedures can be statically or dynamically assigned. Statically assigned groups require an external authority to establish and manage these groups. P_MUL supports the dynamic formation of multicast groups, allowing groups to be established to support a single-message transmission.

402. The rest if this chapter deals with procedures for dynamic multicast group formation. Note that the implementation of dynamic multicasting group is an optional capability in this edition of ACP 142

403. Dynamic instantiation of multicast groups is a useful means of reducing the overall network load within a multicast network, especially in cases when the sender of a message or file has precise knowledge of the addresses of the intended receiving nodes.

404. Joining and leaving of a multicast group will influence the distribution tree for the multicast group. This distribution tree is heavily dependent on the dialogue between neighbouring routers. Therefore the "dynamic multicast group forming procedure" can only be employed efficiently if none of the network links between the transmitting node and the intended receiving node is under EMCON condition at the time of group formation.

REQUEST FOR A MULTICAST GROUP

405. First the transmitting node has to determine a multicast address for the multicast group which shall be dynamically installed. This multicast address can be chosen by different methods, including:

a. Selection from a set of multicast addresses, which is predefined and mutually exclusive for each member of T_Nodes.

b. Random choice from a predefined set of multicast addresses.

c. On behalf of MADCAP (Multicast Address Dynamic Client Allocation Protocol) from a multicast address allocation server [MADCAP99].

406. When a member of T_Nodes node receives a Request_PDU, it checks whether this multicast group is in use. If so, it will reject this group by sending a Reject_PDU, otherwise it does not respond. This is known as the "Silent Procedure" which aims to reduce the network load.

REJECTING A MULTICAST GROUP

407. If a transmitting node detects that another transmitting node is requesting a multicast address, which it is already using, it should reject this request by sending a Reject_PDU.

408. If the requesting node receives such a Reject_PDU it must revoke its request by the transmission of a corresponding Release_PDU, and has to renew its request with a different multicast address.

409. If the node detecting this collision is under EMCON condition, this rejection is impossible. Therefore, the use of the requested multicast group will result in some additional network load.

410. The same result as in para 409 above will occur, if the particular transmitting node, which already owns the requested multicast group, does not receive the Request_PDU, or if the Reject_PDU it generates is not received by the requesting node.

411. If all members of T_Nodes keep track of the ownership of requested multicast addresses, then the rejection would rarely happen.

RELEASE OF A MULTICAST GROUP

412. A transmitting node will release a once-requested or used multicast address in the following two situations:

a. After having received a Reject_PDU, or

b. After completion of a message transmission.

ANNOUNCEMENT TO JOIN A MULTICAST GROUP

413. After the transmission of the Request_PDU the requesting node waits a certain time (WAIT_FOR_REJECT_TIME) to receive Reject_PDUs from members of T_Nodes. If no Reject_PDU has been received when this time has expired, the requesting node announces the requested multicast address to all receiving nodes. Once a multicast group has been announced, the late reception of a corresponding Reject_PDU will be ignored.

414. This information is sent by the transmitting node via the Announce_PDU to the global group GG and port RPORT.

415. The Announce_PDU may be transmitted multiple times as it is essential for subsequent data transmission. The number of transmission times of Announce_PDU is specified by ANNOUNCE_CT.

416. As soon as a member of R_Nodes has received the Announce_PDU, it decides whether it is a member of the announced group, based on whether it is listed in the list of Destination_IDs. If it is not in the list if Destination_IDs it can ignore the Announce_PDU; otherwise it must join the multicast group denoted by the Announce_PDU.

417. The announcing node must wait a certain period of time (ANNOUNCE_DELAY) until all routers within the multicast tree acquire information about the group memberships of those nodes in the list of Destination_IDs. After this time has expired, data transfer may begin.

418. After this announcing phase the transmitting node assumes that the intended receiving nodes have received the Announce_PDU and therefore have joined the announced multicast group with Multicast_Group_Address. It will start the first data transmission phase

by transmitting the Address_PDU(s) and the Data_PDU(s). It will then wait for the ACK_PDU(s). Before the transmitting node restarts a re-transmission phase it has to check whether it can be sure that all intended receiving nodes had received the Announce_PDU. This decision can be made upon the fact, whether it received an ACK_PDU from each intended receiving node. If not the re-transmission of the eventually updated Announce_PDU according to the above rules concerning ANNOUNCE_DELAY and ANNOUNCE_CT followed by the next data transmission phase.

CHAPTER 5

REFERENCES

[FS1037] US Federal Standard, FS1037C, "Telecommunications: Glossary of Telecommunication Terms", August 7, 1966

[MADCAP99] S R Hanna, B V Patel, and M Shah, IETF RFC 2730, "Multicast address dynamic client allocation protocol (MADCAP)", December, 1999

[UDP80] J Postel: "User Datagram Protocol", IETF RFC768, August 1980.

EXAMPLES

INTRODUCTION

The following examples are designed to illustrate how the defined PDUs are used between the different nodes on a multicast network and to give an impression of how the protocol works. Protocol exchanges are described in a function call type notation.

EXAMPLE OF AGREEMENT ABOUT NEW MULTICAST GROUPS

a. This example describes how a new multicast group is agreed among transmitting nodes and announced to the receiving nodes.

b. It is assumed that M0 is the transmitting node, which sends one message to nodes M1, M2, M3 and M4. Only M3 and M4 are assumed to be under EMCON.

c. Firstly, M0 selects a new multicast address (e.g. 240.1.2.3) and transmits a corresponding Request_PDU:

Request_PDU (source_ID =M0, MSID = 9876,
Multicast_Group_Address = 240.1.2.3)

d. It is assumed that a node within the network already "owns this multicast address. This node will detect this collision and will send M0 (unicast mode) a corresponding Reject_PDU.

Reject_PDU (source_ID =M0, MSID = 9876,
Multicast_Group_Address = 240.1.2.3)

e. As soon as M0 receives this Reject_PDU, it releases its old request by transmitting a corresponding Release_PDU, and starts a new request with another multicast address (e.g. 240.1.2.4).

Release_PDU (source_ID =M0, MSID = 9876,
Multicast_Group_Address = 240.1.2.3)

Request_PDU (source_ID =M0, MSID = 9876,
Multicast_Group_Address = 240.1.2.3)

f. It is assumed that this multicast address is not already in use, or that the multicast address is owned by a node in EMCON. Therefore, after the timer WAIT_FOR_REJECT_TIME expires, and M0 has not received any correspondence Reject_PDUs, M0 transmits the following Announce_PDU:

Announce_PDU (source_ID =M0, MSID = 9876,
Expiry_Time = 1234567890,
Multicast_Group_Address = 240.1.2.4)
Count_of_Destination_IDs = 4
Destination_IDs = (M1, M2, M3, M4))

g. After the transmission of this Announce_PDU any further reception of a Reject_PDU will be discarded. M0 starts the timer ANNOUNCE_DELAY to enable the intended receiving nodes to join the announced multicast group and to enable the routers within the network to build the new multicast routing structure. To avoid packet loss of the Announce_PDU, the transmission of this packet will be repeated until ANNOUNCE_DELAY expires.

EXAMPLE OF DATA TRANSFER

a. This example illustrates how data transfer is coordinated.

b. Continuing the example in para A01, it is assumed that M0 is the transmitting node, which sends one message to the nodes M1, M2, M3 and M4 (Figure A-1). Only M3 and M4 are assumed to be under EMCON. The message length is assumed to be a little larger than the maximum PDU size, which means that the message (Data) is split into two fragments. It is also assumed that until now M0 has sent 99 messages to M1, 77 messages to M2, 10 messages to M3 and 14 messages to M4.

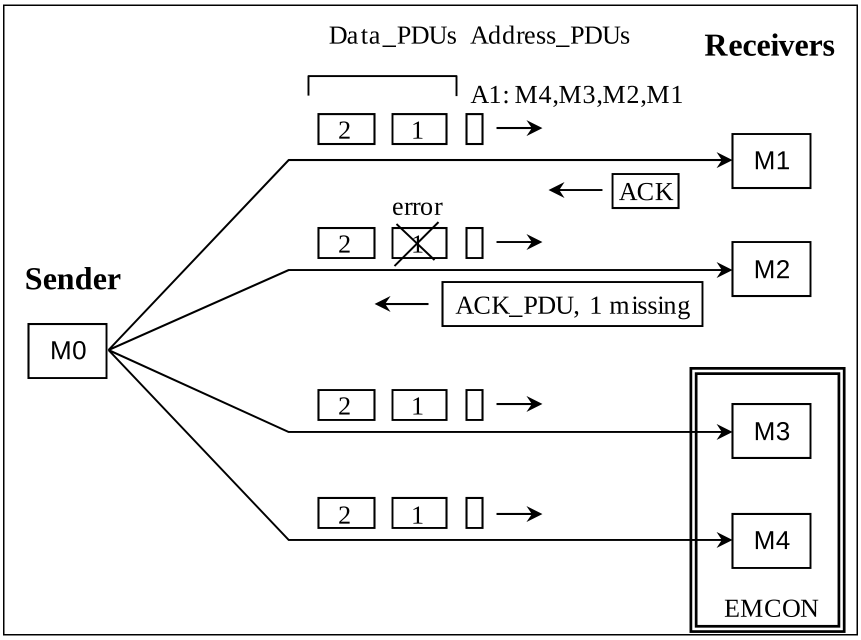


Figure 5-1: M0 sends a message to M1 – M4 in multicast group, M1 and M2 acknowledge

c. M0 constructs an Address_PDU and the two Data_PDUs. It sends all three PDUs to the multicast network.

Address_PDU (source_ID =M0, MSID = 9876,
Expiry_Time = 1234567890,
Total_Number_of_PDUs = 2,
Count_of_Destination_Entries = 4,
Length _of_Reserved_Field = 0,
Destination_Entries = ((M1, 100), (M2, 78), (M3, 11), (M4, 15))),

Data_PDU (Source_ID = M0, MSID = 9876,
Sequence_Number_of_PDU = 1, Data = First N octets of message),

Data_PDU (Source_ID = M0, MSID = 9876,
Sequence_Number_of_PDU = 2, Data = remaining octets of message)

d. Assuming M1, M2, M3 and M4 receive the Address_PDU, M1, M3 and M4 receive the Data_PDUs without failures, while M2 receives the first Data_PDU incorrectly and the second one correctly. As M3 and M4 are in EMCON mode, they cannot send any ACK_PDU. M1 sends a completion ACK_PDU, while M2 sends an ACK_PDU indicating that the first Data_PDU is missing, and that is all is missing (figure A-1).

ACK_PDU (Source _ID_of_ACK_Sender = M1,
Count_of_ACK_Info_Entries = 1.
ACK_Info_Entry = (Length_of_ACK_Info_Entry = 10,
Source_ID = M0, MSID = 9876))

ACK_PDU (Source _ID_of_ACK_Sender = M2,
Count_of_ACK_Info_Entries = 1.
ACK_Info_Entry = (Length_of_ACK_Info_Entry = 14,
Source_ID = M0, MSID = 9876,
Missing_Data_PDU_Seq_Number = 1,
Missing_Data_PDU_Seq_Number = 1)

e. M0 has to re-transmit the first part of the message for M2, because M2 marked this PDU as missing. As M0 has received the completion ACK_PDU of M1, M1 is deleted from the list of Destination_Entries.

Address_PDU (source_ID =M0, MSID = 9876,
Expiry_Time = 1234567890, Total_Number_of_PDUs = 2,
Count_of_Destination_Entries = 3,
Length _of_Reserved_Field = 0,
Destination_Entries = ((M2, 78), (M3, 11), (M3, 11), (M4, 15)))

Data_PDU (Source_ID = M0, MSID = 9876,
Sequence_Number_of_PDU = 1, Data = First N octets of message)

f. Assuming that M2 will receive this PDU correctly, M2 will acknowledge with the following message complete ACK_PDU (Figure A-2):

ACK_PDU (Source _ID_of_ACK_Sender = M2,
Count_of_ACK_Info_Entries = 1.
ACK_Info_Entry = (Length_of_ACK_Info_Entry = 10,
Source_ID = M0, MSID = 9876))

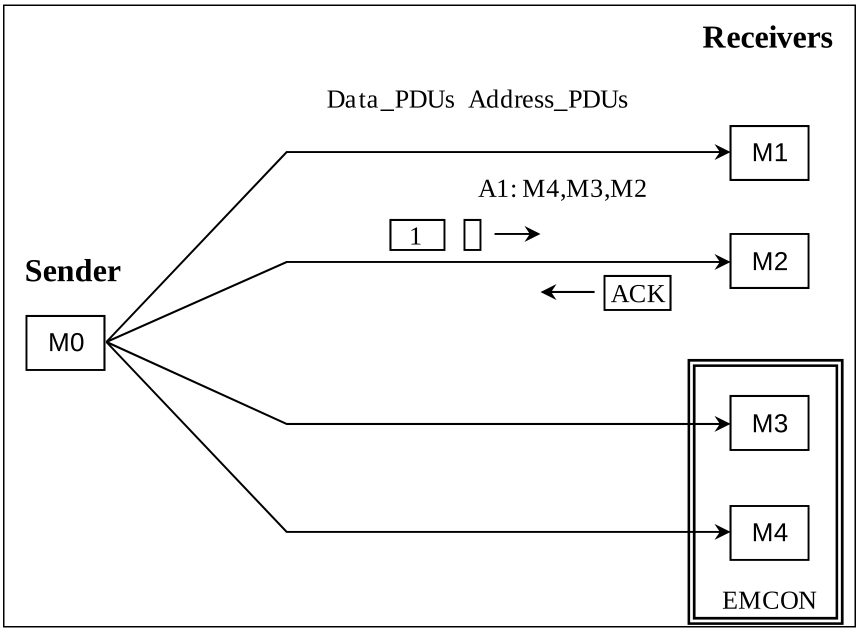


Figure 5-2: Corrupted Data_PDU 1 resent to M2, M2 acknowledges receipt of message

g. As M0 has received the completion ACK_PDU of M2, M2 will be deleted from the list of Destination_Entries.

h. Supposing that M3 has left the EMCON situation (figure A-3), M3 will send its message complete ACK_PDU to M0 as:

ACK_PDU (Source _ID_of_ACK_Sender = M3,
Count_of_ACK_Info_Entries = 1.
ACK_Info_Entry = (Length_of_ACK_Info_Entry = 10,
Source_ID = M0, MSID = 9876))

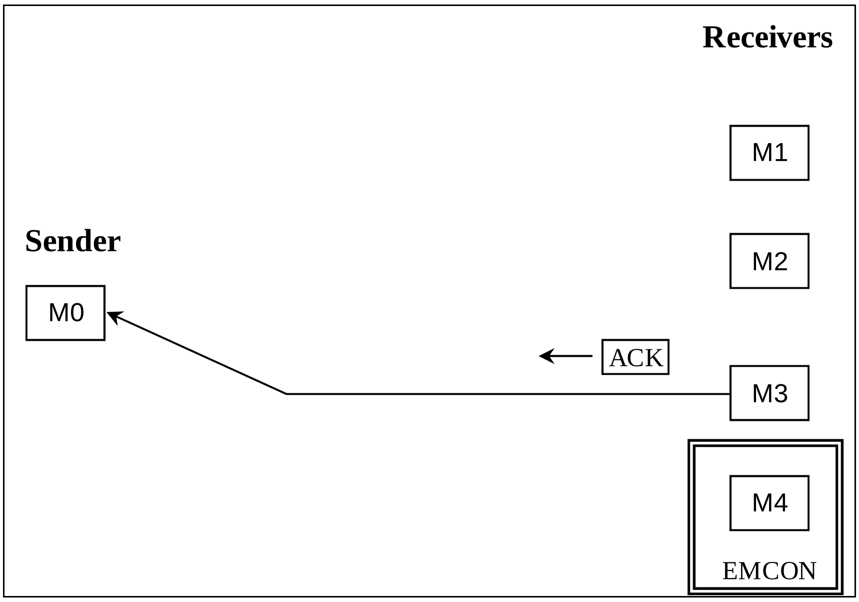


Figure 5-3: M3 leaves EMCON and acknowledges correct receipt of message

i. Assuming that an EMCON Re-transmission for M4 takes place, M0 will re-transmit the total message. As M1, M2 and M3 already have completely acknowledged the message, the Address_PDU will hold only one Destination_Entry for M4 (Figure A-4).

Address_PDU (source_ID =M0, MSID = 9876,
Expiry_Time = 1234567890, Total_Number_of_PDUs = 2,
Count_of_Destination_Entries = 1,
Length _of_Reserved_Field = 0,
Destination_Entries = (M4, M15))

Data_PDU (Source_ID = M0, MSID = 9876,
Sequence_Number_of_PDU = 1, Data = First N octets of message)

Data_PDU (Source_ID = M0, MSID = 9876, Sequence_Number_of_PDU = 2,
Data = Remaining octets of message)

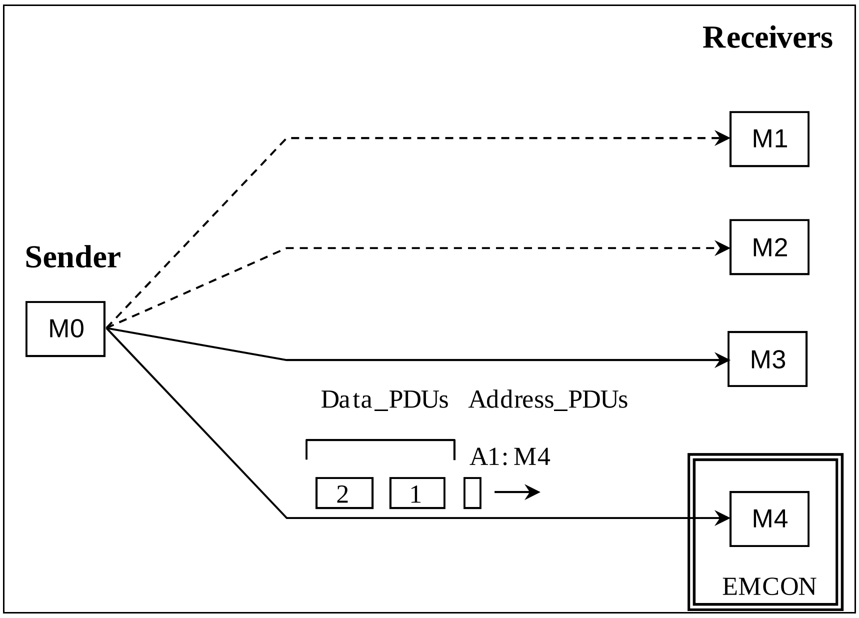


Figure 5-4: Re-transmission of message to M4 occurs

j. As M4 is still under EMCON and the Expiry_Time for the message is exceeded (Figure A-5), M0 will send the following PDUs:

Discard_Message_PDU (Source_ID = M0, MSID = 9876)

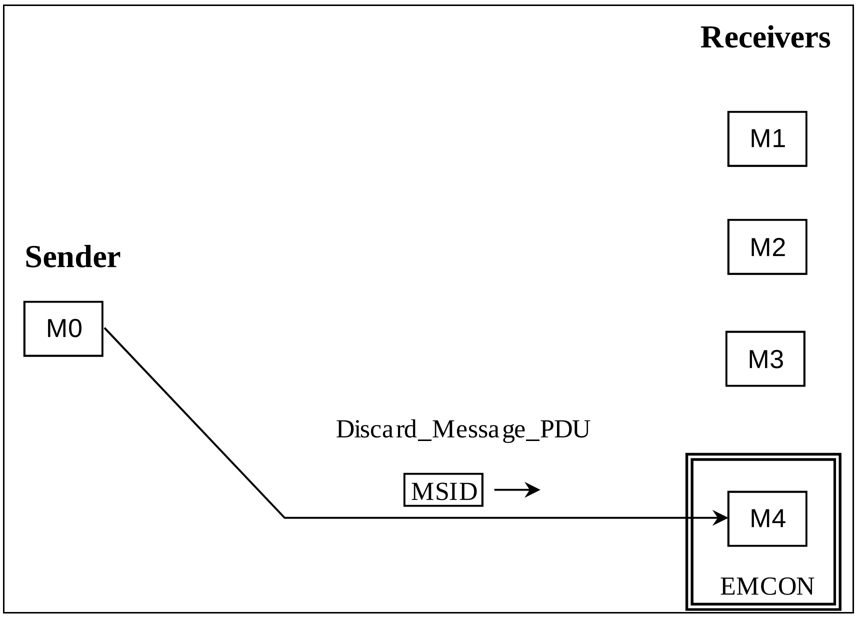


Figure 5-5: M0 forwards a Discard_Message_PDU to M4

k. After a further period of time, M4 leaves EMCON and sends an ACK_PDU to M0 (Figure A-6). As M4 received the total message before the Expiry_Time was reached, it will send the following ACK_PDU:

ACK_PDU (Source_ID_of_ACK_Sender = M4,
Count_of_ACK_Info_Entries = 1,
ACK_Info_Entry = (Length_of_ACK_Info_Entry = 10,
Source_ID = M0, MSID = 9876))

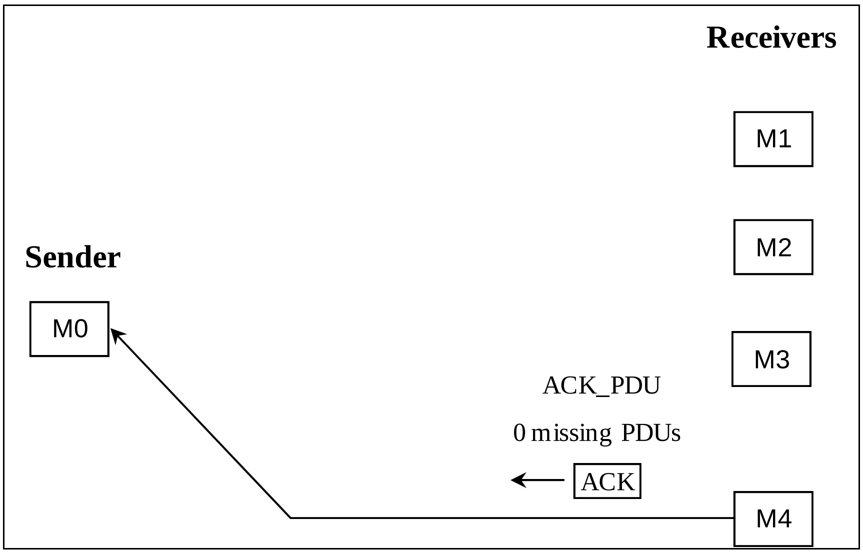


Figure 5-6: M4 leaves EMCON and acknowledges message receipt

l. At this time the Message Transmitter can hand over the message to Multicast_OUT to update the message queue entry. M0 will delete M4 from the list of Destination_Entries and will send out an Address_PDU with an empty list of Destination_Entries.

Address_PDU (source_ID =M0, MSID = 9876,
Expiry_Time = 1234567890, Total_Number_of_PDUs = 2,
Count_of_Destination_Entries = 0,
Length _of_Reserved_Field = 0, Destination_Entry = ( ))

to inform M1, M2, M3 and M4, that M0 received all ACK_PDUs from all receiving nodes (figure A-7). Consequently, all information about the message MSID = 9876 sent by M0 can be deleted.

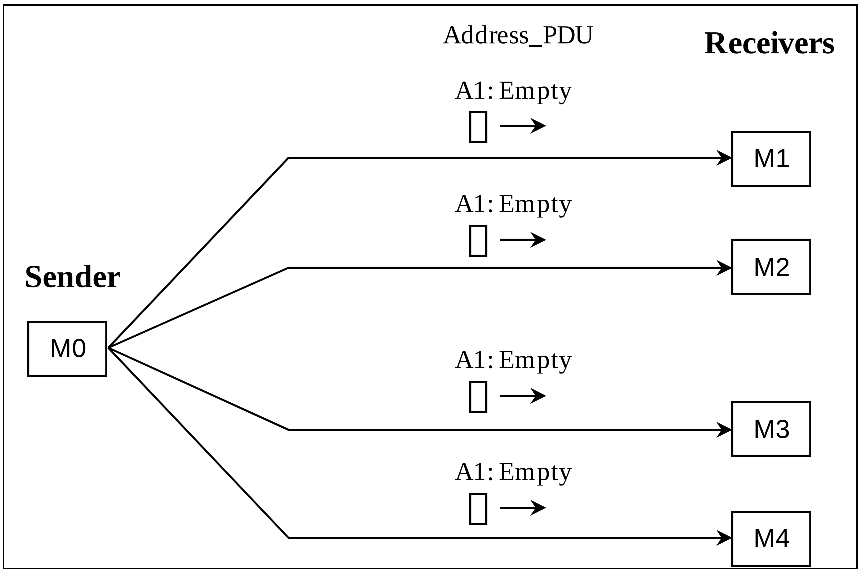


Figure 5-7: M0 indicates that all nodes have received the message

m. In case the message has been sent to a multicast group that had been dynamically agreed upon, the transmitting node M0 will release this multicast group by sending a Release_PDU to all members of the multicast group (figure A-8):

Release_PDU (source_ID = ID = M0, MSID = 9876,
Multicast_Group_Address = 240.1.2.4)

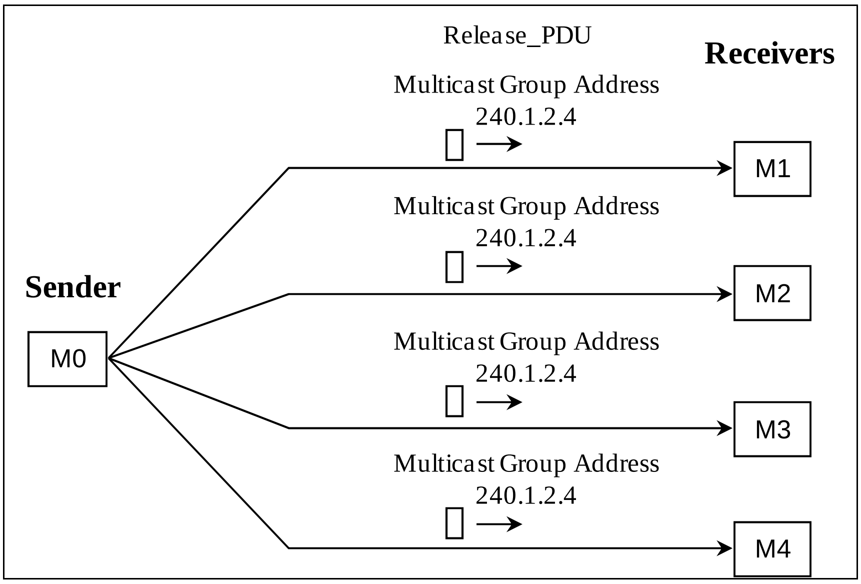


Figure 5-8: Dynamically created multicast group is released by Release_PDU

PARAMETERS AND ALGORITHM

PREDEFINED PROTOCOL PARAMETERS

WAIT_FOR_REJECT_TIME	Time between sending a Request_PDU and an affiliated Announce_PDU.
ANNOUNCE_DELAY	Time between sending an Announce_PDU and the first affiliated Address_PDU.
ANNOUNCE_CT	The number of times the Announce_PDU is transmitted,
RE-TRANSMISSION_TIME	Time a transmitter waits before re-transmitting a message to receivers not in EMCON, if no acknowledgement received. This value shall be dynamic and be calculated based on the size of the message, and on the time elapsed from the Address_PDU for a message is sent and to the ACK_PDU is received in order to reflect the change in transmission conditions. An initial value has to be set on best effort.
BACK-OFF_FACTOR	Multiplying factor applied to RE-TRANSMISSION_TIME on subsequent re-transmissions, to achieve exponentially increasing delay [On nth re-transmission, delay $_{n = A\_R\_T\_*(B\_F)}{}^{n}$]
EMCON_RTC	Re-transmission count – Maximum number of message re-transmission for receivers in EMCON.
EMCON_RTI	Re-transmission interval – Time in seconds a transmitter waits before re-transmitting a message for receivers in EMCON.
MM	Maximum number of new entries in the list of Missing_Data_PDU_Seq_Number field in an ACK_Info_Entry of an ACK_PDU.
ACK_PDU_TIME	time a receiver waits before re-transmitting ACK_PDU(s) if no response is received from the transmitting node.
MULTICAST_GROUP_TABLE	For each multicast group the table lists the IP address of each member and the multicast IP address of the group.
PDU_DELAY	Time between the transmission of two subsequent Data_PDUs. This value should be dynamically adjusted based on the measured throughput of previous messages. The receivers record the time of the reception of the Address_PDU and the time of transmission of the ACK_PDU. The difference in time is sent in the Tval

field of the ACK_PDU. Upon receipt the sender may use the Tval field to estimate the time it has taken to transmit the Data_PDUs and adjust the PDU_DELAY parameter accordingly. It should be noted that if concatenation is used by a radio at the link layer, the use of the Tval field may lead to wrong estimates. In this case using the measured time from transmission of the Address_PDU to the reception of the ACK_PDU may lead to better estimates for the PDU_DELAY. An initial value has to be set on best effort.

RECOMMENDED MULTICAST GROUP ADDRESS (FOR ORGANISATIONAL PURPOSES)

To allow the dynamic building of multicast groups it is necessary to have one multicast address to which all nodes have joined.

239.1.1.1 (GG) This group address is used for group management purposes. The transmitting node is using this group management Request_PDUs, Release_PDUs and Announce_PDUs. All sending and receiving nodes have to join this multicast group.

RECOMMENDED UDP PORT NUMBERS

To allow multiplexed multicast communication between all nodes of a network it is necessary to define some UDP port numbers:

PORT 2751 (TPORT) is used for the transmission of Request_PDUs, Reject_PDUs and Release_PDUs between the transmission programs. All transmitter processes have to listen to this port in conjunction with the multicast group GG.

PORT 2752 (RPORT) is used by the transmitters to send the Announce_PDUs, informing those receivers involved in the concerning message transfer to join a specific multicast group. All receiver processes have to listen to this port in conjunction with the multicast group GG.

PORT 2753 (DPORT) is used for the data traffic from the Message Transmitter of Multicast_OUT to the Message Receiver of Multicast_IN.

PORT 2754 (APORT) is used for the traffic from the Message Receiver of Multicast_IN to the message Transmitter of Multicast_OUT.

OPERATIONAL GUIDANCE

PURPOSE

This annex addresses operational considerations to how this protocol is to be used in the tactical messaging environment and the impact of parameter settings.

INTRODUCTION

P_MUL provides connection-oriented reliable multicast messaging for a pre-established or dynamically established multicast group. In addition, it provides an enhanced connectionless delivery mechanism, which transmits a message multiple times and accepts late acknowledgements when receivers leave EMCON. It can be optimised for low bandwidth and high delay environments but its use is not limited to those environments

CONSIDERATIONS:

a. There are two focus areas with respect to the operation of P_MUL.

(1) Multicast Addressing Scheme

(2) Configurable variables and parameter settings impact on P_MUL performance

b. These focus areas affect the performance and determine the behaviour of the protocol. All factors which influence the behaviour of the protocol will not be discussed nor will a recommendation be made as to how to configure this protocol. This section will simply discuss the protocol's behaviour as a result of how these factors affect it.

MULTICAST ADDRESSING SCHEME

a. The class D addressing scheme can be done according to:

(1) Predefined Grouping Destination Node

(2) Predefined Grouping by Network reachability

(3) Dynamic Group Allocation by Destination Node

(4) One multicast group for all potential recipients.

b. Regardless of the method chosen when implementing this protocol within an operational environment the range of multicast addresses used must be defined and managed. All P_MUL transmitters must know the range of multicast IP addressed.

PREDEFINED GROUPING BY DESTINATION NODE

If the multicast groups are defined according to destination Node, then class D addresses must be reserved to permit all of the possible combinations of the destination Nodes. The number of Class D addresses required grows exponentially to the number of Nodes as examples shown in Figure D-1. This growth is characterised by the equation Number of class D addresses required = n!/((k!)*(n-k)!)
n = Total number of Nodes in the group.
k = Number of Nodes within the group addressed in message

Groupings (k)	Number of Combinations		
	n = 6 ships or nodes in group	n = 10 ships or nodes in group	n = 15 ships or nodes in group
1	6	10	15
2	15	45	105
3	20	120	455
4	15	210	1365
5	6	252	3003
6	1	210	5005
7	-	120	6435
8	-	45	6435
9	-	10	5005
10	-	1	3003
11	-	-	1365
12	-	-	455
13	-	-	105
14	-	-	15
15	-	-	1
Total	63	1023	32767

Requires too many Class D addresses

Figure 5-9: Multicast addressing scheme according to destination Node

PREDEFINED GROUPINGS BY NETWORK REACHABILITY

a. If the multicast groups are defined according to network reachability, then consideration must be given to how best to group the class D. Embedded within this approach is the assumption that there is an awareness of the network topology and that any changes in the network topology is immediately reflected in the multicast group assignments. An example is plotted in Figure D-2.

b. The advantage of this approach is that each network would at worst case only receive 1 message per independent of the number of nodes on the message.

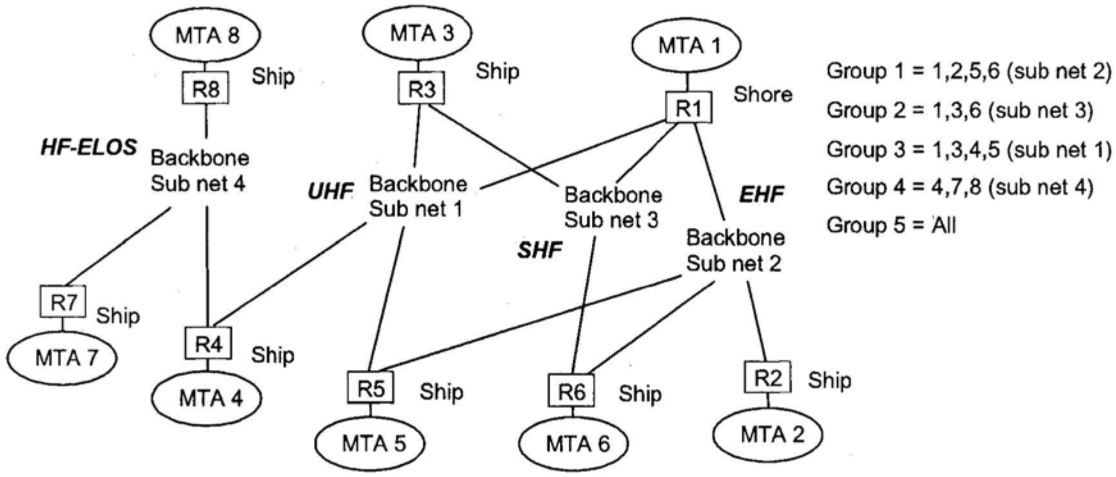


Figure 5-10: Multicast address grouping according to network reachability

DYNAMIC GROUP ALLOCATION BY DESTINATION NODE

With this approach a multicast group is only formed when required. This approach is similar to predefined grouping by destination Node except that the groups are not predetermined and no Class D addresses reserved for the possible combinations. Therefore there is no exponential growth in Class D addresses required.

ONE MULTICAST GROUP

This approach simply defines all of the potential nodes within one multicast group. The disadvantage of this approach is that it can be extremely inefficient if all of the potential recipients are not on a shared network. Unnecessary traffic would be introduced on networks which for a given message did not have recipient nodes in the message.

CONFIGURABLE VARIABLES AND THEIR IMPACT ON P_MUL

a. The assignment of values to the configurable parameters in P_MUL determines the behaviour and performance of the protocol. As the operators of this protocol set these values, the following information can be used as a guide to understand how the protocol will behave based on the decisions of the operator. This document will not discuss all of the configurable variables but a select few that have more significance.

b. **EMCON Re-Transmission counter (EMCON_RTC):** This parameter is used to determine the maximum number of times the transmitter will re-transmit the information to receiving nodes that are in EMCON. This variable can be used to increase the probability of reception for users in EMCON by setting a value other than 0. The operator should consider factors such as the average size of messages, bit error rate of the bearer, and whether or not there is a desire for P_MUL to manage EMCON re-transmissions. If there is a higher level mechanism managing EMCON re-transmissions, then the EMCON Re-transmission Counter value can be set to zero and bandwidth conserved during EMCON with the higher level mechanism administrating EMCON re-transmissions. Typical values range from 0-5.

c. **Acknowledgement PDU Timer (ACK_PDU_TIME):** This timer determines how long the receiver waits before sending acknowledgements to the transmitter when it has not received a response from the transmitter node. This timer can be used to prevent acknowledgement implosion. If this timer is too low then the receiver will tend to send unnecessary acknowledgement not giving the transmitter sufficient time to respond to the receiver's acknowledgements. When this variable is tuned to the network and the message load, it helps to minimise the number of unnecessary re-transmissions by the transmitter.

d. **Acknowledgement Re-transmission Timer (RE-TRANSMISSION_TIME):** This timer determines how long a transmitter waits for acknowledgements from the receivers not in EMCON before re-transmitting a message. This variable can be used to make the transmitter responsive to network delays and round trip times. If this value is too low with respect to network delays and round trip times then there will be unnecessary re-transmission potentially crippling the networks. the higher the value the fewer unnecessary re-transmissions.

e. **Relationship of Minimum Scan Time (MIN_SCAN_TIME) to PDU_DELAY.** **T**he minimum scan time determines how often the receiver scans its directory to see if PDUs have been received, sending acknowledgements when it believes the transmitter should have sent the next PDU. The PDU_DELAY determines how long the transmitter waits before sending successive PDUs. As the PDU_DELAY exceeds the MIN_SCAN_TIME unnecessary acknowledgements increase significantly.

CONCLUSION

P_MUL can be extremely efficient when careful consideration is given to the multicast addressing scheme used and the assignment values to configurable values. Because P_MUL provides for reliable multicast delivery of information, the transmitter and the receivers must be responsive to each other and their communication environment. Inefficiencies are realised when overcompensation and under-compensating occur. When P_MUL parameters are tuned properly to the network and to each other between transmitting and receiving nodes, the protocol is extremely efficient and quiet, in terms of network traffic.

GLOSSARY OF TERMS

ABBREVIATIONS AND DEFINITIONS

ABBREVIATIONS

ACP	Allied Communication Publication
AIS	Allied Information Services
CCEB	Combined Communications-Electronics Board
CSNI	Communications System Networks Interoperability
CWID	Combined Warrior International Demonstration
DVMRP	Distance Vector Multicast Routing Protocol
EMCON	Emission Control
GG	Global Group
IETF	Internet Engineering TF
IGMP	Internet Group Management Protocol
ISME	International Subject Matter Expert
IP	Internet Protocol
MADCAP	Multicast Address Dynamic Client Allocation Protocol
MAP	More Address_PDUs
MMHS	Military Message Handling System
MOSPF	Multicast Extension of OSPF
MTF	Messaging Task Force
OSPF	Open Shortest Path First an Interior Gateway Protocol
P_MUL	A protocol for reliable multicast messaging in bandwidth constrained and delayed acknowledgement (EMCON) environments
PDU	Protocol Data Unit
PIM	Protocol Independent Multicast
RFC	Request for Comment
TCTF	Tactical Communications Task Force
UDP	User Datagram Protocol
XTP	Xpress Transport Protocol

DEFINITIONS

Broadcast operation: The transmission of signals that may be simultaneously received by stations that usually make no acknowledgement.

Datagram: In packet switching, a self-contained packet, independent of other packets, that contains information sufficient for routing from the originating data terminal equipment (DTE) to the destination DTE without relying on prior exchanges between the equipment and the network. *Note:* Unlike virtual call service, when datagram's are sent there are no call establishment or clearing procedures. Thus, the network may not be able to provide protection against loss, duplication, or miss-delivery.

Emission control (EMCON): The selective and controlled use of electromagnetic, acoustic, or other emitters to optimise command and control capabilities while minimising, for operations security (OPSEC): (a) detection by enemy sensors; (b) to minimise mutual interference among friendly systems; and/or (c) to execute a military deception plan.

Full-duplex (FDX) circuit: A circuit that permits simultaneous transmission in both directions.

Half-duplex (HDX) operation: Operation in which communication between two terminals occurs in either direction, but in only one direction at a time. *Note:* Half-duplex operation may occur on a half-duplex circuit or on a duplex circuit, but it may not occur on a simplex circuit. *Synonyms* **one-way reversible operation, Two-way alternate operation**.

Multicast: 1. In a network, a technique that allows data, including packet form, to be simultaneously transmitted to a selected set of destinations. *Note:* Some networks, such as Ethernet, support multicast by allowing a network interface to belong to one or more multicast groups. **2.** To transmit identical data simultaneously to a selected set of destinations in a network, usually without obtaining acknowledgement of receipt of the transmission.

Multicast address: A routing address that (a) is used to address simultaneously all the computers in a group and (b) usually identifies a group of computers that share a common protocol, as opposed to a group of computers that share a common network. *Note:* Multicast address also applies to radio communications. *Synonym (in Internet protocol)* **class d address**.

Open Systems Interconnection (OSI)-Protocol Specification: The lowest level of abstraction within the OSI standards scheme. *Note:* Each OSI-Protocol Specification operates at a single layer. Each defines the primitive operations and permissible responses required to exchange information between peer processes in communicating systems to carry out all or a subset of the services defined within the OSI-Service Definitions for that layer.

Open Systems Interconnection–Reference Model (OSI–RM): An abstract description of the digital communications between application processes running in distinct systems. The model employs a hierarchical structure of seven layers. Each layer performs value-added service at the request of the adjacent higher layer and, in turn, requests more basic services from the adjacent lower layer.

- **Physical Layer:** Layer 1, the lowest of seven hierarchical layers. The Physical layer performs services requested by the Data Link Layer. The major functions and services performed by the physical layer are (a) establishment and termination of a connection to a communications medium; (b) participation in the process whereby the communication resources are effectively shared among multiple users, e.g., contention resolutions and flow control; and (c) conversion between the representation of digital data in user equipment and the corresponding signals transmitted over a communications channel.

- **Data Link Layer:** Layer 2. This layer responds to service requests from the Network Layer and issues service requests to the Physical Layer. The Data Link Layer provides the functional and procedural means to transfer data between network entitles and to detect and possibly correct errors that may occur in the Physical Layer. *Note:* Examples of data link protocols are HDLC and ADCCP for point-to-point or packet-switched networks and LLC for local area networks.

- **Network Layer:** Layer-3. This layer responds to service requests from the Transport Layer and issues service requests to the Data Link Layer. The Network Layer provides the functional and procedural means of transferring variable length data sequences from a source to a destination via one or more networks while maintaining the quality of service requested by the Transport Layer. The Network Layer performs routing, flow control, segmentation/desegmentation, and error control function.

- **Transport Layer:** Layer 4. This layer responds to service requests from the Session Layer and issues service requests to the Network Layer. The purpose of the Transport Layer is to provide transparent transfer of data between end users, thus relieving the upper layers from any concern with providing reliable and cost-effective data transfer.

- **Session Layer:** Layer 5. This layer responds to service requests from the Presentation Layer and issues service requests to the Transport Layer. the Session Layer provides the mechanism for managing the dialogue between end-user application processes. It provides for either duplex or half-duplex operation and establishes checkpointing, adjournment, termination, and restart procedures.

- **Presentation Layer:** Layer 6. This layer responds to service requests from the Application Layer and issues service requests to the Session Layer. The Presentation Layer relieves the Application Layer of concern regarding syntactical differences in data representation within the end-user systems. *Note:* An example of a presentation service would be the conversion of an EBCDIC-coded text file to an ASCII-coded file.

- **Application Layer:** Layer 7. The highest layer. This layer interfaces directly to and performs common application services for the application processes; it also issues requests to the Presentation Layer. The common application services provide semantic conversion between associated application processes. *Note:* Examples of common application services of general interest include the virtual file, virtual terminal and job transfer and manipulation protocols.

Protocol data unit (PDU): 1. Information that is delivered as a unit among peer entities of a network and that may contain control information, address information, or data. 2. In layered systems, a unit of data that is specified in a protocol of a given layer and that consists of protocol-control information of the given layer and possible user data of that layer.

Simple operation: 1. Operation in which transmission occurs in one and only one preassigned direction. *Synonym* **one-way operation**. *Note:* Duplex operation may be achieved by simplex operation of two or more simplex circuits or channels. **2.** Operating method in which transmission is made possible alternately in each direction of a telecommunication channel, for example by means of manual control. *Note:* In general, duplex operation and semi-duplex operation require two frequencies in radio communications; simplex operation may use either one or two. *Note 2:* These two definitions are contradictory; however, both are in common use. The first one is used in telephony and the second one is in radio. The user is cautioned to verify the nature of the service specified by this term.

Source quench: A congestion-control technique in which a computer experiencing data traffic congestion sends a message back to the source of the messages or packets causing the congestion, requesting that the source stop transmitting.

Transmit flow control: In data communications systems, control of the rate at which data are transmitted from a terminal so that data can be received by another terminal. *Note 1:* Transmit flow control may occur between data terminal equipment (DTE) and a switching centre, via data circuit-terminating equipment (DCE), or between two DTEs. The transmission rate may be controlled because of network or DTE requirements. *Note 2:* Transmit flow control can occur independently in the two directions of data transfer, thus permitting the transfer rates in one direction to be different from the transfer rates in the other direction.

UDP: *Abbreviation for* **user datagram protocol**. An Internet protocol for datagram service.

Unidirectional operation: Operation in which data are transmitted from a transmitter to a receiver in only one direction.