IEEE Standards

802.15.4™

IEEE Standard for Information technology— Telecommunications and information exchange between systems— Local and metropolitan area networks— Specific requirements

Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)

IEEE Computer Society

Sponsored by the
LAN/MAN Standards Committee

IEEE

Published by
The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

1 October 2003

Print: SH95127
PDF: SS95127

IEEE Standard for Information technology— Telecommunications and information exchange between systems— Local and metropolitan area networks Specific requirements—

Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)

Sponsor

LAN/MAN Standards Committee
of the
IEEE Computer Society

Approved 12 May 2003

IEEE-SA Standards Board

Abstract: This standard defines the protocol and compatible interconnection for data communication devices using low data rate, low power and low complexity, short-range radio frequency (RF) transmissions in a wireless personal area network (WPAN).

Keywords: ad hoc network, low data rate, low power, LR-WPAN, mobility, personal area network (PAN), radio frequency (RF), short range, wireless, wireless personal area network (WPAN)

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Print: ISBN 0-7381-3686-7 SH95127
PDF: ISBN 0-7381-3677-5 SS95127

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information contained in its standards.

Use of an IEEE Standard is wholly voluntary. The IEEE disclaims liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon this, or any other IEEE Standard document.

The IEEE does not warrant or represent the accuracy or content of the material contained herein, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained herein is free from patent infringement. IEEE Standards documents are supplied "**AS IS**."

The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation. When a document is more than five years old and has not been reaffirmed, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

In publishing and making this document available, the IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is the IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other IEEE Standards document, should rely upon the advice of a competent professional in determining the exercise of reasonable care in any given circumstances.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
USA

Note: Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Introduction

(This introduction is not part of IEEE Std 802.15.4-2003, IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 15.4: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANS).)

IEEE Std 802.15.4-2003

This standard defines the protocol and interconnection of devices via radio communication in a personal area network (PAN). The standard uses carrier sense multiple access with a collision avoidance medium access mechanism and supports star as well as peer-to-peer topologies. The media access is contention based; however, using the optional superframe structure, time slots can be allocated by the PAN coordinator to devices with time critical data. Connectivity to higher performance networks is provided through a PAN coordinator.

This standard specifies two PHYs: an 868/915 MHz direct sequence spread spectrum (DSSS) PHY and a 2450 MHz DSSS PHY. The 2450 MHz PHY supports an over-the-air data rate of 250 kb/s, and the 868/915 MHz PHY supports over-the-air data rates of 20 kb/s and 40 kb/s. The PHY chosen depends on local regulations and user preference.

This standard contains state-of-the-art material. The area covered by this standard is undergoing evolution. Revisions are anticipated to this standard within the next few years to clarify existing material, to correct possible errors, and to incorporate new related material. Details on the contents of this standard are provided on the following pages. Information on the current revision state of this and other IEEE 802® standards may be obtained from:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
P.O.Box 1331
Piscataway, NJ 08855-1331
USA

Participants

At the time the draft of this standard was sent to sponsor ballot, the IEEE P802.15™ Working Group had the following voting members:

Robert F. Heile, *Chair*
James D. Allen, *Vice Chair*
Patrick W. Kinney, *Secretary*
Michael D. McInnis, *Assistant Secretary and Editor*

Ian C. Gifford, *Task Group 1 Chair*
Stephen J. Shellhammer, *Task Group 2 Chair*
John R. Barr, *Task Group 3 Chair*

Patrick W. Kinney, *Task Group 4 Chair*
Phil Jamieson, *Task Group 4 Vice Chair*
José A. Gutierrez, *Task Group 4 Editor-in-Chief*
Marco Naeve, *Task Group 4 Secretary*

Monique Bourgeois, *MAC Technical Editor*
Said Moridi, *PHY Technical Editor*
Phil Jamieson, *Layer Management Technical Editor*

Greg Breen, *Low-Band PHY Technical Editing*
Ed Callaway, *Networking Technical Editing*
Paul Gorday, *High-Band PHY Technical Editing*
Marco Naeve, *General Description Technical Editing*
David Cypher, *PICs/SDLs Technical Editing*
Robert Poor, *Coexistence Technical Editing*
Farron Dacus, *Regulatory Technical Editing*

Roberto Aiello
Masaaki Akahane
Richard Alfvin
James D. Allen
Arun Arunachalam
Naiel Askar
Venkat I. Bahl
Daniel Bailey
Jay Bain
James Baker
Jaiganesh Balakrishnan
John R. Barr
Anuj Batra
Timothy Blaney
Kenneth Boehike
Stan Bottoms
Monique Bourgeois
Mark V. Bowles
Chuck Brabenac
Ed Callaway
Soo-Young Chang
Francois Po_Shin Chin
Aik Chindapol
Craig Conkling
David Cypher
Anand Dabak
Kai Dombrowski
Mary DuVal
Michael Dydyk
Jason L. Ellis
Mark W. Fidler
Jeff R. Foerster
David S. Furuno
Pierre Gandolfo
Atul Garg
Ian C.Gifford
James Gilb
Nada Golmie
Paul Gorday
José A. Gutierrez
Yasuo Harada
Allen Heberling
Robert F. Heile

Barry Herold
Robert Y. Huang
Eran Igler
Katsumi Ishii
Phil Jamieson
Jeyhan Karaoguz
Masami Katagiri
Joy H. Kelly
Stuart J. Kerry
Yongsuk Kim
Young Hwan Kim
Patrick W. Kinney
Günter Kleindl
Bruce P. Kraemer
DoHoon Kwon
Jim Lansford
David Leeper
Liang Li
Yeong-Chang Maa
Steven March
Ralph Mason
Michael D. McInnis
Jim Meyer
Leonard E. Miller
Akira Miura
Andreas Molisch
Antonio Mondragon
Tony Morelli
Said Moridi
Marco Naeve
Chiu Ngo
Kei Obara
Knut Odman
John B. Pardee
Jongun Park
Dave Patton
Marcus Pendergrass
Robert D. Poor
Gregg Rasor
Ivan Reede
Jim Richards
Glyn Roberts
Richard Roberts

William Roberts
Chris Rogers
Philippe Rouzet
Chandos Rypinski
John H. Santhoff
Mark Schrader
Tom Schuster
Erik Schylander
Michael Seals
Stephen J. Shellhammer
Nick Shepherd
Gadi Shor
William Shvodian
Thomas Siep
Kazimierz Siwiak
Carl Stevenson
Rene Struik
Shigeru Sugaya
Kazuhisa Takamura
Katsumi Takaoka
Teik-Kheong Tan
Larry Taylor
Stephen E. Taylor
Hans vanLeeuwen
Ritesh Vishwakarma
Thierry Walrant
Jing Wang
Fijio Watanabe
Mathew Welborn
Richard Wilson
Stephen Wood
Edward G. Woodrow
Hirohisa Yamaguchi
Amos Young
Song-Lin Young
Nakache Yves-paul
Jim Zyren

Major contributions were received from the following individuals:

Tony Adamson
David Archer
David Avery
Venkat Bahl
Daniel Bailey
Edul Batliwala
Pratik Bose
Boaz Carmeli
Farron Dacus
Martin Digon
Ian C. Gifford
Ed Hogervorst
Stephen Korfhage
Charles Luebke
Masahiro Maeda
Ian Marsden
Chris Marshall
Paul Marshall
Fred Martin
Ralph Mason
Rod Miller
Phil Rudland
Niels Schutten
Nick Shepherd
Ari Singer
Ralph D'Souza
Carl Stevenson
Mark Tilinghast
Hans Van Leeuwen
Jacco van Muiswinkel
Luis Pereira
Richard Wilson
Wim Zwart

The following members of the balloting committee voted on this standard:

Morris Balamut
John R. Barr
Shlomo Berliner
Pratik Bose
Monique Bourgeois
Daniel Brueske
Ed Callaway
Yawgeng Chau
Todor Cooklev
Guru Dutt Dhingra
Thomas Dineen
Dominic Espejo
Avraham Freedman
Ian C. Gifford
James Gilb
Paul Gorday
Rajugopal Gubbi
José A. Gutierrez
Simon Harrison
Robert F. Heile
Phil Jamieson
Tony Jeffree
Niket Jindal
James Kemerling
Stuart Kerry
Brian Kiernan
Yongsuk Kim
Patrick W. Kinney
Cees Klik
Gregory Luri
Roger Marks
Peter Martini
Ralph Mason
Lance McBride
Michael D. McInnis
George Miao
Hiroshi Miyano
Said Moridi
Marco Naeve
Paul Nikolich
Erwin Noble
Timothy O'Farrell
Bob O'Hara
Jack Pardee
Subbu Ponnuswamy
Robert Poor
Vikram Punj
Jon Rosdahl
Mark Schrader
Stephen J. Shellhammer
Jerry Thrasher
Johannes Van Leeuwen
Edward Woodrow
Jung Yee
Oren Yuen

When the IEEE-SA Standards Board approved this standard on 12 May 2003, it had the following membership:

Don Wright, *Chair*
Howard M. Frazier, *Vice Chair*
Judith Gorman, *Secretary*

H. Stephen Berger
Joe Bruder
Bob Davis
Richard DeBlasio
Julian Forster*
Toshio Fukuda
Arnold M. Greenspan
Raymond Hapeman
Donald M. Heirman
Laura Hitchcock
Richard H. Hulett
Anant Jain
Lowell G. Johnson
Joseph L. Koepfinger*
Tom McGean
Steve Mills
Daleep C. Mohla
William J. Moylan
Paul Nikolich
Gary Robinson
Malcolm V. Thaden
Geoffrey O. Thompson
Doug Topping
Howard L. Wolfman

*Member Emeritus

Also included are the following nonvoting IEEE-SA Standards Board liaisons:

Alan Cookson, *NIST Representative*
Satish K. Aggarwal, *NRC Representative*

Michelle Turner
IEEE Standards Project Editor

CONTENTS

IEEE Standard for
Information technology—
Telecommunications and information exchange between systems—
Local and metropolitan area networks—
Specific requirements

Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)

1. Overview

Wireless personal area networks (WPANs) are used to convey information over relatively short distances. Unlike wireless local area networks (WLANs), connections effected via WPANs involve little or no infrastructure. This feature allows small, power-efficient, inexpensive solutions to be implemented for a wide range of devices.

This document defines a standard for a low-rate WPAN (LR-WPAN).

1.1 Scope

The scope of this project is to define the physical layer (PHY) and medium access control (MAC) sublayer specifications for low data rate wireless connectivity with fixed, portable, and moving devices with no battery or very limited battery consumption requirements typically operating in the personal operating space (POS) of 10 m. It is foreseen that, depending on the application, a longer range at a lower data rate may be an acceptable trade-off.

It is the intent of this project to work toward a level of coexistence with other wireless devices in conjunction with Coexistence Task Groups, such as 802.15.2™ and 802.11™/ETSI-BRAN/MMAC 5GSG.

1.2 Purpose

The purpose of this document is to provide a standard for ultra-low complexity, ultra-low cost, ultra-low power consumption, and low data rate wireless connectivity among inexpensive devices. The raw data rate will be high enough (maximum of 250 kb/s) to satisfy a set of simple needs such as interactive toys, but scalable down to the needs of sensor and automation needs (20 kb/s or below) for wireless communications.

2. References

The following standards and specifications contain provisions which, through references in this text, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards and specifications are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the references listed below.

2.1 IEEE documents[1]

IEEE Std 802®-1990, IEEE Standards for Local and Metropolitan Area Networks: Overview and Architecture.

2.2 ISO documents[2]

ISO/IEC 7498-1:1994, Information technology—Open systems interconnection—Basic reference model: The basic model.

ISO/IEC 8802-2:1998 (IEEE Std 802.2™, 1998 Edition), Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 2: Logical link control.

ISO/IEC 9646-1:1994, Information technology—Open systems interconnection—Conformance testing methodology and framework— Part 1: General concepts.

ISO/IEC 9646-7:1995 (ITU-T Rec. X.296 (1994)), Information technology—Open systems interconnection—Conformance testing methodology and framework—Part 7: Implementation conformance statements.

ISO/IEC 10039:1991, Information technology—Open systems interconnection—Local area networks—Medium Access Control (MAC) service definition.

ISO/IEC 15802-1:1995, Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Common specifications—Part 1: Medium Access Control (MAC) service definition.

2.3 ITU-T documents[3]

ITU-T Recommendation X.210, Service Definitions—Open Systems Interconnection—Layer Service Definition Conventions.

ITU-T Recommendation Z.100, CCITT Specification and Description Language (SDL).

2.4 Other documents

NIST FIPS Pub 197: Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197, U.S. Department of Commerce/N.I.S.T., November 2001.[4]

[1] IEEE Publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA http://standards.ieee.org/catalog/.

[2] ISO and ISO/IEC publications are available from the ISO Central Secretariat, Case Postale 56, 1 rue de Varembé, CH-1211, Genève 20, Switzerland/Suisse http://www.iso.ch/. They are also available in the Unites States from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036, USA http://www.ansi.org.

[3] ITU-T publications are available from the International Telecommunications Union, Place des Nations, CH-1211, Geneva 20, Switzerland/Suisse (http://www.itu.int/).

[4] FIPS publications are available from the National Technical Information Service (NTIS), U. S. Dept. of Commerce, 5285 Port Royal Rd., Springfield, VA 22161 (http://www.ntis.org/).

3. Definitions

For the purposes of this standard, the following terms and definitions apply. Terms not defined in this clause can be found in the *The Authoritative Dictionary of IEEE Standards Terms*, Seventh Edition [B1].

3.1 access control list (ACL): A table used by a device to determine which devices are authorized to perform a specific function.

3.2 ad hoc network: An ad hoc network is typically created in a spontaneous manner. The principal characteristic of an ad hoc network is its limited temporal and spatial extent.

3.3 alternate personal area network (PAN) coordinator: A coordinator that is capable of replacing the personal area network (PAN) coordinator, should it leave the network for any reason. A PAN can have zero or more alternate PAN coordinators.

3.4 association: The service used to establish a device’s membership in a wireless personal area network (WPAN).

3.5 authentic data: Data whose source is verifiable through cryptographic protection.

3.6 authentication: The service used to establish the identity of one device as a member of the set of devices authorized to communicate securely to other devices in the set.

3.7 confidentiality: Assurance that communicated data remain private to the parties for whom the data are intended.

3.8 coordinator: An full-function device (FFD) that is configured to provide synchronization services through the transmission of beacons. If a coordinator is the principal controller of a personal area network (PAN), it is called the PAN coordinator.

3.9 coverage area: The area where two or more IEEE 802.15.4 ™ units can exchange messages with acceptable quality and performance.

3.10 data integrity: Assurance that the data have not been modified from their original form.

3.11 device: Any entity [reduced-function device (RFD) or full-function device (FFD)] containing an implementation of the IEEE 802.15.4 medium access control (MAC) and physical interface to the wireless medium.

3.12 disassociation: The service that removes an existing association.

3.13 frame: The format of aggregated bits from a medium access control (MAC) sublayer entity that are transmitted together in time.

3.14 full-function device (FFD): A device capable of operating as a coordinator or device and implementing the complete protocol set.

3.15 integrity code: A data string generated using a symmetric key that is typically appended to data in order to provide data integrity and source authentication (also called a *message integrity code*).

3.16 key establishment: A public-key process by which two entities securely establish a symmetric key that is known only by the participating entities.

3.17 key management: Methods for controlling keying material throughout the life cycle of the low-rate wireless personal area network (LR-WPAN) including creation, distribution, and destruction.

3.18 key transport: A process by which an entity sends a key to another entity.

3.19 logical channel: One of a variety of channels on a physical link.

3.20 message integrity code: *See:* **integrity code**.

3.21 mobile device: A device that uses network communications while in motion.

3.22 m-sequence: Maximal length linear feedback shift register sequence.

3.23 nonce: A time stamp, a counter, or a special marker intended to prevent unauthorized replay.

3.24 orphaned device: A device that has lost contact with its associated personal area network (PAN) coordinator.

3.25 personal area network (PAN) coordinator: A coordinator that is the principal controller of a personal area network (PAN). An IEEE 802.15.4 network has exactly one PAN coordinator.

3.26 payload data: The contents of a data message that is being transmitted.

3.27 payload protection: The generic term for providing security services on payload data, including confidentiality, data integrity, and authentication.

3.28 protocol data unit (PDU): The unit of data exchanged between two peer entities.

3.29 packet: The format of aggregated bits that are transmitted together in time across the physical medium.

3.30 personal operating space (POS): The space about a person or object that is typically about 10 m in all directions and envelops the person or object whether stationary or in motion.

3.31 portable device: A device that may be moved from location to location, but uses network communications only while at a fixed location.

3.32 pseudo-random number generation: The process of generating a deterministic sequence of bits from a given seed that has the statistical properties of a random sequence of bits when the seed is not known.

3.33 random number generator: A device that provides a sequence of bits that is unpredictable.

3.34 reduced-function device (RFD): A device operating with a minimal implementation of the IEEE 802.15.4 protocol.

3.35 security suite: A group of security operations designed to provide security services on medium access control (MAC) frames.

3.36 self-organizing: The ability of network nodes to detect the presence of other nodes and to organize into a structured, functioning network without human intervention.

3.37 self-healing: The ability of the network to detect, and recover from, faults appearing in either network nodes or communication links, without human intervention.

3.38 service data unit (SDU): Information that is delivered as a unit through a service access point (SAP).

3.39 symmetric key: A secret key that is shared between two or more parties that may be used for encryption/decryption or integrity protection/integrity verification depending on its intended use.

3.40 transaction: The exchange of related, consecutive frames between two peer medium access control (MAC) entities, required for a successful transmission of a MAC command or data frame**.**

3.41 wireless medium (WM): The medium used to implement the transfer of protocol data units (PDUs) between peer physical layer (PHY) entities of a low-rate wireless personal area network (LR-WPAN).

4. Acronyms and abbreviations

ACL	access control list
AES	advanced encryption standard
ASN.1	Abstract Syntax Notation Number 1
AWGN	additive white Gaussian noise
BE	backoff exponent
BER	bit error rate
BI	beacon interval
BO	beacon order
BPSK	binary phase-shift keying
BSN	beacon sequence number
CAP	contention access period
CBC-MAC	cipher block chaining message authentication code
CCA	clear channel assessment
CCM	CTR + CBC-MAC
CFP	contention-free period
CID	cluster identifier
CLH	cluster head
CRC	cyclic redundancy check
CSMA-CA	carrier sense multiple access with collision avoidance
CTR	counter mode
CW	contention window (length)
DSN	data sequence number
DSSS	direct sequence spread spectrum
ED	energy detection
EIRP	effective isotropic radiated power
EMC	electromagnetic compatibility
ERP	effective radiated power
EVM	error-vector magnitude
FCS	frame check sequence
FFD	full-function device
FH	frequency hopping
FHSS	frequency hopping spread spectrum
GTS	guaranteed time slot
IFS	interframe space or spacing
IR	infrared
ISM	industrial, scientific, and medical
IUT	implementation under test
LAN	local area network
LIFS	long interframe spacing
LLC	logical link control
LQ	link quality
LQI	link quality indication
LPDU	LLC protocol data unit
LR-WPAN	low-rate wireless personal area network

LSB	least significant bit
MAC	medium access control
MCPS	MAC common part sublayer
MCPS-SAP	MAC common part sublayer-service access point
MFR	MAC footer
MHR	MAC header
MIC	message integrity code
MLME	MAC sublayer management entity
MLME-SAP	MAC sublayer management entity-service access point
MSB	most significant bit
MSC	message sequence chart
MPDU	MAC protocol data unit
MSDU	MAC service data unit
NB	number of backoff (periods)
O-QPSK	offset quadrature phase-shift keying
OSI	open systems interconnection
PAN	personal area network
PANPC	personal area networkcomputer
PD-SAP	PHY data service access point
PDU	protocol data unit
PER	packet error rate
PHR	PHY header
PHY	physical layer
PIB	PAN information base
PICS	protocol implementation conformance statement
PLME	physical layer management entity
PLME-SAP	physical layer management entity-service access point
PN	pseudo-random noise
POS	personal operating space
PPDU	PHY protocol data unit
PRF	pulse repetition frequency
PSD	power spectral density
PSDU	PHY service data unit
ppm	parts per million
RF	radio frequency
RFD	reduced-function device
RSSI	received signal strength indication
RX	receive or receiver
SAP	service access point
SD	superframe duration
SDL	specification and description language
SPDU	SSCS protocol data units
SDU	service data unit
SFD	start-of-frame delimiter
SHR	synchronization header
SIFS	short interframe spacing

SO	superframe order
SRD	short-range device
SSCS	service specific convergence sublayer
SUT	system under test
TRX	transceiver
TX	transmit or transmitter
UML	unified modeling language
WLAN	wireless local area network
WPAN	wireless personal area network

5. General description

A LR-WPAN is a simple, low-cost communication network that allows wireless connectivity in applications with limited power and relaxed throughput requirements. The main objectives of an LR-WPAN are ease of installation, reliable data transfer, short-range operation, extremely low cost, and a reasonable battery life, while maintaining a simple and flexible protocol.

Some of the characteristics of an LR-WPAN are

— Over-the-air data rates of 250 kb/s, 40 kb/s, and 20 kb/s
— Star or peer-to-peer operation
— Allocated 16 bit short or 64 bit extended addresses
— Allocation of guaranteed time slots (GTSs)
— Carrier sense multiple access with collision avoidance (CSMA-CA) channel access
— Fully acknowledged protocol for transfer reliability
— Low power consumption
— Energy detection (ED)
— Link quality indication (LQI)
— 16 channels in the 2450 MHz band, 10 channels in the 915 MHz band, and 1 channel in the 868 MHz band

Two different device types can participate in an LR-WPAN network; a full-function device (FFD) and a reduced-function device (RFD). The FFD can operate in three modes serving as a personal area network (PAN) coordinator, a coordinator, or a device. An FFD can talk to RFDs or other FFDs, while an RFD can talk only to an FFD. An RFD is intended for applications that are extremely simple, such as a light switch or a passive infrared sensor; they do not have the need to send large amounts of data and may only associate with a single FFD at a time. Consequently, the RFD can be implemented using minimal resources and memory capacity.

5.1 Components of the IEEE 802.15.4 WPAN

A system conforming to IEEE 802.15.4 consists of several components. The most basic is the device. A device can be an RFD or an FFD. Two or more devices within a POS communicating on the same physical channel constitute a WPAN. However, a network shall include at least one FFD, operating as the PAN coordinator.

An IEEE 802.15.4 network is part of the WPAN family of standards although the coverage of an LR-WPAN may extend beyond the POS, which typically defines the WPAN.

A well-defined coverage area does not exist for wireless media because propagation characteristics are dynamic and uncertain. Small changes in position or direction may result in drastic differences in the signal strength or quality of the communication link. These effects occur whether a device is stationary or mobile as moving objects may impact station-to-station propagation.

5.2 Network topologies

Depending on the application requirements, the LR-WPAN may operate in either of two topologies: the star topology or the peer-to-peer topology. Both are shown in Figure 1. In the star topology the communication is established between devices and a single central controller, called the PAN coordinator. A device typically has some associated application and is either the initiation point or the termination point for network communications. A PAN coordinator may also have a specific application, but it can be used to initiate, terminate, or route communication around the network. The PAN coordinator is the primary controller of the

PAN. All devices operating on a network of either topology shall have unique 64 bit extended addresses. This address can be used for direct communication within the PAN, or it can be exchanged for a short address allocated by the PAN coordinator when the device associates. The PAN coordinator may be mains powered, while the devices will most likely be battery powered. Applications that benefit from a star topology include home automation, personal computer (PC) peripherals, toys and games, and personal health care.

The peer-to-peer topology also has a PAN coordinator; however, it differs from the star topology in that any device can communicate with any other device as long as they are in range of one another. Peer-to-peer topology allows more complex network formations to be implemented, such as mesh networking topology. Applications such as industrial control and monitoring, wireless sensor networks, asset and inventory tracking, intelligent agriculture, and security would benefit from such a network topology. A peer-to-peer network can be ad hoc, self-organizing and self-healing. It may also allow multiple hops to route messages from any device to any other device on the network. Such functions can be added at the network layer, but are not part of this standard.

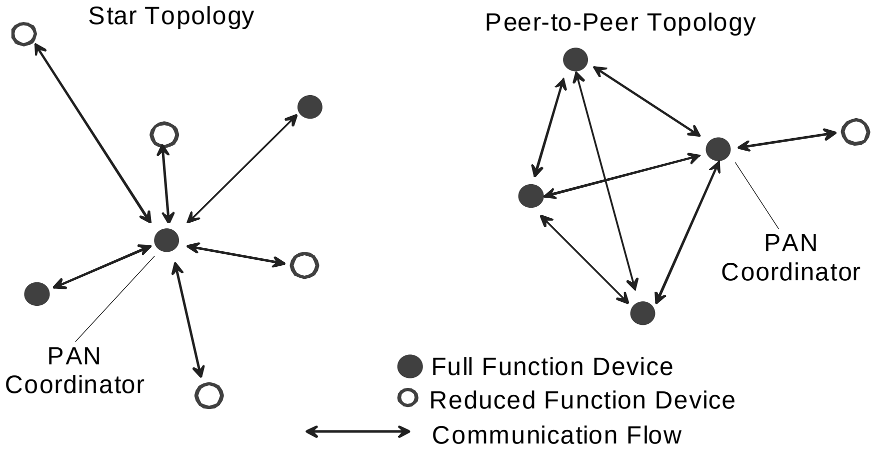


Figure 1—Star and peer-to-peer topology examples

Each independent PAN will select a unique identifier. This PAN identifier allows communication between devices within a network using short addresses and enables transmissions between devices across independent networks.

5.2.1 Network formation

The network formation is performed by the network layer, which is not part of this standard. However, this subclause provides a brief overview on how each supported topology may be formed.

5.2.1.1 Star network formation

The basic structure of a star network can be seen in Figure 1. After an FFD is activated for the first time, it may establish its own network and become the PAN coordinator. All star networks operate independently from all other star networks currently in operation. This is achieved by choosing a PAN identifier, which is not currently used by any other network within the radio sphere of influence. Once the PAN identifier is chosen, the PAN coordinator can allow other devices to join its network; both FFDs and RFDs may join the network. The detailed procedure can be found in 7.5.2 and 7.5.3.

5.2.1.2 Peer-to-peer network formation

In a peer-to-peer topology, each device is capable of communicating with any other device within its radio sphere of influence. One device will be nominated as the PAN coordinator, for instance, by virtue of being the first device to communicate on the channel. Further network structures can be constructed out of the peer-to-peer topology and may impose topological restrictions on the formation of the network.

An example of the use of the peer-to-peer communications topology is the cluster-tree. The cluster-tree network is a special case of a peer-to-peer network in which most devices are FFDs. An RFD may connect to a cluster tree network as a leave node at the end of a branch, because it may only associate with one FFD at a time. Any of the FFDs may act as a coordinator and provide synchronization services to other devices or other coordinators. Only one of these coordinators can be the overall PAN coordinator, which may have greater computational resources than any other device in the PAN. The PAN coordinator forms the first cluster by establishing itself as the cluster head (CLH) with a cluster identifier (CID) of zero, choosing an unused PAN identifier, and broadcasting beacon frames to neighboring devices. A candidate device receiving a beacon frame may request to join the network at the CLH. If the PAN coordinator permits the device to join, it will add the new device as a child device in its neighbor list. Then the newly joined device will add the CLH as its parent in its neighbor list and begin transmitting periodic beacons; other candidate devices may then join the network at that device. If the original candidate device is not able to join the network at the CLH, it will search for another parent device. The detailed procedures describing how a PAN is started and how devices join a PAN can be found in 7.5.2 and 7.5.3. The simplest form of a cluster tree network is a single cluster network, but larger networks are possible by forming a mesh of multiple neighboring clusters. Once predetermined application or network requirements are met, the PAN coordinator may instruct a device to become the CLH of a new cluster adjacent to the first one. Other devices gradually connect and form a multicluster network structure, such as the one seen in Figure 2. The lines in Figure 2 represent the parent-child relationships of the devices and not the communication flow. The advantage of a multicluster structure is increased coverage area, while the disadvantage is an increase in message latency.

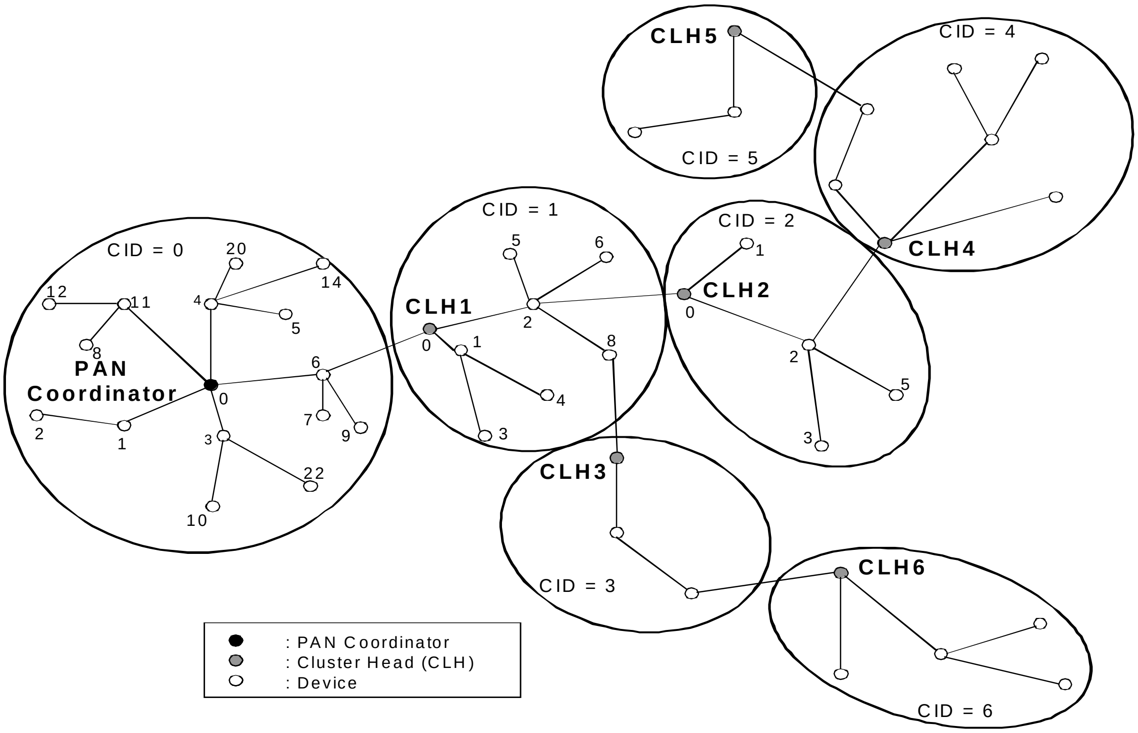


Figure 2—Cluster tree network

5.3 Architecture

The LR-WPAN architecture is defined in terms of a number of blocks in order to simplify the standard. These blocks are called layers. Each layer is responsible for one part of the standard and offers services to the higher layers. The layout of the blocks is based on the open systems interconnection (OSI) seven-layer model (see 2.2).

The interfaces between the layers serve to define the logical links that are described in this standard.

An LR-WPAN device comprises a PHY, which contains the radio frequency (RF) transceiver along with its low-level control mechanism, and a MAC sublayer that provides access to the physical channel for all types of transfer. Figure 3 shows these blocks in a graphical representation, which are described in more detail in 5.3.1 and 5.3.2.

The upper layers, shown in Figure 3, consist of a network layer, which provides network configuration, manipulation, and message routing, and an application layer, which provides the intended function of the device. The definition of these upper layers is outside the scope of this standard. An IEEE 802.2™ Type 1 logical link control (LLC) (see 2.1) can access the MAC sublayer through the service specific convergence sublayer (SSCS), defined in Annex A. The LR-WPAN architecture can be implemented either as embedded devices or as devices requiring the support of an external device such as a PC.

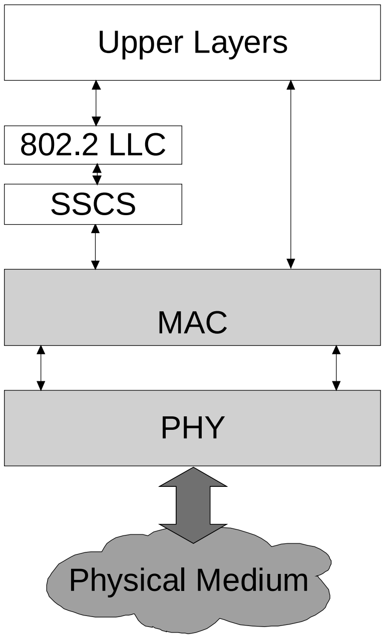


Figure 3—LR-WPAN device architecture

5.3.1 PHY

The PHY provides two services: the PHY data service and the PHY management service interfacing to the physical layer management entity (PLME). The PHY data service enables the transmission and reception of PHY protocol data units (PPDUs) across the physical radio channel.

Clause 6 contains the specifications for the PHY.

The features of the PHY are activation and deactivation of the radio transceiver, ED, LQI, channel selection, clear channel assessment (CCA), and transmitting as well as receiving packets across the physical medium. The radio shall operate at one of the following license-free bands:

— 868–868.6 MHz (e.g., Europe),
— 902–928 MHz (e.g., North America) or
— 2400–2483.5 MHz (worldwide).

Refer to Annex F for an informative summary of regulatory requirements.

5.3.2 MAC sublayer

The MAC sublayer provides two services: the MAC data service and the MAC management service interfacing to the MAC sublayer management entity (MLME) service access point (SAP) (known as MLME-SAP). The MAC data service enables the transmission and reception of MAC protocol data units (MPDUs) across the PHY data service.

The features of the MAC sublayer are beacon management, channel access, GTS management, frame validation, acknowledged frame delivery, association, and disassociation. In addition, the MAC sublayer provides hooks for implementing application appropriate security mechanisms.

Clause 7 contains the specifications for the MAC sublayer.

5.4 Functional overview

A brief overview of the general functions of a LR-WPAN is given in 5.4.1 through 5.4.6 and includes information on the superfame structure, the data transfer model, the frame structure, robustness, power consumption considerations, and security.

5.4.1 Superframe structure

The LR-WPAN standard allows the optional use of a superframe structure. The format of the superframe is defined by the coordinator. The superframe is bounded by network beacons, is sent by the coordinator (see Figure 4), and is divided into 16 equally sized slots. The beacon frame is transmitted in the first slot of each superframe. If a coordinator does not wish to use a superframe structure, it may turn off the beacon transmissions. The beacons are used to synchronize the attached devices, to identify the PAN, and to describe the structure of the superframes. Any device wishing to communicate during the contention access period (CAP) between two beacons shall compete with other devices using a slotted CSMA-CA mechanism. All transactions shall be completed by the time of the next network beacon.

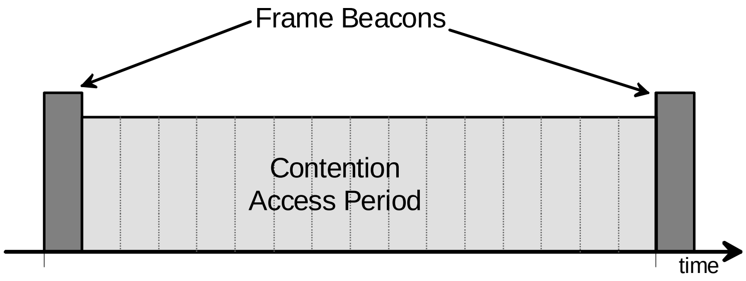


Figure 4—Superframe structure without GTSs

The superframe can have an active and an inactive portion. During the inactive portion, the coordinator shall not interact with its PAN and may enter a low-power mode.

For low-latency applications or applications requiring specific data bandwidth, the PAN coordinator may dedicate portions of the active superframe to that application. These portions are called guaranteed time slots (GTSs). The GTSs form the contention-free period (CFP), which always appears at the end of the active superframe starting at a slot boundary immediately following the CAP, as shown in Figure 5. The PAN coordinator may allocate up to seven of these GTSs, and a GTS may occupy more than one slot period. However, a sufficient portion of the CAP shall remain for contention-based access of other networked devices or new devices wishing to join the network. All contention-based transactions shall be complete before the CFP begins. Also each device transmitting in a GTS shall ensure that its transaction is complete before the time of the next GTS or the end of the CFP. More information on the superframe structure can be found in 7.5.1.1.

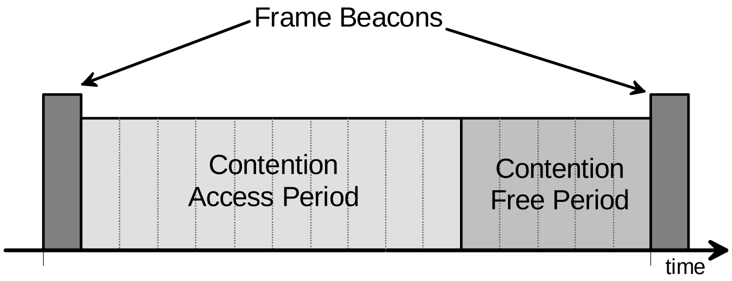


Figure 5—Superframe structure with GTSs

5.4.2 Data transfer model

Three types of data transfer transactions exist. The first one is the data transfer to a coordinator in which a device transmits the data. The second transaction is the data transfer from a coordinator in which the device receives the data. The third transaction is the data transfer between two peer devices. In star topology only two of these transactions are used, because data may be exchanged only between the coordinator and a device. In a peer-to-peer topology data may be exchanged between any two devices on the network; consequently all three transactions may be used in this topology.

The mechanisms for each transfer type depend on whether the network supports the transmission of beacons. A beacon-enabled network is used for supporting low-latency devices, such as PC peripherals. If the network does not need to support such devices, it can elect not to use the beacon for normal transfers. However, the beacon is still required for network association. The structure of the frames used for the data transfer is described in 5.4.3.

5.4.2.1 Data transfer to a coordinator

This data transfer transaction is the mechanism to transfer data from a device to a coordinator.

When a device wishes to transfer data to a coordinator in a beacon-enabled network, it first listens for the network beacon. When the beacon is found, the device synchronizes to the superframe structure. At the appropriate point, the device transmits its data frame, using slotted CSMA-CA, to the coordinator. The coordinator acknowledges the successful reception of the data by transmitting an optional acknowledgment frame. The transaction is now complete. This sequence is summarized in Figure 6.

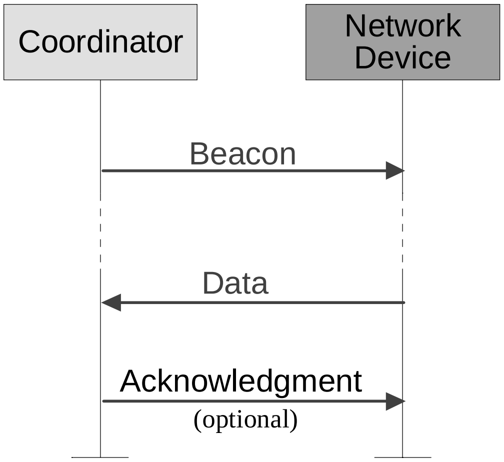


Figure 6—Communication to a coordinator in a beacon-enabled network

When a device wishes to transfer data in a nonbeacon-enabled network, it simply transmits its data frame, using unslotted CSMA-CA, to the coordinator. The coordinator acknowledges the successful reception of the data by transmitting an optional acknowledgment frame. The transaction is now complete. This sequence is summarized in Figure 7.

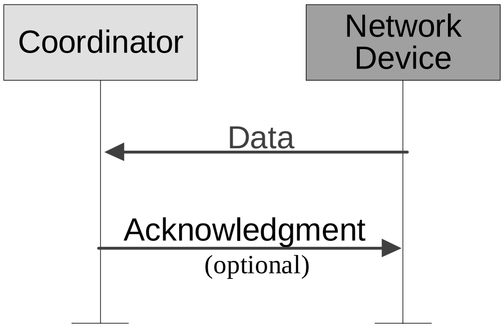


Figure 7—Communication to a coordinator in a nonbeacon-enabled network

5.4.2.2 Data transfer from a coordinator

This data transfer transaction is the mechanism for transferring data from a coordinator to a device.

When the coordinator wishes to transfer data to a device in a beacon-enabled network, it indicates in the network beacon that the data message is pending. The device periodically listens to the network beacon and, if a message is pending, transmits a MAC command requesting the data, using slotted CSMA-CA. The coordinator acknowledges the successful reception of the data request by transmitting an optional acknowledgment frame. The pending data frame is then sent using slotted CSMA-CA. The device acknowledges the successful reception of the data by transmitting an acknowledgment frame. The transaction is now complete. Upon receiving the acknowledgement, the message is removed from the list of pending messages in the beacon. This sequence is summarized in Figure 8.

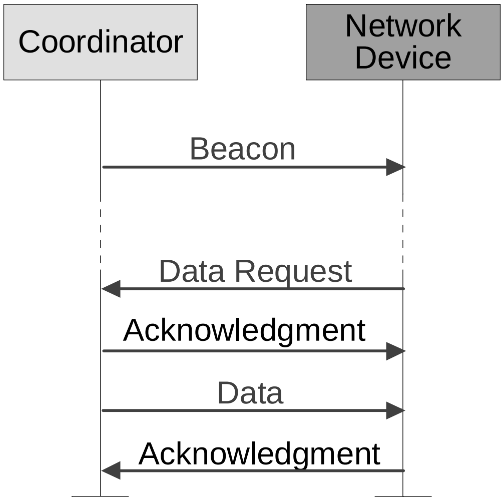


Figure 8—Communication from a coordinator a beacon-enabled network

When a coordinator wishes to transfer data to a device in a nonbeacon-enabled network, it stores the data for the appropriate device to make contact and request the data. A device may make contact by transmitting a MAC command requesting the data, using unslotted CSMA-CA, to its coordinator at an application-defined rate. The coordinator acknowledges the successful reception of the data request by transmitting an acknowledgment frame. If data are pending, the coordinator transmits the data frame, using unslotted CSMA-CA, to the device. If data are not pending, the coordinator transmits a data frame with a zero-length payload to indicate that no data were pending. The device acknowledges the successful reception of the data by transmitting an acknowledgment frame. The transaction is complete. This sequence is summarized in Figure 9.

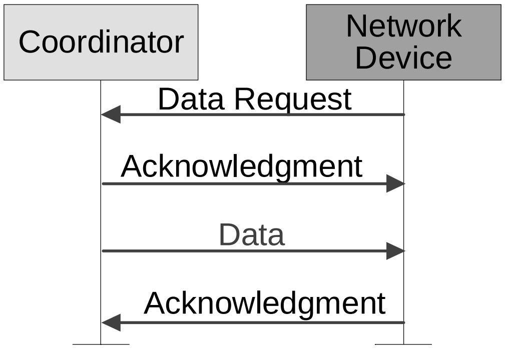


Figure 9—Communication from a coordinator in a nonbeacon-enabled network

5.4.2.3 Peer-to-peer data transfers

In a peer-to-peer PAN, every device may communicate with every other device in its radio sphere of influence. In order to do this effectively, the devices wishing to communicate will need to either receive constantly or synchronize with each other. In the former case, the device can simply transmit its data using unslotted CSMA-CA. In the latter case, other measures need to be taken in order to achieve synchronization. Such measures are beyond the scope of this standard.

5.4.3 Frame structure

The frame structures have been designed to keep the complexity to a minimum while at the same time making them sufficiently robust for transmission on a noisy channel. Each successive protocol layer adds to the structure with layer-specific headers and footers. The LR-WPAN defines four frame structures:

— A beacon frame, used by a coordinator to transmit beacons
— A data frame, used for all transfers of data
— An acknowledgment frame, used for confirming successful frame reception
— A MAC command frame, used for handling all MAC peer entity control transfers

The structure of each of the four frame types is described in 5.4.3.1 through 5.4.3.4. The diagrams in these subclauses illustrate the fields that are added by each layer of the protocol. The packet structure illustrated below the PHY represents the bits that are actually transmitted on the physical medium.

5.4.3.1 Beacon frame

Figure 10 shows the structure of the beacon frame, which originates from the MAC sublayer. A coordinator can transmit network beacons in a beacon-enabled network. The MAC service data unit (MSDU) contains the superframe specification, pending address specification, address list, and beacon payload fields (see 7.2.2.1). The MSDU is prefixed with a MAC header (MHR) and appended with a MAC footer (MFR). The MHR contains the MAC frame control fields, beacon sequence number (BSN), and addressing information fields. The MFR contains a 16 bit frame check sequence (FCS). The MHR, MSDU, and MFR together form the MAC beacon frame (i.e., MPDU).

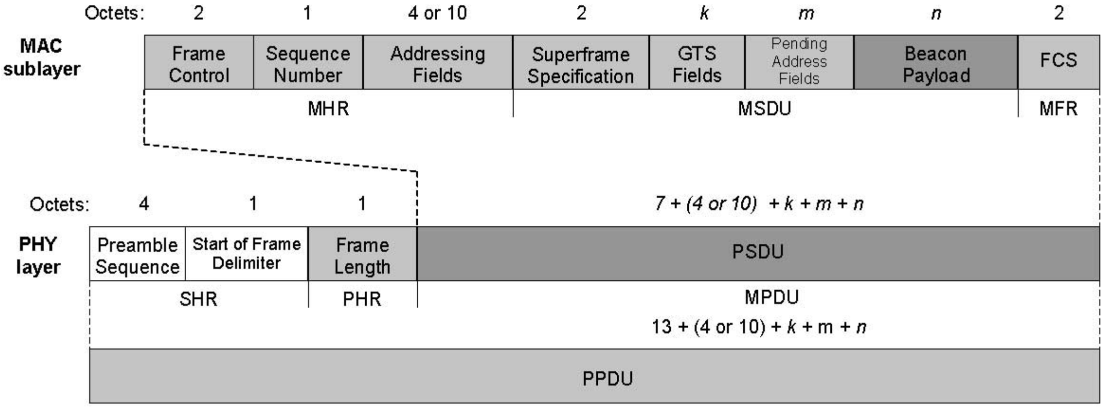


Figure 10—Schematic view of the beacon frame

The MPDU is then passed to the PHY as the PHY beacon packet payload (PHY service data unit, PSDU). The PSDU is prefixed with a synchronization header (SHR), containing the preamble sequence and start-of-frame delimiter (SFD) fields, and a PHY header (PHR) containing the length of the PSDU in octets. The preamble sequence enables the receiver to achieve symbol synchronization. The SHR, PHR, and PSDU together form the PHY beacon packet, (i.e., PPDU).

5.4.3.2 Data frame

Figure 11 shows the structure of the data frame, which originates from the upper layers.

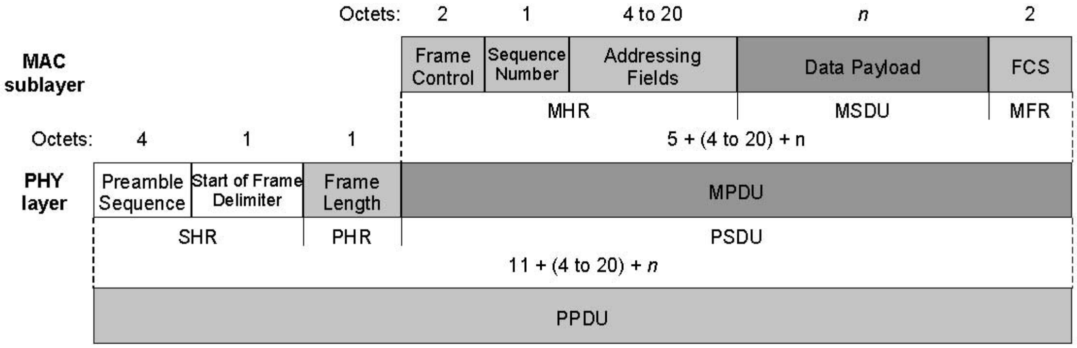


Figure 11—Schematic view of the data frame

The data payload is passed to the MAC sublayer and is referred to as the MSDU. The MSDU is prefixed with an MHR and appended with an MFR. The MHR contains the frame control, sequence number, and addressing information fields. The MFR is composed of a 16 bit FCS. The MHR, MSDU, and MFR together form the MAC data frame, (i.e., MPDU).

The MPDU is passed to the PHY as the PHY data frame payload, (i.e., PSDU). The PSDU is prefixed with an SHR, containing the preamble sequence and SFD fields, and a PHR containing the length of the PSDU in octets. The preamble sequence and the data SFD enable the receiver to achieve symbol synchronization. The SHR, PHR, and PSDU together form the PHY data packet, (i.e., PPDU).

5.4.3.3 Acknowledgment frame

Figure 12 shows the structure of the acknowledgment frame, which originates from the MAC sublayer. The MAC acknowledgment frame is constructed from an MHR and an MFR. The MHR contains the MAC frame control and data sequence number fields. The MFR is composed of a 16 bit FCS. The MHR and MFR together form the MAC acknowledgment frame (i.e., MPDU).

The MPDU is passed to the PHY as the PHY acknowledgment frame payload, (i.e., PSDU). The PSDU is prefixed with the SHR, containing the preamble sequence and SFD fields, and the PHR containing the length of the PSDU in octets. The SHR, PHR, and PSDU together form the PHY acknowledgment packet, (i.e., PPDU).

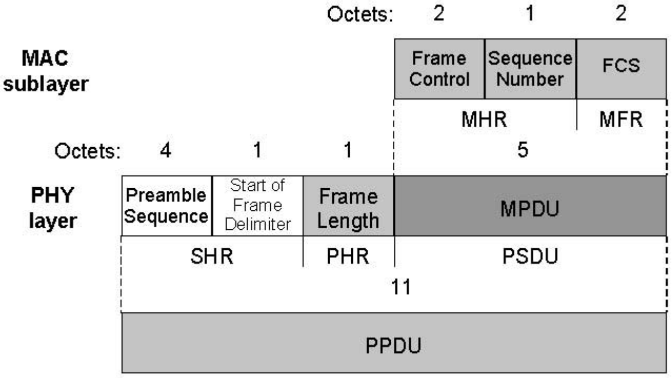


Figure 12—Schematic view of the acknowledgment frame

5.4.3.4 MAC command frame

Figure 13 shows the structure of the MAC command frame, which originates from the MAC sublayer. The MSDU contains the command type field and command specific data, called the command payload (see 7.2.2.4). The MSDU is prefixed with an MHR and appended with an MFR. The MHR contains the MAC frame control, data sequence number, and addressing information fields. The MFR contains a 16 bit FCS. The MHR, MSDU, and MFR together form the MAC command frame, (i.e., MPDU).

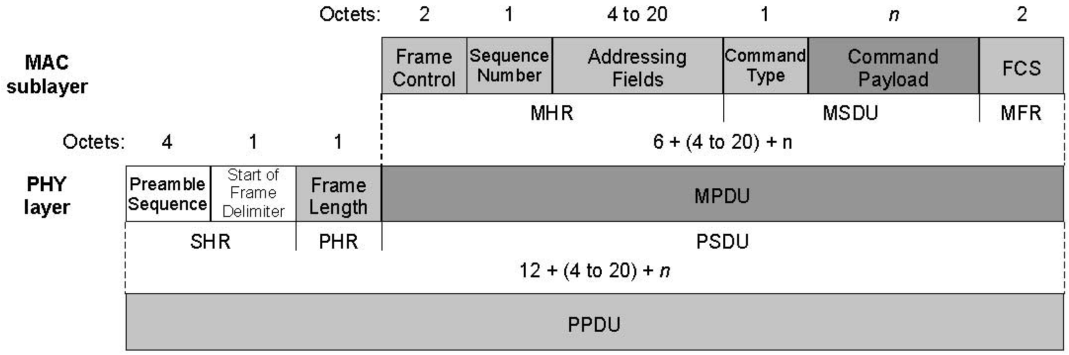


Figure 13—Schematic view of the MAC command frame

The MPDU is then passed to the PHY as the PHY command frame payload, (i.e., PSDU). The PSDU is prefixed with an SHR, containing the preamble sequence and SFD fields, and a PHR containing the length of the PSDU in octets. The preamble sequence enables the receiver to achieve symbol synchronization. The SHR, PHR, and PSDU together form the PHY command packet, (i.e., PPDU).

5.4.4 Robustness

The LR-WPAN employs various mechanisms to ensure robustness in the data transmission. These mechanisms are the CSMA-CA mechanism, frame acknowledgment, and data verification and are briefly discussed in 5.4.4.1 through 5.4.4.3.

5.4.4.1 CSMA-CA mechanism

The LR-WPAN uses two types of channel access mechanism, depending on the network configuration. Nonbeacon-enabled networks use an unslotted CSMA-CA channel access mechanism. Each time a device wishes to transmit data frames or MAC commands, it shall wait for a random period. If the channel is found to be idle, following the random backoff, the device shall transmit its data. If the channel is found to be busy, following the random backoff, the device shall wait for another random period before trying to access the channel again. Acknowledgment frames shall be sent without using a CSMA-CA mechanism.

Beacon-enabled networks use a slotted CSMA-CA channel access mechanism, where the backoff slots are aligned with the start of the beacon transmission. Each time a device wishes to transmit data frames during the CAP, it shall locate the boundary of the next backoff slot and then wait for a random number of backoff slots. If the channel is busy, following this random backoff, the device shall wait for another random number of backoff slots before trying to access the channel again. If the channel is idle, the device can begin transmitting on the next available backoff slot boundary. Acknowledgment and beacon frames shall be sent without using a CSMA-CA mechanism.

The CSMA-CA mechanism is discussed in 7.5.1.

5.4.4.2 Frame acknowledgment

A successful reception and validation of a data or MAC command frame can be optionally confirmed with an acknowledgment. If the receiving device is unable to handle the received data frame for any reason, the message is not acknowledged.

If the originator does not receive an acknowledgment after some period, it assumes that the transmission was unsuccessful and retries the frame transmission. If an acknowledgment is still not received after several retries, the originator can choose either to terminate the transaction or to try again. When the acknowledgment is not required, the originator assumes the transmission was successful.

The use of acknowledgment is discussed in detail in 7.5.6.4.

5.4.4.3 Data verification

In order to detect bit errors, an FCS mechanism, employing a 16 bit International Telecommunication Union—Telecommunication Standardization Sector (ITU-T) cyclic redundancy check (CRC), is used to protect every frame.

The FCS mechanism is discussed in 7.2.1.8.

5.4.5 Power consumption considerations

In many applications that use this standard, the devices will be battery powered where their replacement or recharging in relatively short intervals is impractical; therefore the power consumption is of significant concern. This standard was developed with the limited power supply availability in mind. However, the physical implementation of this standard will require additional power management considerations that are beyond the scope of this standard.

The protocol has been developed to favor battery-powered devices. However, in certain applications some of these devices could potentially be mains powered. Battery-powered devices will require duty-cycling to reduce power consumption. These devices will spend most of their operational life in a sleep state; however, each device shall periodically listen to the RF channel in order to determine whether a message is pending. This mechanism allows the application designer to decide on the balance between battery consumption and message latency. Mains-powered devices have the option of listening to the RF channel continuously.

5.4.6 Security

Although the diverse range of applications to which this standard is targeted imposes significant constraints on requiring a baseline security implementation in the MAC sublayer, some required security functionality is needed in order to provide basic security services and interoperability among all devices implementing this standard. This baseline includes the ability to maintain an access control list (ACL) and use symmetric cryptography to protect transmitted frames. The ability to perform this security functionality does not imply, however, that security shall be used at any given time by any given device. The higher layers determine when security is to be used at the MAC sublayer and provide all keying material necessary to provide the security services. Key management, device authentication, and freshness protection may be provided by the higher layers, but are out of scope of this standard. A brief introduction of some security terms is provided in 5.4.6.1 and 5.4.6.2; for more detailed information refer to Clause 7.

5.4.6.1 Security services

The security mechanisms in this standard are symmetric-key based using keys provided by higher layer processes. The management and establishment of these keys is the responsibility of the implementer. The security provided by these mechanisms assume the keys are generated, transmitted, and stored in a secure manner.

5.4.6.1.1 Access control

Access control is a security service that provides the ability for a device to select the other devices with which it is willing to communicate. In this standard, if the access control service is provided, a device shall maintain a list of devices in its ACL from which it expects to receive frames.

5.4.6.1.2 Data encryption

In this standard data encryption is a security service that uses a symmetric cipher to protect data from being read by parties without the cryptographic key. Data may be encrypted using a key shared by a group of devices (typically stored as the default key) or using a key shared between two peers (typically stored in an individual ACL entry). In this standard, data encryption may be provided on beacon payloads, command payloads, and data payloads.

5.4.6.1.3 Frame integrity

In this standard frame integrity is a security service that uses a message integrity code (MIC) to protect data from being modified by parties without the cryptographic key. It further provides assurance that data came from a party with the cryptographic key. In this standard, integrity may be provided on data frames, beacon frames, and command frames. The key used to provide frame integrity may be shared by a group of devices (typically stored as the default key) or by two peers (typically stored in an individual ACL entry).

5.4.6.1.4 Sequential freshness

Sequential freshness is a security service that uses an ordered sequence of inputs to reject frames that have been replayed. When a frame is received, the freshness value is compared with the last known freshness value. If the freshness value is newer than the last known value, the check has passed, and the freshness value is updated to the new value. If the freshness value is not newer than the last known freshness value, the check has failed. This service provides evidence that the received data are newer than the last data received by that device, but it does not provide a strict sense of time.

5.4.6.2 Security modes

Depending on the mode in which the device is operating and the security suite selected, the MAC sublayer may provide different security services.

5.4.6.2.1 Unsecured mode

Because security is not used for unsecured mode, no security services are provided by devices operating in unsecured mode.

5.4.6.2.2 ACL mode

Devices operating in ACL mode provide limited security services for communications with other devices. While in ACL mode, the higher layer may chose to reject frames based on whether the MAC sublayer indicates that a frame is purported to originate from a specific device. Because cryptographic protection is not provided in the MAC sublayer in this mode, the higher layer should implement other mechanisms to ensure the identity of the sending device. The service that is provided while in ACL mode is access control (see 5.4.6.1.1).

5.4.6.2.3 Secured mode

Devices operating in secured mode may provide any of the security services defined in 5.4.6.1. The specific security services are dependent on the security suite in use, and these services are specified by the security suite itself. Services that may be provided while in secured mode include

- Access control (see 5.4.6.1.1)
- Data encryption (see 5.4.6.1.2)
- Frame integrity (see 5.4.6.1.3)
- Sequential freshness (see 5.4.6.1.4)

5.5 Concept of primitives

This subclause provides a brief overview of the concept of service primitives. Refer to IEEE Std 802.2-1998[5] for more detailed information.

The services of a layer are the capabilities it offers to the user in the next higher layer or sublayer by building its functions on the services of the next lower layer. This concept is illustrated in Figure 14, showing the service hierarchy and the relationship of the two correspondent N-users and their associated N-layer (or sublayer) peer protocol entities.

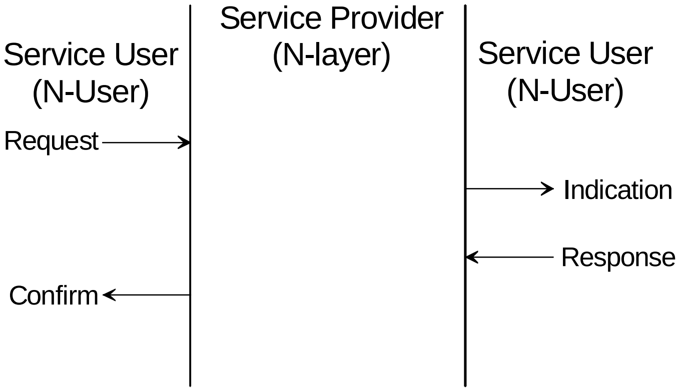


Figure 14—Service primitives

The services are specified by describing the information flow between the N-user and the N-layer. This information flow is modeled by discrete, instantaneous events, which characterize the provision of a service. Each event consists of passing a service primitive from one layer to the other through a layer SAP associated with an N-user. Service primitives convey the required information by providing a particular service. These service primitives are an abstraction because they specify only the provided service rather than the means by which it is provided. This definition is independent of any other interface implementation.

Services are specified by describing the service primitives and parameters that characterize it. A service may have one or more related primitives that constitute the activity that is related to that particular service. Each service primitive may have zero or more parameters that convey the information required to provide the service.

A primitive can be one of four generic types:

[5] For information on references, see Clause 2.

- **Request:** The request primitive is passed from the N-user to the N-layer to request that a service is initiated.
- **Indication:** The indication primitive is passed from the N-layer to the N-user to indicate an internal N-layer event that is significant to the N-user. This event may be logically related to a remote service request, or it may be caused by an N-layer internal event.
- **Response:** The response primitive is passed from the N-user to the N-layer to complete a procedure previously invoked by an indication primitive.
- **Confirm:** The confirm primitive is passed from the N-layer to the N-user to convey the results of one or more associated previous service requests.

6. PHY specification

This clause specifies two PHY options for IEEE 802.15.4. The PHY is responsible for the following tasks:

- Activation and deactivation of the radio transceiver
- ED within the current channel
- LQI for received packets
- CCA for CSMA-CA
- Channel frequency selection
- Data transmission and reception

Constants and attributes that are specified and maintained by the PHY are written in the text of this clause in italics. Constants have a general prefix of “a”, e.g., *aMaxPHYPacketSize*, and are listed in Table 18. Attributes have a general prefix of “phy”, e.g., *phyCurrentChannel*, and are listed in Table 19.

6.1 General requirements and definitions

This subclause specifies requirements that are common to both of the IEEE 802.15.4 PHYs.

6.1.1 Operating frequency range

A compliant device shall operate in one or several frequency bands using the modulation and spreading formats summarized in Table 1.

Table 1—Frequency bands and data rates

PHY (MHz)	Frequency band (MHz)	Spreading parameters		Data parameters		
		Chip rate (kchip/s)	**Modulation**	**Bit rate (kb/s)**	**Symbol rate (ksymbol/s)**	**Symbols**
868/915	868–868.6	300	BPSK	20	20	Binary
	902–928	600	BPSK	40	40	Binary
2450	2400–2483.5	2000	O-QPSK	250	62.5	16-ary Orthogonal

This standard is intended to conform with established regulations in Europe, Japan, Canada, and the United States. The regulatory documents listed below are for information only and are subject to change and revisions at any time. IEEE 802.15.4 devices shall also comply with specific regional legislation. Additional regulatory information is provided in Annex F.

Europe:

- Approval standards: European Telecommunications Standards Institute (ETSI)
- Documents: ETSI EN 300 328-1 [B11][6], ETSI EN 300 328-2 [B12], ETSI EN 300 220-1 [B10], ERC 70-03 [B13]
- Approval authority: National type approval authorities

[6] The numbers in brackets correspond to the numbers of the bibliography in Annex G.

Japan:

- Approval standards: Association of Radio Industries and Businesses (ARIB)
- Document: ARIB STD-T66 [B14]
- Approval authority: Ministry of Public Management, Home Affairs, Posts and Telecommunications (MPHPT)

United States:

- Approval standards: Federal Communications Commission (FCC), United States
- Document: FCC CFR47, Section 15.247 [B14]

Canada:

- Approval standards: Industry Canada (IC), Canada
- Document: GL36 [B15]

6.1.2 Channel assignments and numbering

A total of 27 channels, numbered 0 to 26, are available across the three frequency bands. Sixteen channels are available in the 2450 MHz band, 10 in the 915 MHz band, and 1 in the 868 MHz band. The center frequency of these channels is defined as follows:

$$F_c = 868.3 \text{ in megahertz, for } k = 0$$

$$F_c = 906 + 2\,(k - 1) \text{ in megahertz, for } k = 1, 2, ..., 10$$

$$\text{and} \qquad F_c = 2405 + 5\,(k - 11) \text{ in megahertz, for } k = 11, 12, ..., 26$$

where

k is the channel number.

For each PHY supported, a compliant device shall support all channels allowed by regulations for the region in which the device operates.

6.1.3 RF power measurement

Unless otherwise stated, all RF power measurements, either transmit or receive, shall be made at the appropriate transceiver to antenna connector. The measurements shall be made with equipment that is either matched to the impedance of the antenna connector or corrected for any mismatch. For devices without an antenna connector, the measurements shall be interpreted as effective isotropic radiated power (EIRP) (i.e., a 0 dBi gain antenna); and any radiated measurements shall be corrected to compensate for the antenna gain in the implementation.

6.1.4 Transmit power

The maximum transmit power shall conform with local regulations. Refer to Annex F for additional information on regulatory limits. A compliant device shall have its nominal transmit power level indicated by its PHY parameter, *phyTransmitPower* (see 6.4).

6.1.5 Out-of-band spurious emission

The out-of-band spurious emissions shall conform with local regulations. Refer to Annex F for additional information on regulatory limits on out-of-band emissions.

6.1.6 Receiver sensitivity definitions

The definitions in Table 2 are referenced by subclauses elsewhere in this standard regarding receiver sensitivity.

Table 2—Receiver sensitivity definitions

Term	Definition of term	Conditions
Packet error rate (PER)	Average fraction of transmitted packets that are not detected correctly.	– Average measured over random PSDU data.
Receiver sensitivity	Threshold input signal power that yields a specified PER.	– PSDU length = 20 octets. – PER < 1%. – Power measured at antenna terminals. – Interference not present.

6.2 PHY service specifications

The PHY provides an interface between the MAC sublayer and the physical radio channel, via the RF firmware and RF hardware. The PHY conceptually includes a management entity called the PLME. This entity provides the layer management service interfaces through which layer management functions may be invoked. The PLME is also responsible for maintaining a database of managed objects pertaining to the PHY. This database is referred to as the PHY PAN information base (PIB).

Figure 15 depicts the components and interfaces of the PHY.

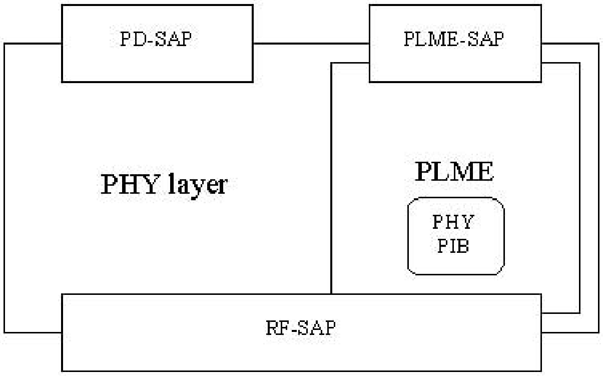


Figure 15—The PHY reference model

The PHY provides two services, accessed through two SAPs: the PHY data service, accessed through the PHY data SAP (PD-SAP), and the PHY management service, accessed through the PLME's SAP (PLME-SAP).

6.2.1 PHY data service

The PD-SAP supports the transport of MPDUs between peer MAC sublayer entities. Table 3 lists the primitives supported by the PD-SAP. These primitives are discussed in the subclauses referenced in the table.

Table 3—PD-SAP primitives

PD-SAP primitive	Request	Confirm	Indication
PD-DATA	6.2.1.1	6.2.1.2	6.2.1.3

6.2.1.1 PD-DATA.request

The PD-DATA.request primitive requests the transfer of an MPDU (i.e., PSDU) from the MAC sublayer to the local PHY entity.

6.2.1.1.1 Semantics of the service primitive

The semantics of the PD-DATA.request primitive is as follows:

```
PD-DATA.request                    (
                                   psduLength,
                                   psdu
                                   )
```

Table 4 specifies the parameters for the PD-DATA.request primitive.

Table 4—PD-DATA.request parameters

Name	Type	Valid range	Description
psduLength	Unsigned Integer	≤ *aMaxPHYPacketSize*	The number of octets contained in the PSDU to be transmitted by the PHY entity.
psdu	Set of octets	—	The set of octets forming the PSDU to be transmitted by the PHY entity.

6.2.1.1.2 When generated

The PD-DATA.request primitive is generated by a local MAC sublayer entity and issued to its PHY entity to request the transmission of an MPDU.

6.2.1.1.3 Effect on receipt

The receipt of the PD-DATA.request primitive by the PHY entity will cause the transmission of the supplied PSDU. Provided the transmitter is enabled (TX_ON state), the PHY will first construct a PPDU, containing the supplied PSDU, and then transmit the PPDU. When the PHY entity has completed the transmission, it will issue the PD-DATA.confirm primitive with a status of SUCCESS.

If the PD-DATA.request primitive is received while the receiver is enabled (RX_ON state) or if the transceiver is disabled (TRX_OFF state), the PHY entity will issue the PD-DATA.confirm primitive with a status of RX_ON or TRX_OFF, respectively.

6.2.1.2 PD-DATA.confirm

The PD-DATA.confirm primitive confirms the end of the transmission of an MPDU (i.e., PSDU) from a local MAC sublayer entity to a peer MAC sublayer entity.

6.2.1.2.1 Semantics of the service primitive

The semantics of the PD-DATA.confirm primitive is as follows:

```
PD-DATA.confirm                    (
                                   status
                                   )
```

Table 5 specifies the parameters for the PD-DATA.confirm primitive.

Table 5—PD-DATA.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, RX_ON, or TRX_OFF	The result of the request to transmit a packet.

6.2.1.2.2 When generated

The PD-DATA.confirm primitive is generated by the PHY entity and issued to its MAC sublayer entity in response to a PD-DATA.request primitive. The PD-DATA.confirm primitive will return a status of either SUCCESS, indicating that the request to transmit was successful, or an error code of RX_ON or TRX_OFF. The reasons for these status values are fully described in 6.2.1.1.3.

6.2.1.2.3 Effect on receipt

On receipt of the PD-DATA.confirm primitive, the MAC sublayer entity is notified of the result of its request to transmit. If the transmission attempt was successful, the status parameter is set to SUCCESS. Otherwise, the status parameter will indicate the error.

6.2.1.3 PD-DATA.indication

The PD-DATA.indication primitive indicates the transfer of an MPDU (i.e., PSDU) from the PHY to the local MAC sublayer entity.

6.2.1.3.1 Semantics of the service primitive

The semantics of the PD-DATA.indication primitive is as follows:

```
PD-DATA.indication          (
                            psduLength,
                            psdu,
                            ppduLinkQuality
                            )
```

Table 6 specifies the parameters for the PD-DATA.indication primitive.

Table 6—PD-DATA.indication parameters

Name	Type	Valid range	Description
psduLength	Unsigned Integer	≤ *aMaxPHYPacketSize*	The number of octets contained in the PSDU received by the PHY entity.
psdu	Set of octets	—	The set of octets forming the PSDU received by the PHY entity.
ppduLinkQuality	Integer	0 x 00–0 x ff	Link quality (LQ) value measured during reception of the PPDU (see 6.7.8).

6.2.1.3.2 When generated

The PD-DATA.indication primitive is generated by the PHY entity and issued to its MAC sublayer entity to transfer a received PSDU. This primitive will not be generated if the received psduLength field is zero or greater than *aMaxPHYPacketSize*.

6.2.1.3.3 Effect on receipt

On receipt of the PD-DATA.indication primitive, the MAC sublayer is notified of the arrival of an MPDU across the PHY data service.

6.2.2 PHY management service

The PLME-SAP allows the transport of management commands between the MLME and the PLME. Table 7 lists the primitives supported by the PLME-SAP. These primitives are discussed in the clauses referenced in the table.

Table 7—PLME-SAP primitives

PLME-SAP primitive	Request	Confirm
PLME-CCA	6.2.2.1	6.2.2.2
PLME-ED	6.2.2.3	6.2.2.4
PLME-GET	6.2.2.5	6.2.2.6

Table 7—PLME-SAP primitives *(continued)*

PLME-SAP primitive	Request	Confirm
PLME-SET-TRX-STATE	6.2.2.7	6.2.2.8
PLME-SET	6.2.2.9	6.2.2.10

6.2.2.1 PLME-CCA.request

The PLME-CCA.request primitive requests that the PLME perform a CCA as defined in 6.7.9.

6.2.2.1.1 Semantics of the service primitive

The semantics of the PLME-CCA.request primitive is as follows:

```
PLME-CCA.request                    ()
```

There are no parameters associated with the PLME-CCA.request primitive.

6.2.2.1.2 When generated

The PLME-CCA.request primitive is generated by the MLME and issued to its PLME whenever the CSMA-CA algorithm requires an assessment of the channel.

6.2.2.1.3 Effect on receipt

If the receiver is enabled on receipt of the PLME-CCA.request primitive, the PLME will cause the PHY to perform a CCA. When the PHY has completed the CCA, the PLME will issue the PLME-CCA.confirm primitive with a status of either BUSY or IDLE, depending on the result of the CCA.

If the PLME-CCA.request primitive is received while the transceiver is disabled (TRX_OFF state) or if the transmitter is enabled (TX_ON state), the PLME will issue the PLME-CCA.confirm primitive with a status of TRX_OFF or TX_ON, respectively.

6.2.2.2 PLME-CCA.confirm

The PLME-CCA.confirm primitive reports the results of a CCA.

6.2.2.2.1 Semantics of the service primitive

The semantics of the PLME-CCA.confirm primitive is as follows:

```
PLME-CCA.confirm                    (
                                    status
                                    )
```

Table 8 specifies the parameters for the PLME-CCA.confirm primitive.

Table 8—PLME-CCA.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	TRX_OFF, TX_ON, BUSY, or IDLE	The result of the request to perform a CCA.

6.2.2.2.2 When generated

The PLME-CCA.confirm primitive is generated by the PLME and issued to its MLME in response to a PLME-CCA.request primitive. The PLME-CCA.confirm primitive will return a status of either BUSY or IDLE, indicating a successful CCA, or an error code of TRX_OFF or TX_ON. The reasons for these status values are fully described in 6.2.2.1.3.

6.2.2.2.3 Effect on receipt

On receipt of the PLME-CCA.confirm primitive, the MLME is notified of the results of the CCA. If the CCA attempt was successful, the status parameter is set to either BUSY or IDLE. Otherwise, the status parameter will indicate the error.

6.2.2.3 PLME-ED.request

The PLME-ED.request primitive requests that the PLME perform an ED measurement (see 6.7.7).

6.2.2.3.1 Semantics of the service primitive

The semantics of the PLME-ED.request primitive is as follows:

```
PLME-ED.request                    ()
```

There are no parameters associated with the PLME-ED.request primitive.

6.2.2.3.2 When generated

The PLME-ED.request primitive is generated by the MLME and issued to its PLME to request an ED measurement.

6.2.2.3.3 Effect on receipt

If the receiver is enabled on receipt of the PLME-ED.request primitive, the PLME will cause the PHY to perform an ED measurement. When the PHY has completed the ED measurement, the PLME will issue the PLME-ED.confirm primitive with a status of SUCCESS.

If the PLME-ED.request primitive is received while the transceiver is disabled (TRX_OFF state) or if the transmitter is enabled (TX_ON state), the PLME will issue the PLME-ED.confirm primitive with a status of TRX_OFF or TX_ON, respectively.

6.2.2.4 PLME-ED.confirm

The PLME-ED.confirm primitive reports the results of the ED measurement.

6.2.2.4.1 Semantics of the service primitive

The semantics of the PLME-ED.confirm primitive is as follows:

```
PLME-ED.confirm                 (
                                status,
                                EnergyLevel
                                )
```

Table 9 specifies the parameters for the PLME-ED.confirm primitive.

Table 9—PLME-ED.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, TRX_OFF, or TX_ON	The result of the request to perform an ED measurement.
EnergyLevel	Integer	0 x 00–0 x ff	ED level for the current channel.

6.2.2.4.2 When generated

The PLME-ED.confirm primitive is generated by the PLME and issued to its MLME in response to a PLME-ED.request primitive. The PLME-ED.confirm primitive will return a status of SUCCESS, indicating a successful ED measurement, or an error code of TRX_OFF or TX_ON. The reasons for these status values are fully described in 6.2.2.3.3.

6.2.2.4.3 Effect on receipt

On receipt of the PLME-ED.confirm primitive, the MLME is notified of the results of the ED measurement. If the ED measurement attempt was successful, the status parameter is set to SUCCESS. Otherwise, the status parameter will indicate the error.

6.2.2.5 PLME-GET.request

The PLME-GET.request primitive requests information about a given PHY PIB attribute.

6.2.2.5.1 Semantics of the service primitive

The semantics of the PLME-GET.request primitive is as follows:

```
PLME-GET.request                (
                                PIBAttribute
                                )
```

Table 10 specifies the parameters for the PLME-GET.request primitive.

Table 10—PLME-GET.request parameters

Name	Type	Valid range	Description
PIBAttribute	Enumeration	See Table 19	The identifier of the PHY PIB attribute to get.

6.2.2.5.2 When generated

The PLME-GET.request primitive is generated by the MLME and issued to its PLME to obtain information from the PHY PIB.

6.2.2.5.3 Effect on receipt

On receipt of the PLME-GET.request primitive, the PLME will attempt to retrieve the requested PHY PIB attribute from its database. If the identifier of the PIB attribute is not found in the database, the PLME will issue the PLME-GET.confirm primitive with a status of UNSUPPORTED_ATTRIBUTE.

If the requested PHY PIB attribute is successfully retrieved, the PLME will issue the PLME-GET.confirm primitive with a status of SUCCESS.

6.2.2.6 PLME-GET.confirm

The PLME-GET.confirm primitive reports the results of an information request from the PHY PIB.

6.2.2.6.1 Semantics of the service primitive

The semantics of the PLME-GET.confirm primitive is as follows:

```
PLME-GET.confirm                    (
                                    status,
                                    PIBAttribute,
                                    PIBAttributeValue
                                    )
```

Table 11 specifies the parameters for the PLME-GET.confirm primitive.

Table 11—PLME-GET.confirm parameters

Name	Type	Valid range	Description
Status	Enumeration	SUCCESS or UNSUPPORTED_ATTRIBUTE	The result of the request for PHY PIB attribute information.
PIBAttribute	Enumeration	See Table 19	The identifier of the PHY PIB attribute to get.
PIBAttributeValue	Various	Attribute specific	The value of the indicated PHY PIB attribute to get.

6.2.2.6.2 When generated

The PLME-GET.confirm primitive is generated by the PLME and issued to its MLME in response to a PLME-GET.request primitive. The PLME-GET.confirm primitive will return a status of either SUCCESS, indicating that the request to read a PHY PIB attribute was successful, or an error code of UNSUPPORTED_ATTRIBUTE. The reasons for these status values are fully described in subclause 6.2.2.5.3.

6.2.2.6.3 Effect on receipt

On receipt of the PLME-GET.confirm primitive, the MLME is notified of the results of its request to read a PHY PIB attribute. If the request to read a PHY PIB attribute was successful, the status parameter is set to SUCCESS. Otherwise, the status parameter will indicate the error.

6.2.2.7 PLME-SET-TRX-STATE.request

The PLME-SET-TRX-STATE.request primitive requests that the PHY entity change the internal operating state of the transceiver. The transceiver will have three main states:

- Transceiver disabled (TRX_OFF).
- Transmitter enabled (TX_ON).
- Receiver enabled (RX_ON).

6.2.2.7.1 Semantics of the service primitive

The semantics of the PLME-SET-TRX-STATE.request primitive is as follows:

```
PLME-SET-TRX-STATE.request      (
                                state
                                )
```

Table 12 specifies the parameters for the PLME-SET-TRX-STATE.request primitive.

Table 12—PLME-SET-TRX-STATE.request parameters

Name	Type	Valid range	Description
state	Enumeration	RX_ON, TRX_OFF, FORCE_TRX_OFF, or TX_ON	The new state in which to configure the transceiver.

6.2.2.7.2 When generated

The PLME-SET-TRX-STATE.request primitive is generated by the MLME and issued to its PLME when the current operational state of the receiver needs to be changed.

6.2.2.7.3 Effect on receipt

On receipt of the PLME-SET-TRX-STATE.request primitive, the PLME will cause the PHY to change to the requested state. If the state change is accepted, the PHY will issue the PLME-SET-TRX-STATE.confirm primitive with a status of SUCCESS. If this primitive requests a state that the transceiver is already configured, the PHY will issue the PLME-SET-TRX-STATE.confirm primitive with a status indicating the current state, i.e., RX_ON, TRX_OFF, or TX_ON. If this primitive is issued with RX_ON or TRX_OFF argument and the PHY is busy transmitting a PPDU, the PHY will issue the PLME-SET-TRX-STATE.confirm primitive with a status BUSY_TX and defer the state change till the end of transmission. If this primitive is issued with TX_ON or TRX_OFF argument and the PHY is in RX_ON state and has already received a valid SFD, the PHY will issue the PLME-SET-TRX-STATE.confirm primitive with a status BUSY_RX and defer the state change till the end of reception of the PPDU. If this primitive is issued with FORCE_TRX_OFF, the PHY will cause the PHY to go the TRX_OFF state irrespective of the state the PHY is in.

6.2.2.8 PLME-SET-TRX-STATE.confirm

The PLME-SET-TRX-STATE.confirm primitive reports the result of a request to change the internal operating state of the transceiver.

6.2.2.8.1 Semantics of the service primitive

The semantics of the PLME-SET-TRX-STATE.confirm primitive is as follows:

```
PLME-SET-TRX-STATE.confirm        (
                                  status
                                  )
```

Table 13 specifies the parameters for the PLME-SET-TRX-STATE.confirm primitive.

Table 13—PLME-SET-TRX-STATE.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, RX_ON, TRX_OFF, TX_ON, BUSY_RX, or BUSY_TX	The result of the request to change the state of the transceiver.

6.2.2.8.2 When generated

The PLME-SET-TRX-STATE.confirm primitive is generated by the PLME and issued to its MLME after attempting to change the internal operating state of the transceiver.

6.2.2.8.3 Effect on receipt

On receipt of the PLME-SET-TRX-STATE.confirm primitive, the MLME is notified of the result of its request to change the internal operating state of the transceiver. A status value of SUCCESS indicates that the internal operating state of the transceiver was accepted. A status value of RX_ON, TRX_OFF, or TX_ON indicates that the transceiver is already in the requested internal operating state. A status value of BUSY_TX is issued when the PHY is requested to change its state to RX_ON or TRX_OFF while transmitting. A status value of BUSY_RX is issued when the PHY is in RX_ON state, has already received a valid SFD, and is requested to change its state to TX_ON or TRX_OFF.

6.2.2.9 PLME-SET.request

The PLME-SET.request primitive attempts to set the indicated PHY PIB attribute to the given value.

6.2.2.9.1 Semantics of the service primitive

The semantics of the PLME-SET.request primitive is as follows:

```
PLME-SET.request                (
                                PIBAttribute,
                                PIBAttributeValue
                                )
```

Table 14 specifies the parameters for the PLME-SET.request primitive.

Table 14—PLME-SET.request parameters

Name	Type	Valid range	Description
PIBAttribute	Enumeration	See Table 19	The identifier of the PIB attribute to set.
PIBAttributeValue	Various	Attribute specific	The value of the indicated PIB attribute to set.

6.2.2.9.2 When generated

The PLME-SET.request primitive is generated by the MLME and issued to its PLME to write the indicated PHY PIB attribute.

6.2.2.9.3 Effect on receipt

On receipt of the PLME-SET.request primitive, the PLME will attempt to write the given value to the indicated PHY PIB attribute in its database. If the PIBAttribute parameter specifies an attribute that is not found in the database (see Table 19), the PLME will issue the PLME-SET.confirm primitive with a status of UNSUPPORTED_ATTRIBUTE. If the PIBAttibuteValue parameter specifies a value that is out of the valid range for the given attribute, the PLME will issue the PLME-SET.confirm primitive with a status of INVALID_PARAMETER.

If the requested PHY PIB attribute is successfully written, the PLME will issue the PLME-SET.confirm primitive with a status of SUCCESS.

6.2.2.10 PLME-SET.confirm

The PLME-SET.confirm primitive reports the results of the attempt to set a PIB attribute.

6.2.2.10.1 Semantics of the service primitive

The semantics of the PLME-SET.confirm primitive is as follows:

```
PLME-SET.confirm                (
                                status,
                                PIBAttribute
                                )
```

Table 15 specifies the parameters for the PLME-SET.confirm primitive.

Table 15—PLME-SET.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, UNSUPPORTED_ATTRIBUTE, or INVALID_PARAMETER	The status of the attempt to set the request PIB attribute.
PIBAttribute	Enumeration	See Table 19	The identifier of the PIB attribute being confirmed.

6.2.2.10.2 When generated

The PLME-SET.confirm primitive is generated by the PLME and issued to its MLME in response to a PLME-SET.request primitive. The PLME-SET.confirm primitive will return a status of either SUCCESS, indicating that the requested value was written to the indicated PHY PIB attribute, or an error code of UNSUPPORTED_ATTRIBUTE or INVALID_PARAMETER. The reasons for these status values are fully described in subclause 6.2.2.9.3.

6.2.2.10.3 Effect on receipt

On receipt of the PLME-SET.confirm primitive, the MLME is notified of the result of its request to set the value of a PHY PIB attribute. If the requested value was written to the indicated PHY PIB attribute, the status parameter is set to SUCCESS. Otherwise, the status parameter will indicate the error.

6.2.3 PHY enumerations description

Table 16 shows a description of the PHY enumeration values defined in the PHY specification.

Table 16—PHY enumerations description

Enumeration	Value	Description
BUSY	0 x 00	The CCA attempt has detected a busy channel.
BUSY_RX	0 x 01	The transceiver is asked to change its state while receiving.
BUSY_TX	0 x 02	The transceiver is asked to change its state while transmitting.
FORCE_TRX_OFF	0 x 03	The transceiver is to be switched off.
IDLE	0 x 04	The CCA attempt has detected an idle channel.
INVALID_PARAMETER	0 x 05	A SET/GET request was issued with a parameter in the primitive that is out of the valid range.
RX_ON	0 x 06	The transceiver is in or is to be configured into the receiver enabled state.
SUCCESS	0 x 07	A SET/GET, an ED operation, or a transceiver state change was successful.

Table 16—PHY enumerations description *(continued)*

Enumeration	Value	Description
TRX_OFF	0 x 08	The transceiver is in or is to be configured into the transceiver disabled state.
TX_ON	0 x 09	The transceiver is in or is to be configured into the transmitter enabled state.
UNSUPPORTED_ATTRIBUTE	0 x 0a	A SET/GET request was issued with the identifier of an attribute that is not supported.

6.3 PPDU format

This clause specifies the format of the PPDU packet.

For convenience, the PPDU packet structure is presented so that the leftmost field as written in this standard shall be transmitted or received first. All multiple octet fields shall be transmitted or received least significant octet first and each octet shall be transmitted or received least significant bit (LSB) first. The same transmission order should apply to data fields transferred between the PHY and MAC sublayer.

Each PPDU packet consists of the following basic components:

— A SHR, which allows a receiving device to synchronize and lock onto the bit stream.
— A PHR, which contains frame length information.
— A variable length payload, which carries the MAC sublayer frame.

6.3.1 General packet format

The PPDU packet structure shall be formatted as illustrated in Figure 16.

Octets: 4	1	1		variable
Preamble	SFD	Frame length (7 bits)	Reserved (1 bit)	PSDU
SHR		PHR		PHY payload

Figure 16—Format of the PPDU

6.3.1.1 Preamble field

The preamble field is used by the transceiver to obtain chip and symbol synchronization with an incoming message. The preamble field shall be composed of 32 binary zeros.

6.3.1.2 SFD field

The SFD is an 8 bit field indicating the end of the synchronization (preamble) field and the start of the packet data. The SFD shall be formatted as illustrated in Figure 17.

Bits: 0	1	2	3	4	5	6	7
1	1	1	0	0	1	0	1

Figure 17—Format of the SFD field

6.3.1.3 Frame length field

The frame length field is 7 bits in length and specifies the total number of octets contained in the PSDU (i.e., PHY payload). It is a value between 0 and *aMaxPHYPacketSize* (see 6.4). Table 17 summarizes the type of payload versus the frame length value.

Table 17—Frame length values

Frame length values	Payload
0–4	Reserved
5	MPDU (Acknowledgment)
6–7	Reserved
8 to *aMaxPHYPacketSize*	MPDU

6.3.1.4 PSDU field

The PSDU field has a variable length and carries the data of the PHY packet. For all packet types of length five octets or greater than seven octets, the PSDU contains the MAC sublayer frame (i.e., MPDU).

6.4 PHY constants and PIB attributes

This subclause specifies the constants and attributes required by the PHY.

6.4.1 PHY constants

The constants that define the characteristics of the PHY are presented in Table 18. These constants are hardware dependent and cannot be changed during operation.

Table 18—PHY constants

Constant	Description	Value
aMaxPHYPacketSize	The maximum PSDU size (in octets) the PHY shall be able to receive.	127
aTurnaroundTime	RX-to-TX or TX-to-RX maximum turnaround time (see 6.7.1 and 6.7.2)	12 symbol periods

6.4.2 PHY PIB attributes

The PHY PIB comprises the attributes required to manage the PHY of a device. Each of these attributes can be read or written using the PLME-GET.request and PLME-SET.request primitives, respectively. The attributes contained in the PHY PIB are presented in Table 19.

Table 19—PHY PIB attributes

Attribute	Identifier	Type	Range	Description
phyCurrentChannel	0 x 00	Integer	0–26	The RF channel to use for all following transmissions and receptions (see 6.1.2).
phyChannelsSupported	0 x 01	Bitmap	See description	The 5 most significant bits (MSBs) (b_{27},... , b_{31}) of *phyChannelsSupported* shall be reserved and set to 0, and the 27 LSBs (b_0, b_1, ... b_{26}) shall indicate the status (1=available, 0=unavailable) for each of the 27 valid channels (b_k shall indicate the status of channel *k* as in 6.1.2).
phyTransmitPower	0 x 02	Bitmap	0 x 00–0xbf	The 2 MSBs represent the tolerance on the transmit power: 00 = ± 1 dB 01 = ± 3 dB 10 = ± 6 dB The 6 LSBs represent a signed integer in twos-complement format, corresponding to the nominal transmit power of the device in decibels relative to 1 mW. The lowest value of *phyTransmitPower* shall be interpreted as less than or equal to –32 dBm.
phyCCAMode	0 x 03	Integer	1–3	The CCA mode (see 6.7.9).

6.5 2450 MHz PHY specifications

The requirements for the 2450 MHz PHY are specified in 6.5.1 through 6.5.3.

6.5.1 Data rate

The data rate of the IEEE 802.15.4 (2450 MHz) PHY shall be 250 kb/s.

6.5.2 Modulation and spreading

The 2450 MHz PHY employs a 16-ary quasi-orthogonal modulation technique. During each data symbol period, four information bits are used to select one of 16 nearly orthogonal pseudo-random noise (PN) sequences to be transmitted. The PN sequences for successive data symbols are concatenated, and the aggregate chip sequence is modulated onto the carrier using offset quadrature phase-shift keying (O-QPSK).

6.5.2.1 Reference modulator diagram

The functional block diagram in Figure 18 is provided as a reference for specifying the 2450 MHz PHY modulation and spreading functions. The number in each block refers to the subclause that describes that function.

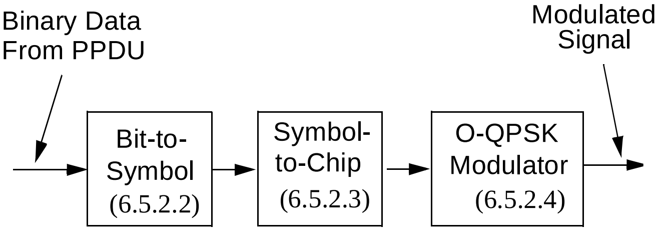


Figure 18—Modulation and spreading functions

6.5.2.2 Bit-to-symbol mapping

All binary data contained in the PPDU shall be encoded using the modulation and spreading functions shown in Figure 18. This subclause describes how binary information is mapped into data symbols.

The 4 LSBs (b_0, b_1, b_2, b_3) of each octet shall map into one data symbol, and the 4 MSBs (b_4, b_5, b_6, b_7) of each octet shall map into the next data symbol. Each octet of the PPDU is processed through the modulation and spreading functions (see Figure 18) sequentially, beginning with the preamble field and ending with the last octet of the PSDU. Within each octet, the least significant symbol (b_0, b_1, b_2, b_3) is processed first and the most significant symbol (b_4, b_5, b_6, b_7) is processed second.

6.5.2.3 Symbol-to-chip mapping

Each data symbol shall be mapped into a 32-chip PN sequence as specified in Table 20. The PN sequences are related to each other through cyclic shifts and/or conjugation (i.e., inversion of odd-indexed chip values).

Table 20—Symbol-to-chip mapping

Data symbol (decimal)	Data symbol (binary) (b_0, b_1, b_2, b_3)	Chip values (c_0 c_1 ... c_{30} c_{31})
0	0 0 0 0	1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0
1	1 0 0 0	1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0
2	0 1 0 0	0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0
3	1 1 0 0	0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1
4	0 0 1 0	0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1
5	1 0 1 0	0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0
6	0 1 1 0	1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1
7	1 1 1 0	1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1
8	0 0 0 1	1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1
9	1 0 0 1	1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1
10	0 1 0 1	0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1
11	1 1 0 1	0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0
12	0 0 1 1	0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0
13	1 0 1 1	0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1

Table 20—Symbol-to-chip mapping *(continued)*

Data symbol (decimal)	Data symbol (binary) (b_0, b_1, b_2, b_3)	Chip values $(c_0\ c_1 \dots c_{30}\ c_{31})$
14	0 1 1 1	1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0
15	1 1 1 1	1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0

6.5.2.4 O-QPSK modulation

The chip sequences representing each data symbol are modulated onto the carrier using O-QPSK with half-sine pulse shaping. Even-indexed chips are modulated onto the in-phase (I) carrier and odd-indexed chips are modulated onto the quadrature-phase (Q) carrier. Because each data symbol is represented by a 32-chip sequence, the chip rate (nominally 2.0 Mchip/s) is 32 times the symbol rate. To form the offset between I-phase and Q-phase chip modulation, the Q-phase chips shall be delayed by T_c with respect to the I-phase chips (see Figure 19), where T_c is the inverse of the chip rate.

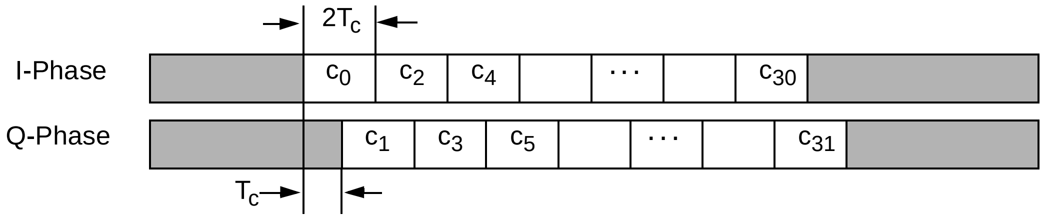


Figure 19—O-QPSK chip offsets

6.5.2.5 Pulse shape

The half-sine pulse shape used to represent each baseband chip is described by Equation (1):

$$p(t) = \begin{cases} \sin\left(\pi \dfrac{t}{2T_c}\right), & 0 \le t \le 2T_c \\ 0, & otherwise \end{cases} \tag{1}$$

Figure 20 shows a sample baseband chip sequence with half-sine pulse shaping.

6.5.2.6 Chip transmission order

During each symbol period the least significant chip, c_0, is transmitted first and the most significant chip, c_{31}, is transmitted last.

6.5.3 2450 MHz band radio specification

In addition to meeting regional regulatory requirements, devices operating in the 2450 MHz band shall also meet the radio requirements in 6.5.3.1 through 6.5.3.4.

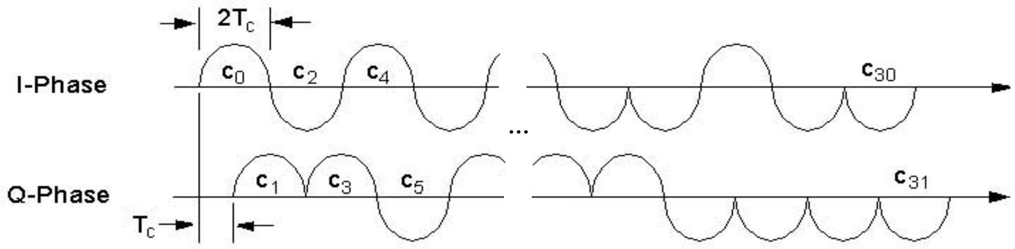


Figure 20—Sample baseband chip sequences with pulse shaping

6.5.3.1 Transmit power spectral density (PSD) mask

The transmitted spectral products shall be less than the limits specified in Table 21. For both relative and absolute limits, average spectral power shall be measured using a 100 kHz resolution bandwidth. For the relative limit, the reference level shall be the highest average spectral power measured within ± 1 MHz of the carrier frequency.

Table 21—Transmit PSD limits

Frequency	Relative limit	Absolute limit
$\|f - f_c\| > 3.5$ MHz	–20 dB	–30 dBm

6.5.3.2 Symbol rate

The 2450 MHz PHY symbol rate shall be 62.5 ksymbol/s ± 40 ppm.

6.5.3.3 Receiver sensitivity

Under the conditions specified in 6.1.6, a compliant device shall be capable of achieving a sensitivity of –85 dBm or better.

6.5.3.4 Receiver jamming resistance

The minimum jamming resistance levels are given in Table 22. The adjacent channel is one on either side of the desired channel that is closest in frequency to the desired channel, and the alternate channel is one more removed from the adjacent channel. For example, when channel 13 is the desired channel, channel 12 and channel 14 are the adjacent channels, and channel 11 and channel 15 are the alternate channels.

Table 22—Minimum receiver jamming resistance requirements for 2450 MHz PHY

Adjacent channel rejection	Alternate channel rejection
0 dB	30 dB

The adjacent channel rejection shall be measured as follows. The desired signal shall be a compliant 2450 MHz IEEE 802.15.4 signal of pseudo-random data. The desired signal is input to the receiver at a level 3 dB above the maximum allowed receiver sensitivity given in 6.5.3.3.

In either the adjacent or the alternate channel, an IEEE 802.15.4 signal is input at the relative level specified in Table 22. The test shall be performed for only one interfering signal at a time. The receiver shall meet the error rate criteria defined in 6.1.6 under these conditions.

6.6 868/915 MHz band PHY specifications

The requirements for the 868/915 MHz band PHY are specified in 6.6.1 through 6.6.3.

6.6.1 868/915 MHz band data rates

The data rate of the 868/915 MHz band PHY shall be 20 kb/s when operating in the 868 MHz band and 40 kb/s when operating in the 915 MHz band.

6.6.2 Modulation and spreading

The 868/915 MHz PHY shall employ direct sequence spread spectrum (DSSS) with binary phase-shift keying (BPSK) used for chip modulation and differential encoding used for data symbol encoding.

6.6.2.1 Reference modulator diagram

The functional block diagram in Figure 21 is provided as a reference for specifying the 868/915 MHz band PHY modulation and spreading functions. The number in each block refers to the subclause that describes that function. Each bit in the PPDU shall be processed through the differential encoding, bit-to-chip mapping and modulation functions in octet-wise order, beginning with the preamble field and ending with the last octet of the PSDU. Within each octet, the LSB, b0, is processed first and the MSB, b7, is processed last.

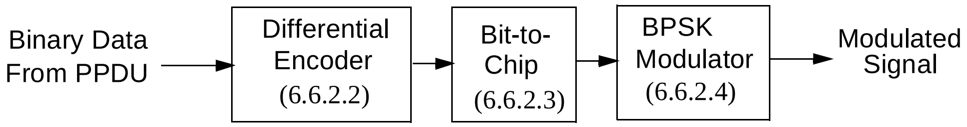


Figure 21—Modulation and spreading functions

6.6.2.2 Differential encoding

Differential encoding is the modulo-2 addition (exclusive or) of a raw data bit with the previous encoded bit. This is performed by the transmitter and can be described by Equation (2):

$$E_n = R_n \oplus E_{n-1} \tag{2}$$

where

R_n is the raw data bit being encoded,
E_n is the corresponding differentially encoded bit,
E_{n-1} is the previous differentially encoded bit.

For each packet transmitted, R_1 is the first raw data bit to be encoded and E_0 is assumed to be zero.

Conversely, the decoding process, as performed at the receiver, can be described by Equation (3):

$$R_n = E_n \oplus E_{n-1} \tag{3}$$

For each packet received, E_1 is the first bit to be decoded, and E_0 is assumed to be zero.

6.6.2.3 Bit-to-chip mapping

Each input bit shall be mapped into a 15-chip PN sequence as specified in Table 23.

Table 23—Symbol-to-chip mapping

Input bits	Chip values (c_0 c_1 ... c_{14})
0	1 1 1 1 0 1 0 1 1 0 0 1 0 0 0
1	0 0 0 0 1 0 1 0 0 1 1 0 1 1 1

6.6.2.4 BPSK modulation

The chip sequences are modulated onto the carrier using BPSK with raised cosine pulse shaping (roll-off factor = 1). The chip rate is 300 kchip/s for the 868 MHz band and 600 kchip/s in the 915 MHz band.

6.6.2.4.1 Pulse shape

The raised cosine pulse shape (roll-off factor = 1) used to represent each baseband chip is described by Equation (4):

$$p(t) = \frac{\sin(\pi t/T_c)}{\pi t/T_c}\,\frac{\cos(\pi t/T_c)}{1-(4t^2/T_c^2)} \tag{4}$$

6.6.2.4.2 Chip transmission order

During each symbol period, the least significant chip, c_0, is transmitted first, and the most significant chip, c_{14}, is transmitted last.

6.6.3 868/915 MHz band radio specification

In addition to meeting regional regulatory requirements, devices operating in the 868/915 MHz bands shall also meet the radio requirements in 6.6.3.1 through 6.6.3.5.

6.6.3.1 Operating frequency range

The 868/915 MHz PHY operates in the 868.0–868.6 MHz frequency band and in the 902–928 MHz frequency band.

6.6.3.2 915 MHz band transmit PSD mask

The transmitted spectral products shall be less than the limits specified in Table 24. For both relative and absolute limits, average spectral power shall be measured using a 100 kHz resolution bandwidth. For the relative limit, the reference level shall be the highest average spectral power measured within ± 600 kHz of the carrier frequency.

Table 24—915 MHz band transmit PSD limits

Frequency	Relative limit	Absolute limit
$\|f - f_c\| > 1.2$ MHz	–20 dB	–20 dBm

6.6.3.3 Symbol rate

The IEEE 802.15.4 PHY symbol rate shall be 20 ksymbol/s when operating in the 868 MHz band and 40 ksymbol/s when operating in the 915 MHz band with an accuracy of ± 40 ppm.

6.6.3.4 Receiver sensitivity

Under the conditions specified in 6.1.6, a compliant device shall be capable of achieving a sensitivity of –92 dBm or better.

6.6.3.5 Receiver jamming resistance

This subclause applies only to the 902–928 MHz band as there is only one channel available in the 868.0–868.6 MHz band.

The minimum jamming resistance levels are given in Table 25. The adjacent channel is one on either side of the desired channel that is closest in frequency to the desired channel, and the alternate channel is one more removed from the adjacent channel. For example, when channel 5 is the desired channel, channel 4 and channel 6 are the adjacent channels and channel 3 and channel 7 are the alternate channels.

Table 25—Minimum receiver jamming resistance requirements for 915 MHz PHY

Adjacent channel rejection	Alternate channel rejection
0 dB	30 dB

The adjacent channel rejection shall be measured as follows: The desired signal shall be a compliant 915 MHz IEEE 802.15.4 signal of pseudo-random data. The desired signal is input to the receiver at a level 3 dB above the maximum allowed receiver sensitivity given in 6.6.3.4.

In either the adjacent or the alternate channel, an IEEE 802.15.4 signal is input at the relative level specified in Table 25. The test shall be performed for only one interfering signal at a time. The receiver shall meet the error rate criteria defined in 6.1.6 under these conditions.

6.7 General radio specifications

The specifications in 6.7.1 through 6.7.9 apply to either or both the 2450 MHz PHY and the 868/915 MHz PHY.

6.7.1 TX-to-RX turnaround time

The TX-to-RX turnaround time shall be less than *aTurnaroundTime* (see 6.4.1).

The TX-to-RX turnaround time shall be measured at the air interface from the trailing edge of the last transmitted symbol until the receiver is ready to begin the reception of the next PHY packet.

6.7.2 RX-to-TX turnaround time

The RX-to-TX turnaround time shall be less than *aTurnaroundTime* (see 6.4.1).

The RX-to-TX turnaround time shall be measured at the air interface from the trailing edge of the last chip (of the last symbol) of a received packet until the transmitter is ready to begin transmission of the resulting acknowledgment. Actual transmission start times are specified by the MAC sublayer (see 7.5.6.4.2).

6.7.3 Error-vector magnitude (EVM) definition

The modulation accuracy of an IEEE 802.15.4 transmitter is determined with an EVM measurement. In order to calculate the EVM measurement, a time record of N received complex chip values $(\tilde{I}_j, \tilde{Q}_j)$ is captured. For each received complex chip, a decision is made about which complex chip value was transmitted. The ideal position of the chosen complex chip (the center of the decision box) is represented by the vector (I_j, Q_j). The error vector $(\delta I_j, \delta Q_j)$ is defined as the distance from this ideal position to the actual position of the received point (see Figure 22).

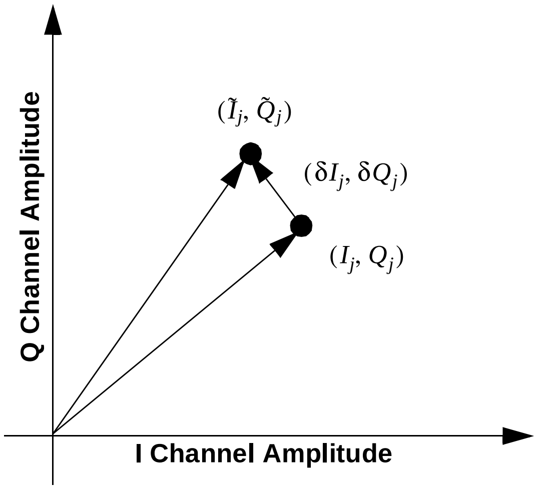


Figure 22—Error vector calculation

Thus, the received vector is the sum of the ideal vector and the error vector.

$$(\tilde{I}_j, \tilde{Q}_j) = (I_j, Q_j) + (\delta I_j, \delta Q\ ) \tag{5}$$

The EVM for IEEE Std 802.15.4-2003 is defined as

$$EVM \equiv \sqrt{\frac{\frac{1}{N}\sum_{j=1}^{N}(\delta I_j^{\,2} + \delta Q_j^{\,2})}{S^2}} \quad \text{X } 100\% \tag{6}$$

where

S is the magnitude of the vector to the ideal constellation point,

$(\delta I_j, \delta Q_j)$ is the error vector.

6.7.3.1 EVM calculated values

An IEEE 802.15.4 transmitter shall have EVM values of less than 35% when measured for 1000 chips. The error-vector measurement shall be made on baseband I and Q chips after recovery through a reference receiver system. The reference receiver shall perform carrier lock, symbol timing recovery, and amplitude adjustment while making the measurements.

6.7.4 Transmit center frequency tolerance

The transmitted center frequency tolerance shall be ± 40 ppm maximum.

6.7.5 Transmit power

An IEEE 802.15.4 transmitter shall be capable of transmitting at least –3 dBm. Devices should transmit lower power when possible in order to reduce interference to other devices and systems.

The maximum transmit power is limited by local regulatory bodies.

6.7.6 Receiver maximum input level of desired signal

The receiver maximum input level is the maximum power level of the desired signal, in decibels relative to 1 mW, present at the input of the receiver for which the error rate criterion in 6.1.6 is met. An IEEE 802.15.4 receiver shall have a receiver maximum input level greater than or equal to –20 dBm.

6.7.7 Receiver ED

The receiver ED measurement is intended for use by a network layer as part of a channel selection algorithm. It is an estimate of the received signal power within the bandwidth of an IEEE 802.15.4 channel. No attempt is made to identify or decode signals on the channel. The ED time shall be equal to 8 symbol periods.

The ED result shall be reported to the MLME using PLME-ED.confirm (see 6.2.2.4) as an 8 bit integer ranging from 0x00 to 0xff. The minimum ED value (0) shall indicate received power less than 10 dB above the specified receiver sensitivity (see 6.5.3.3 and 6.6.3.4), and the range of received power spanned by the ED values shall be at least 40 dB. Within this range, the mapping from the received power in decibels to ED value shall be linear with an accuracy of ± 6 dB.

6.7.8 LQI

The LQI measurement is a characterization of the strength and/or quality of a received packet. The measurement may be implemented using receiver ED, a signal-to-noise ratio estimation, or a combination of these methods. The use of the LQI result by the network or application layers is not specified in this standard.

The LQI measurement shall be performed for each received packet, and the result shall be reported to the MAC sublayer using PD-DATA.indication (see 6.2.1.3) as an integer ranging from 0x00 to 0xff. The minimum and maximum LQI values (0x00 and 0xff) should be associated with the lowest and highest quality IEEE 802.15.4 signals detectable by the receiver, and LQ values in between should be uniformly distributed between these two limits. At least eight unique values of LQ shall be used.

6.7.9 CCA

The IEEE 802.15.4 PHY shall provide the capability to perform CCA according to at least one of the following three methods:

- CCA Mode 1: Energy above threshold. CCA shall report a busy medium upon detecting any energy above the ED threshold.
- CCA Mode 2: Carrier sense only. CCA shall report a busy medium only upon the detection of a signal with the modulation and spreading characteristics of IEEE 802.15.4. This signal may be above or below the ED threshold.
- CCA Mode 3: Carrier sense with energy above threshold. CCA shall report a busy medium only upon the detection of a signal with the modulation and spreading characteristics of IEEE 802.15.4 with energy above the ED threshold.

For any of the CCA modes, if the PLME-CCA.request primitive (see 6.2.2.1) is received by the PHY during reception of a PPDU, CCA shall report a busy medium. PPDU reception is considered to be in progress following detection of the SFD, and it remains in progress until the number of octets specified by the decoded PHR has been received.

A busy channel shall be indicated by the PLME-CCA.confirm primitive (6.2.2.2) with a status of BUSY.

A clear channel shall be indicated by the PLME-CCA.confirm primitive (6.2.2.2) with a status of IDLE.

The PHY PIB attribute *phyCCAMode* (see 6.4) shall indicate the appropriate operation mode. The CCA parameters are subject to the following criteria:

a) The ED threshold shall be at most 10 dB above the specified receiver sensitivity (see 6.5.3.3 and 6.6.3.4).
b) The CCA detection time shall be equal to 8 symbol periods.

7. MAC sublayer specification

This clause specifies the MAC sublayer of IEEE Std 802.15.4-2003. The MAC sublayer handles all access to the physical radio channel and is responsible for the following tasks:

- Generating network beacons if the device is a coordinator.
- Synchronizing to the beacons.
- Supporting PAN association and disassociation.
- Supporting device security.
- Employing the CSMA-CA mechanism for channel access.
- Handling and maintaining the GTS mechanism.
- Providing a reliable link between two peer MAC entities.

Constants and attributes that are specified and maintained by the MAC sublayer are written in the text of this clause in italics. Constants have a general prefix of “a”, e.g., *aBaseSlotDuration*, and are listed in Table 70 (in 7.5). Attributes have a general prefix of “mac”, e.g., *macAckWaitDuration*, and are listed in Table 71 and Table 72 (in 7.5).

7.1 MAC sublayer service specification

The MAC sublayer provides an interface between the SSCS and the PHY. The MAC sublayer conceptually includes a management entity called the MLME. This entity provides the service interfaces through which layer management functions may be invoked. The MLME is also responsible for maintaining a database of managed objects pertaining to the MAC sublayer. This database is referred to as the MAC sublayer PIB.

Figure 23 depicts the components and interfaces of the MAC sublayer.

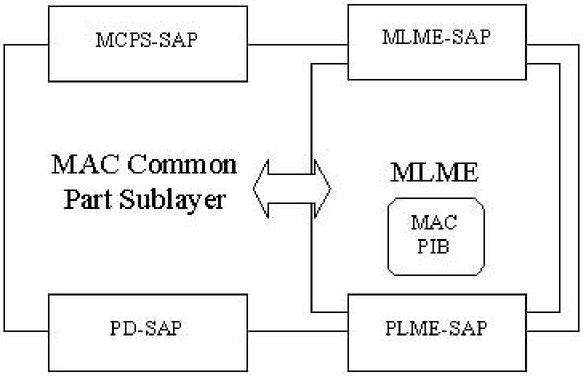


Figure 23—The MAC sublayer reference model

The MAC sublayer provides two services, accessed through two SAPs:

- The MAC data service, accessed through the MAC common part sublayer (MCPS) data SAP (MCPS-SAP), and
- The MAC management service, accessed through the MLME-SAP.

These two services provide the interface between the SSCS and the PHY, via the PD-SAP and PLME-SAP interfaces (see 6.2). In addition to these external interfaces, an implicit interface also exists between the MLME and the MCPS that allows the MLME to use the MAC data service.

7.1.1 MAC data service

The MCPS-SAP supports the transport of SSCS protocol data units (SPDUs) between peer SSCS entities. Table 26 lists the primitives supported by the MCPS-SAP. Primitives marked with a diamond (♦) are optional for an RFD. These primitives are discussed in the subclauses referenced in the table.

Table 26—MCPS-SAP primitives

MCPS-SAP primitive	Request	Confirm	Indication
MCPS-DATA	7.1.1.1	7.1.1.2	7.1.1.3
MCPS-PURGE	7.1.1.4♦	7.1.1.5♦	—

All devices shall provide an interface for the MCPS-SAP primitives.

7.1.1.1 MCPS-DATA.request

The MCPS-DATA.request primitive requests the transfer of a data SPDU (i.e., MSDU) from a local SSCS entity to a single peer SSCS entity.

7.1.1.1.1 Semantics of the service primitive

The semantics of the MCPS-DATA.request primitive is as follows:

```
MCPS-DATA.request           (
                            SrcAddrMode,
                            SrcPANId,
                            SrcAddr,
                            DstAddrMode,
                            DstPANId,
                            DstAddr,
                            msduLength,
                            msdu,
                            msduHandle,
                            TxOptions
                            )
```

Table 27 specifies the parameters for the MCPS-DATA.request primitive.

Table 27—MCPS-DATA.request parameters

Name	Type	Valid range	Description
SrcAddrMode	Integer	0 x 00–0 x 03	The source addressing mode for this primitive and subsequent MPDU. This value can take one of the following values: 0 x 00 = no address (addressing fields omitted). 0 x 01 = reserved. 0 x 02 = 16 bit short address. 0 x 03 = 64 bit extended address.

Table 27—MCPS-DATA.request parameters *(continued)*

Name	Type	Valid range	Description
SrcPANId	Integer	0 x 000–0 x ffff	The 16 bit PAN identifier of the entity from which the MSDU is being transferred.
SrcAddr	Device address	As specified by the SrcAddrMode parameter	The individual device address of the entity from which the MSDU is being transferred.
DstAddrMode	Integer	0 x 00–0 x 03	The destination addressing mode for this primitive and subsequent MPDU. This value can take one of the following values: 0 x 00 = no address (addressing fields omitted). 0 x 01 = reserved. 0 x 02 = 16 bit short address. 0 x 03 = 64 bit extended address.
DstPANId	Integer	0 x 0000–0 x ffff	The 16 bit PAN identifier of the entity to which the MSDU is being transferred.
DstAddr	Device address	As specified by the DstAddrMode parameter	The individual device address of the entity to which the MSDU is being transferred.
msduLength	Integer	≤ aMaxMACFrameSize	The number of octets contained in the MSDU to be transmitted by the MAC sublayer entity.
msdu	Set of octets	—	The set of octets forming the MSDU to be transmitted by the MAC sublayer entity.
msduHandle	Integer	0 x 00–0 x ff	The handle associated with the MSDU to be transmitted by the MAC sublayer entity.
TxOptions	Bitmap	0000 xxxx (where x can be 0 or 1)	The transmission options for this MSDU. These are a bitwise OR of one or more of the following: 0 x 01 = acknowledged transmission. 0 x 02 = GTS transmission. 0 x 04 = indirect transmission. 0 x 08 = security enabled transmission.

7.1.1.1.2 When generated

The MCPS-DATA.request primitive is generated by a local SSCS entity when a data SPDU (i.e., MSDU) is to be transferred to a peer SSCS entity.

7.1.1.1.3 Effect on receipt

On receipt of the MCPS-DATA.request primitive, the MAC sublayer entity begins the transmission of the supplied MSDU.

The flags in the SrcAddrMode and DstAddrMode parameters correspond to the addressing subfields in the frame control field (see 7.2.1.1) and are used to construct both the frame control and addressing fields of the MHR.

The MAC sublayer builds an MPDU to transmit from the supplied arguments. The TxOptions parameter indicates how the MAC sublayer data service transmits the supplied MSDU. The indirect transmission bit is ignored if the GTS transmission bit is set to 1, the destination address is not present, or the MAC sublayer receiving this primitive is not the MAC sublayer of a coordinator.

If the TxOptions parameter specifies that a GTS transmission is required, the MAC sublayer will determine whether it has a valid GTS. If the device is a PAN coordinator, it will determine whether it has a receive GTS with the device with the given destination address. If a valid GTS could not be found, the MAC sublayer will issue the MCPS-DATA.confirm primitive with a status of INVALID_GTS. If a valid GTS was found, the MAC sublayer will defer, if necessary, until the GTS. At this time, the MAC sublayer transmits the MPDU without using CSMA-CA, provided that the entire transmission and acknowledgment, if requested, can be completed before the end of the GTS. If the TxOptions parameter specifies that a GTS transmission is not required, the MAC sublayer will transmit the MSDU using either slotted CSMA-CA in the CAP for a beacon-enabled PAN or unslotted CSMA-CA for a nonbeacon-enabled PAN. Specifying a GTS transmission in the TxOptions parameter overrides an indirect transmission request.

If the TxOptions parameter specifies that an indirect transmission is required and this primitive is received by the MAC sublayer of a coordinator, the information contained in the primitive will be added to the list of pending transactions stored on the coordinator. These transactions can then be extracted at the discretion of each device concerned using the method described in 7.5.6.3. If there is no capacity to store the transaction, the MAC sublayer will discard the MSDU and issue the MCPS-DATA.confirm primitive with a status of TRANSACTION_OVERFLOW. If there is capacity to store the transaction, the coordinator will add the information to the list. If the transaction is not handled within *macTransactionPersistenceTime*, the transaction information will be discarded and the MAC sublayer will issue the MCPS-DATA.confirm primitive with a status of TRANSACTION_EXPIRED. The transaction handling procedure is described in 7.5.5. If the TxOptions parameter specifies that an indirect transmission is required and either the device receiving this primitive is not a coordinator or the TxOptions parameter also specifies a GTS transmission, the indirect transmission option will be ignored. If the TxOptions parameter specifies that an indirect transmission is not required, the MAC sublayer will transmit the MSDU using CSMA-CA either in the CAP for a beacon-enabled PAN or immediately for a nonbeacon-enabled PAN.

If the TxOptions parameter specifies that security is not required for this frame, the MAC sublayer will set the security enabled subfield of the frame control field to 0 (see 7.2.1.1.2) and not perform any security operations on the frame. If the TxOptions parameter specifies that security is required for this frame, the MAC sublayer will set the security enabled subfield of the frame control field to 1 and obtain the key and security information, corresponding to the DstAddr parameter, from the ACL entries in the MAC PIB, as described in 7.5.8.4.1. If the DstAddrMode parameter indicates that the addressing fields are omitted, the MAC sublayer will obtain the key and security information corresponding to the address of the PAN coordinator. If an appropriate key could not be found in the ACL, the MAC sublayer will discard the frame and issue the MCPS-DATA.confirm primitive with a status of UNAVAILABLE_KEY. If an appropriate key was found in the ACL, the MAC sublayer will use it to apply security to the frame, according to the security information found in the ACL (see 7.5.8.4). If the length of the resulting frame is longer than *aMaxMACFrameSize*, the MAC sublayer will discard the frame and issue the MCPS-DATA.confirm primitive with a status of FRAME_TOO_LONG. If any other error occurs during the secure processing of the frame, the MAC sublayer will discard the frame and issue the MCPS-DATA.confirm primitive with a status of FAILED_SECURITY_CHECK.

If the requested transaction is too large to fit in the CAP or GTS, as appropriate, the MAC sublayer shall discard the frame and issue the MCPS-DATA.confirm primitive with a status of FRAME_TOO_LONG.

If the transmission uses CSMA-CA and the CSMA-CA algorithm failed due to adverse conditions on the channel, the MAC sublayer will discard the MSDU and issue the MCPS-DATA.confirm primitive with a status of CHANNEL_ACCESS_FAILURE.

To transmit the frame, the MAC sublayer first enables the transmitter by issuing the PLME-SET-TRX-STATE.request primitive with a state of TX_ON to the PHY. On receipt of the PLME-SET-TRX-STATE.confirm primitive with a status of either SUCCESS or TX_ON, the constructed MPDU is then transmitted by issuing the PD-DATA.request primitive. Finally, on receipt of the PD-DATA.confirm primitive, the MAC sublayer disables the transmitter by issuing the PLME-SET-TRX-STATE.request primitive with a state of RX_ON or TRX_OFF to the PHY, depending on whether the receiver is to be enabled following the transmission.

If the TxOptions parameter specifies that an acknowledged transmission is required, the MAC sublayer will enable its receiver immediately following the transmission of the MPDU and wait for an acknowledgment from the recipient for at most *macAckWaitDuration* symbols. If the MAC sublayer does not receive an acknowledgment within this time, it will retry its transmission at most *aMaxFrameRetries* times. If the MAC sublayer still does not receive an acknowledgment from the recipient, it will discard the MSDU and issue the MCPS-DATA.confirm primitive with a status of NO_ACK.

If the MPDU was successfully transmitted and an acknowledgment, if requested, was received, the MAC sublayer will issue the MCPS-DATA.confirm primitive with a status of SUCCESS.

If any parameter in the MCPS-DATA.request primitive is not supported or is out of range, the MAC sublayer will issue the MCPS-DATA.confirm primitive with a status of INVALID_PARAMETER.

7.1.1.2 MCPS-DATA.confirm

The MCPS-DATA.confirm primitive reports the results of a request to transfer a data SPDU (MSDU) from a local SSCS entity to a single peer SSCS entity.

7.1.1.2.1 Semantics of the service primitive

The semantics of the MCPS-DATA.confirm primitive is as follows:

```
MCPS-DATA.confirm            (
                             msduHandle,
                             status
                             )
```

Table 28 specifies the parameters for the MCPS-DATA.confirm primitive.

Table 28—MCPS-DATA.confirm parameters

Name	Type	Valid range	Description
msduHandle	Integer	0 x 00–0 x ff	The handle associated with the MSDU being confirmed.
status	Enumeration	SUCCESS, TRANSACTION_OVERFLOW, TRANSACTION_EXPIRED, CHANNEL_ACCESS_FAILURE, INVALID_GTS, NO_ACK, UNAVAILABLE_KEY, FRAME_TOO_LONG, FAILED_SECURITY_CHECK, or INVALID_PARAMETER	The status of the last MSDU transmission.

7.1.1.2.2 When generated

The MCPS-DATA.confirm primitive is generated by the MAC sublayer entity in response to an MCPS-DATA.request primitive. The MCPS-DATA.confirm primitive returns a status of either SUCCESS, indicating that the request to transmit was successful, or an error code of TRANSACTION_OVERFLOW, TRANSACTION_EXPIRED, CHANNEL_ACCESS_FAILURE, INVALID_GTS, UNAVAILABLE_KEY, NO_ACK, FRAME_TOO_LONG, FAILED_SECURITY_CHECK, or INVALID_PARAMETER. The reasons for these status values are fully described in 7.1.1.1.3.

7.1.1.2.3 Effect on receipt

On receipt of the MCPS-DATA.confirm primitive, the SSCS of the initiating device is notified of the result of its request to transmit. If the transmission attempt was successful, the status parameter will be set to SUCCESS. Otherwise, the status parameter will indicate the error.

7.1.1.3 MCPS-DATA.indication

The MCPS-DATA.indication primitive indicates the transfer of a data SPDU (i.e., MSDU) from the MAC sublayer to the local SSCS entity.

7.1.1.3.1 Semantics of the service primitive

The semantics of the MCPS-DATA.indication primitive is as follows:

```
MCPS-DATA.indication          (
                              SrcAddrMode,
                              SrcPANId,
                              SrcAddr,
                              DstAddrMode,
                              DstPANId
                              DstAddr,
                              msduLength,
                              msdu,
                              mpduLinkQuality,
                              SecurityUse,
                              ACLEntry
                              )
```

Table 29 specifies the parameters for the MCPS-DATA.indication primitive.

Table 29—MCPS-DATA.indication parameters

Name	Type	Valid range	Description
SrcAddrMode	Integer	0 x 00–0 x 03	The source addressing mode for this primitive corresponding to the received MPDU. This value can take one of the following values: 0 x 00 = no address (addressing fields omitted). 0 x 01 = reserved. 0 x 02 = 16 bit short address. 0 x 03 = 64 bit extended address.
SrcPANId	Integer	0 x 0000–0 x ffff	The 16 bit PAN identifier of the entity from which the MSDU was received.

Table 29—MCPS-DATA.indication parameters *(continued)*

Name	Type	Valid range	Description
SrcAddr	Device address	As specified by the SrcAddrMode parameter	The individual device address of the entity from which the MSDU was received.
DstAddrMode	Integer	0 x 00–0 x 03	The destination addressing mode for this primitive corresponding to the received MPDU. This value can take one of the following values: 0 x 00 = no address (addressing fields omitted). 0 x 01 = reserved. 0 x 02 = 16 bit short device address. 0 x 03 = 64 bit extended device address.
DstPANId	Integer	0 x 0000–0 x ffff	The 16 bit PAN identifier of the entity to which the MSDU is being transferred.
DstAddr	Device address	As specified by the DstAddrMode parameter	The individual device address of the entity to which the MSDU is being transferred.
msduLength	Integer	≤ *aMaxMACFrame-Size*	The number of octets contained in the MSDU being indicated by the MAC sublayer entity.
msdu	Set of octets	—	The set of octets forming the MSDU being indicated by the MAC sublayer entity.
mpduLinkQuality	Integer	0 x 00–0 x ff	LQ value measured during reception of the MPDU. Lower values represent lower LQ (see 6.7.8).
SecurityUse	Boolean	TRUE or FALSE	An indication of whether the received data frame is using security. This value is set to TRUE if the security enable subfield was set to 1 or FALSE if the security enabled subfield was set to 0.
ACLEntry	Integer	0 x 00–0 x 08	The *macSecurityMode* parameter value from the ACL entry associated with the sender of the data frame. This value is set to 0 x 08 if the sender of the data frame was not found in the ACL.

7.1.1.3.2 When generated

The MCPS-DATA.indication primitive is generated by the MAC sublayer and issued to the SSCS on receipt of a data frame at the local MAC sublayer entity that passes the appropriate message filtering operations as described in 7.5.6.2.

7.1.1.3.3 Effect on receipt

On receipt of the MCPS-DATA.indication primitive, the SSCS is notified of the arrival of data at the device.

7.1.1.4 MCPS-PURGE.request

The MCPS-PURGE.request primitive allows the next higher layer to purge an MSDU from the transaction queue.

7.1.1.4.1 Semantics of the service primitive

The semantics of the MCPS-PURGE.request primitive is as follows:

```
MCPS-PURGE.request              (
                                msduHandle
                                )
```

Table 30 specifies the parameters for the MCPS-PURGE.request primitive.

Table 30—MCPS-PURGE.request parameters

Name	Type	Valid range	Description
msduHandle	Integer	0 x 00–0 x ff	The handle of the MSDU to be purged from the transaction queue.

7.1.1.4.2 When generated

The MCPS-PURGE.request primitive is generated by the next higher layer whenever a MSDU is to be purged from the transaction queue.

7.1.1.4.3 Effect on receipt

On receipt of the MCPS-PURGE.request primitive, the MAC sublayer attempts to find in its transaction queue the MSDU indicated by the msduHandle parameter. If an MSDU matching the given handle is found, the MSDU is discarded from the transaction queue, and the MAC sublayer issues the MCPS-PURGE.confirm primitive with a status of SUCCESS. If an MSDU matching the given handle is not found, the MAC sublayer issues the MCPS-PURGE.confirm primitive with a status of INVALID_HANDLE.

7.1.1.5 MCPS-PURGE.confirm

The MCPS-PURGE.confirm primitive allows the MAC sublayer to notify the next higher layer of the success of its request to purge an MSDU from the transaction queue.

7.1.1.5.1 Semantics of the service primitive

The semantics of the MCPS-PURGE.confirm primitive is as follows:

```
MCPS-PURGE.confirm              (
                                msduHandle,
                                status
                                )
```

Table 31 specifies the parameters for the MCPS-PURGE.confirm primitive.

Table 31—MCPS-PURGE.confirm parameters

Name	Type	Valid range	Description
msduHandle	Integer	0 x 00–0 x ff	The handle of the MSDU requested to be purge from the transaction queue.
status	Enumeration	SUCCESS or INVALID_HANDLE	The status of the request to be purged an MSDU from the transaction queue.

7.1.1.5.2 When generated

The MCPS-PURGE.confirm primitive is generated by the MAC sublayer entity in response to an MCPS-PURGE.request primitive. The MCPS-PURGE.confirm primitive returns a status of either SUCCESS, indicating that the purge request was successful, or INVALID_HANDLE, indicating an error. The reasons for these status values are fully described in 7.1.1.4.3.

7.1.1.5.3 Effect on receipt

On receipt of the MCPS-PURGE.confirm primitive, the next higher layer is notified of the result of its request to purge an MSDU from the transaction queue. If the purge request was successful, the status parameter will be set to SUCCESS. Otherwise, the status parameter will indicate the error.

7.1.1.6 Data service message sequence charts

Figure 24 illustrates the sequence of messages necessary for a successful data transfer between two devices. Figure 81 and Figure 82 (see 7.7) illustrate this scenario, including the steps taken by the PHY.

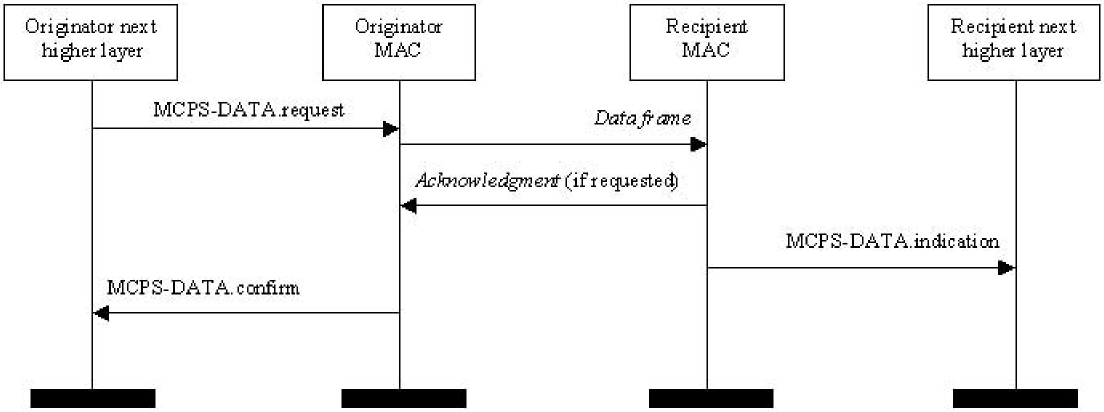


Figure 24—Message sequence chart describing the MAC data service

7.1.2 MAC management service

The MLME-SAP allows the transport of management commands between the next higher layer and the MLME. Table 32 summarizes the primitives supported by the MLME through the MLME-SAP interface.

Primitives marked with a diamond (♦) are optional for an RFD. The primitives are discussed in the subclauses referenced in the table.

Table 32—Summary of the primitives accessed through the MLME-SAP

Name	Request	Indication	Response	Confirm
MLME-ASSOCIATE	7.1.3.1	7.1.3.2♦	7.1.3.3♦	7.1.3.4
MLME-DISASSOCIATE	7.1.4.1	7.1.4.2		7.1.4.3
MLME-BEACON-NOTIFY		7.1.5.1		
MLME-GET	7.1.6.1			7.1.6.2
MLME-GTS	7.1.7.1♦	7.1.7.3♦		7.1.7.2♦
MLME-ORPHAN		7.1.8.1♦	7.1.8.2♦	
MLME-RESET	7.1.9.1			7.1.9.2
MLME-RX-ENABLE	7.1.10.1			7.1.10.2
MLME-SCAN	7.1.11.1			7.1.11.2
MLME-COMM-STATUS		7.1.12.1		
MLME-SET	7.1.13.1			7.1.13.2
MLME-START	7.1.14.1♦			7.1.14.2♦
MLME-SYNC	7.1.15.1			
MLME-SYNC-LOSS		7.1.15.2		
MLME-POLL	7.1.16.1			7.1.16.2

7.1.3 Association primitives

MLME-SAP association primitives define how a device becomes associated with a PAN.

All devices shall provide an interface for the request and confirm association primitives. The indication and response association primitives are optional for an RFD.

7.1.3.1 MLME-ASSOCIATE.request

The MLME-ASSOCIATE.request primitive allows a device to request an association with a coordinator.

7.1.3.1.1 Semantics of the service primitive

The semantics of the MLME-ASSOCIATE.request primitive is as follows:

```
MLME-ASSOCIATE.request          (
                                LogicalChannel,
                                CoordAddrMode,
                                CoordPANId,
                                CoordAddress,
                                CapabilityInformation,
                                SecurityEnable
                                )
```

Table 33 specifies the parameters for the MLME-ASSOCIATE.request primitive.

Table 33—MLME-ASSOCIATE.request parameters

Name	Type	Valid range	Description
LogicalChannel	Integer	Selected from the available channels supported by the PHY	The logical channel on which to attempt association.
CoordAddrMode	Integer	0 x 02–0 x 03	The coordinator addressing mode for this primitive and subsequent MPDU. This value can take one of the following values: 2=16 bit short address. 3=64 bit extended address.
CoordPANId	Integer	0 x 0000–0 x ffff	The identifier of the PAN with which to associate.
CoordAddress	Device address	As specified by the CoordAddrMode parameter.	The address of the coordinator with which to associate.
CapabilityInformation	Bitmap	See 7.3.1.1.2	Specifies the operational capabilities of the associating device.
SecurityEnable	Boolean	TRUE or FALSE	TRUE if security is enabled for this transfer or FALSE otherwise.

7.1.3.1.2 When generated

The MLME-ASSOCIATE.request primitive is generated by the next higher layer of an unassociated device and issued to its MLME to request an association with a coordinator. If the device wishes to associate with a coordinator on a beacon-enabled PAN, the MLME may optionally track the beacon of that coordinator prior to issuing this primitive.

7.1.3.1.3 Effect on receipt

On receipt of the MLME-ASSOCIATE.request primitive, the MLME of an unassociated device first updates *phyCurrentChannel* with the value of the LogicalChannel parameter by issuing the PLME-SET.request primitive and then updates *macPANId* with the value of the CoordPANId parameter. The MLME then generates an association request command (see 7.3.1.1) and sends it to the coordinator with the specified PAN identifier and address.

The SecurityEnable parameter specifies whether security is to be applied to the association request command frame. Typically, the association request command should not be implemented using security. However, if the device requesting association does have knowledge of the security information of the coordinator, then security may be specified in this case. If the SecurityEnable parameter is set to FALSE, the MLME will set the security enabled subfield of the frame control field to 0 (see 7.2.1.1.2) and not perform any security operations on the frame. If the SecurityEnable parameter is set to TRUE, the MLME will set the security enabled subfield of the frame control field to 1 and obtain the key and security information, corresponding to the CoordinatorAddress parameter, from the ACL entries in the MAC PIB, as described in 7.5.8.4.1. If an appropriate key could not be found in the ACL, the MLME will discard the frame and issue the MLME-ASSOCIATE.confirm primitive with a status of UNAVAILABLE_KEY. If an appropriate key was found in the ACL, the MLME will use it to apply security to the frame, according to the security information found in the ACL (see 7.5.8.4). If any other error occurs during the secure processing of the frame, the MLME will discard the frame and issue the MLME-ASSOCIATE.confirm primitive with a status of FAILED_SECURITY_CHECK.

If the association request command cannot be sent to the coordinator due to the CSMA algorithm indicating a busy channel, the MLME will issue the MLME-ASSOCIATE.confirm primitive with a status of CHANNEL_ACCESS_FAILURE.

To transmit the association request frame, the MLME first enables the transmitter by issuing the PLME-SET-TRX-STATE.request primitive with a state of TX_ON to the PHY. On receipt of the PLME-SET-TRX-STATE.confirm primitive with a status of either SUCCESS or TX_ON, the association request command frame is then transmitted by issuing the PD-DATA.request primitive. Finally, on receipt of the PD-DATA.confirm primitive, the MLME enables the receiver by issuing the PLME-SET-TRX-STATE.request primitive with a state of RX_ON to the PHY in preparation for the acknowledgment.

If the MLME successfully transmits an association request command, the MLME will expect an acknowledgment in return. If this does not occur, the association request command frame will be retried. If an acknowledgment is not received after *aMaxFrameRetries* attempts, the MLME will issue the MLME-ASSOCIATE.confirm primitive with a status of NO_ACK.

If the MLME of an unassociated device successfully receives an acknowledgment to its association request command, the MLME will wait for at most *aResponseWaitTime* symbols for the association response command to become available. If the MLME of the device does not extract an association response command frame from the coordinator within this time, it will issue the MLME-ASSOCIATE.confirm primitive with a status of NO_DATA.

If the MLME of the device extracts an association response command frame from the coordinator, it will then issue the MLME-ASSOCIATE.confirm primitive with a status equal to the contents of the association status field in the association response command (see 7.3.1.2.3).

On receipt of the association request command, the MLME of the coordinator issues the MLME-ASSOCIATE.indication primitive.

If any parameter in the MLME-ASSOCIATE.request primitive is either not supported or out of range, the MLME will issue the MLME-ASSOCIATE.confirm primitive with a status of INVALID_PARAMETER.

7.1.3.2 MLME-ASSOCIATE.indication

The MLME-ASSOCIATE.indication primitive is used to indicate the reception of an association request command.

7.1.3.2.1 Semantics of the service primitive

The semantics of the MLME-ASSOCIATE.indication primitive is as follows:

```
MLME-ASSOCIATE.indication          (
                                   DeviceAddress,
                                   CapabilityInformation,
                                   SecurityUse,
                                   ACLEntry
                                   )
```

Table 34 specifies the parameters for the MLME-ASSOCIATE.indication primitive.

Table 34—MLME-ASSOCIATE.indication parameters

Name	Type	Valid range	Description
DeviceAddress	Device address	An extended 64 bit IEEE address.	The address of the device requesting association.
CapabilityInformation	Bitmap	See 7.3.1.1.2	The operational capabilities of the device requesting association.
SecurityUse	Boolean	TRUE or FALSE	An indication of whether the received MAC command frame is using security. This value is set to TRUE if the security enable subfield was set to 1 or FALSE if the security enabled subfield was set to 0.
ACLEntry	Integer	0 x 00– 0 x 08	The *macSecurityMode* parameter value from the ACL entry associated with the sender of the data frame. This value is set to 0 x 08 if the sender of the data frame was not found in the ACL.

7.1.3.2.2 When generated

The MLME-ASSOCIATE.indication primitive is generated by the MLME of the coordinator and issued to its next higher layer to indicate the reception of an association request command (see 7.3.1.1).

7.1.3.2.3 Effect on receipt

When the next higher layer of a coordinator receives the MLME-ASSOCIATE.indication primitive, the coordinator determines whether to accept or reject the unassociated device using an algorithm outside the scope of this standard. The next higher layer of the coordinator then issues the MLME-ASSOCIATE.response primitive to its MLME.

The association decision and the response should become available at the coordinator within a time of *aResponseWaitTime*. After this time, the device requesting association attempts to extract the association response command frame from the coordinator, using the method described in 7.5.6.3, in order to determine whether the association was successful.

7.1.3.3 MLME-ASSOCIATE.response

The MLME-ASSOCIATE.response primitive is used to initiate a response to an MLME-ASSOCIATE.indication primitive.

7.1.3.3.1 Semantics of the service primitive

The semantics of the MLME-ASSOCIATE.response primitive is as follows:

```
MLME-ASSOCIATE.response        (
                               DeviceAddress,
                               AssocShortAddress,
                               status,
                               SecurityEnable
                               )
```

Table 35 specifies the parameters for the MLME-ASSOCIATE.response primitive.

Table 35—MLME-ASSOCIATE.response parameters

Name	Type	Valid range	Description
DeviceAddress	Device address	An extended 64 bit IEEE address	The address of the device requesting association.
AssocShortAddress	Integer	0 x 0000–0 x ffff	The short device address allocated by the coordinator on successful association. This parameter is set to 0xffff if the association was unsuccessful.
status	Enumeration	See 7.3.1.2.3	The status of the association attempt.
SecurityEnable	Boolean	TRUE or FALSE	TRUE if security is enabled for this transfer or FALSE otherwise.

7.1.3.3.2 When generated

The MLME-ASSOCIATE.response primitive is generated by the next higher layer of a coordinator and issued to its MLME in order to respond to the MLME-ASSOCIATE.indication primitive.

7.1.3.3.3 Effect on receipt

When the MLME of a coordinator receives the MLME-ASSOCIATE.response primitive, it generates an association response command (see 7.3.1.2). The command is sent to the device requesting association using indirect transmission, i.e., the command frame is added to the list of pending transactions stored on the coordinator and extracted at the discretion of the device concerned using the method described in 7.5.6.3.

The SecurityEnable parameter specifies whether security is to be applied to the association response command frame. If the SecurityEnable parameter is set to FALSE, the MLME will set the security enabled subfield of the frame control field to 0 (see 7.2.1.1.2) and not perform any security operations on the frame. If the

SecurityEnable parameter is set to TRUE, the MLME will set the security enabled subfield of the frame control field to 1 and obtain the key and security information, corresponding to the DeviceAddress parameter, from the ACL entries in the MAC PIB, as described in 7.5.8.4.1. If an appropriate key could not be found in the ACL, the MLME will discard the frame and issue the MLME-COMM-STATUS.indication primitive with a status of UNAVAILABLE_KEY. If an appropriate key was found in the ACL, the MLME will use it to apply security to the frame, according to the security information found in the ACL (see 7.5.8.4). If any other error occurs during the secure processing of the frame, the MLME will discard the frame and issue the MLME-COMM-STATUS.indication primitive with a status of FAILED_SECURITY_CHECK.

Upon receipt of the MLME-ASSOCIATE.response primitive, the coordinator attempts to add the information contained in the primitive to its list of pending transactions. If there is no capacity to store the transaction, the MAC sublayer will discard the MSDU and issue the MLME-COMM-STATUS.indication primitive with a status of TRANSACTION_OVERFLOW. If there is capacity to store the transaction, the coordinator will add the information to the list. If the transaction is not handled within *macTransactionPersistenceTime*, the transaction information will be discarded and the MAC sublayer will issue the MLME-COMM-STATUS.indication primitive with a status of TRANSACTION_EXPIRED. The transaction handling procedure is described in 7.5.5.

If the CSMA-CA algorithm failed due to adverse conditions on the channel, the MAC sublayer will discard the MSDU and issue the MLME-COMM-STATUS.indication primitive with a status of CHANNEL_ACCESS_FAILURE.

To transmit the association response command frame, the MAC sublayer first enables the transmitter by issuing the PLME-SET-TRX-STATE.request primitive with a state of TX_ON to the PHY. On receipt of the PLME-SET-TRX-STATE.confirm primitive with a status of either SUCCESS or TX_ON, the constructed MPDU is then transmitted by issuing the PD-DATA.request primitive. Finally, on receipt of the PD-DATA.confirm primitive, the MAC sublayer disables the transmitter by issuing the PLME-SET-TRX-STATE.request primitive with a state of RX_ON or TRX_OFF to the PHY, depending on whether the receiver is to be enabled following the transmission.

The MAC sublayer enables its receiver immediately following the transmission of the MPDU and waits for an acknowledgment from the recipient for at most *macAckWaitDuration* symbols. If the MAC sublayer does not receive an acknowledgment within this time, it will retry its transmission at most *aMaxFrameRetries* times. If the MAC sublayer still does not receive an acknowledgment from the recipient, it will discard the MSDU and issue the MLME-COMM-STATUS.indication primitive with a status of NO_ACK.

If the MPDU was successfully transmitted and an acknowledgment was received, if requested, the MAC sublayer will issue the MLME-COMM-STATUS.indication primitive with a status of SUCCESS.

If any parameter in the MLME-ASSOCIATE.response primitive is not supported or is out of range, the MAC sublayer will issue the MLME-COMM-STATUS.indication primitive with a status of INVALID_PARAMETER.

7.1.3.4 MLME-ASSOCIATE.confirm

The MLME-ASSOCIATE.confirm primitive is used to inform the next higher layer of the initiating device whether its request to associate was successful or unsuccessful.

7.1.3.4.1 Semantics of the service primitive

The semantics of the MLME-ASSOCIATE.confirm primitive is as follows:

```
MLME-ASSOCIATE.confirm          (
                                AssocShortAddress,
                                status
                                )
```

Table 36 specifies the parameters for the MLME-ASSOCIATE.confirm primitive.

Table 36—MLME-ASSOCIATE.confirm parameters

Name	Type	Valid range	Description
AssocShortAddress	Integer	0 x 0000–0 x ffff	The short device address allocated by the coordinator on successful association. This parameter will be equal to 0 x ffff if the association attempt was unsuccessful.
status	Enumeration	The value of the status field of the associate response command (see 7.3.1.2.3), SUCCESS, CHANNEL_ACCESS_FAILURE, NO_ACK, NO_DATA, UNAVAILABLE_KEY, FAILED_SECURITY_CHECK, or INVALID_PARAMETER.	The status of the association attempt.

7.1.3.4.2 When generated

The MLME-ASSOCIATE.confirm primitive is generated by the initiating MLME and issued to its next higher layer in response to an MLME-ASSOCIATE.request primitive. If the request was successful, the status parameter will indicate a successful association, as contained in the status field of the association response command. Otherwise, the status parameter indicates either an error code from the received association response command or an error code of CHANNEL_ACCESS_FAILURE, NO_ACK, NO_DATA, UNAVAILABLE_KEY, FAILED_SECURITY_CHECK, or INVALID_PARAMETER. The reasons for these status values are fully described in 7.1.3.1.3.

7.1.3.4.3 Effect on receipt

On receipt of the MLME-ASSOCIATE.confirm primitive, the next higher layer of the initiating device is notified of the result of its request to associate with a coordinator. If the association attempt was successful, the status parameter will indicate a successful association, as contained in the status field of the associate response command; and the device will be provided with a short address. If this short address is in the range of 0x0000 to 0xfffd, it may be used for communication in the PAN. If the short address is equal to 0xfffe, the device will use its extended 64 bit address for communication in the PAN. If the association attempt was unsuccessful, the address will be equal to 0 x ffff, and the status parameter will indicate the error.

7.1.3.5 Association message sequence charts

Figure 25 illustrates the sequence of messages necessary for a device to successfully associate with a PAN. Figure 78 and Figure 79 (see 7.7) illustrate the sequence of messages necessary for a device to associate with a coordinator and for a coordinator to allow association by a device, respectively; these figures include steps taken by the PHY.

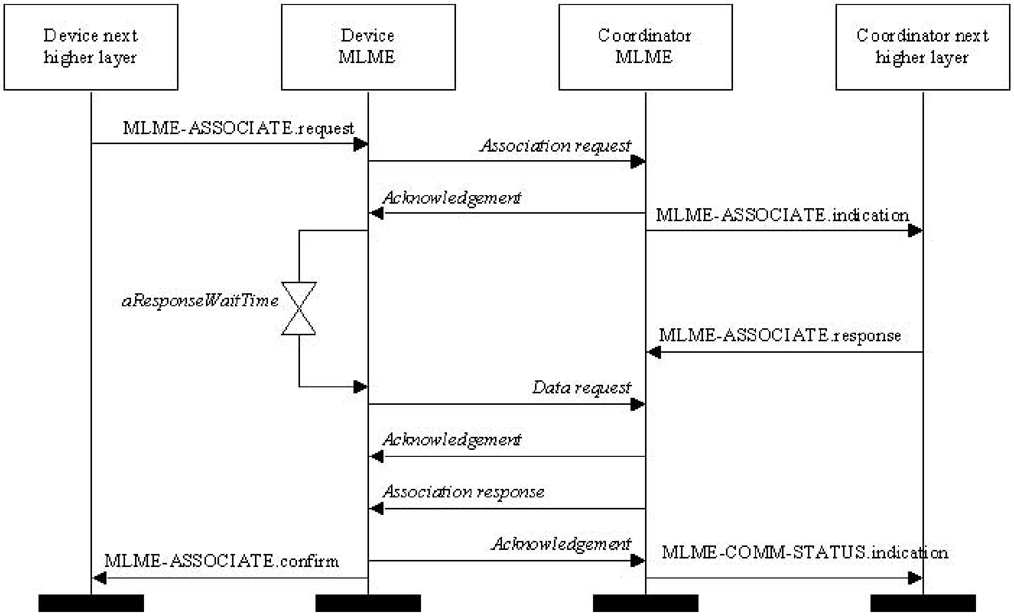


Figure 25—Message sequence chart for association

7.1.4 Disassociation primitives

The MLME-SAP disassociation primitives define how a device can disassociate from a PAN.

All devices shall provide an interface for these disassociation primitives.

7.1.4.1 MLME-DISASSOCIATE.request

The MLME-DISASSOCIATE.request primitive is used by an associated device to notify the coordinator of its intent to leave the PAN. It is also used by the coordinator to instruct an associated device to leave the PAN.

7.1.4.1.1 Semantics of the service primitive

The semantics of the MLME-DISASSOCIATE.request primitive is as follows:

```
MLME-DISASSOCIATE.request      (
                               DeviceAddress,
                               DisassociateReason,
                               SecurityEnable
                               )
```

Table 37 specifies the parameters for the MLME-DISASSOCIATE.request primitive.

Table 37—MLME-DISASSOCIATE.request parameters

Name	Type	Valid range	Description
DeviceAddress	Device address	An extended 64 bit IEEE address.	The address of the device to which to send the disassociation notification command.
DisassociateReason	Integer	0 x 00–0 x ff	The reason for the disassociation (see 7.3.1.3.2).
SecurityEnable	Boolean	TRUE or FALSE	TRUE if security is enabled for this transfer or FALSE otherwise.

7.1.4.1.2 When generated

The MLME-DISASSOCIATE.request primitive is generated by the next higher layer of an associated device and issued to its MLME to request disassociation from the PAN. It is also generated by the next higher layer of the coordinator and issued to its MLME to instruct an associated device to leave the PAN.

7.1.4.1.3 Effect on receipt

On receipt of the MLME-DISASSOCIATE.request primitive, the MLME generates a disassociation notification command (see 7.3.1.3). If the DeviceAddress parameter is equal to *macCoordExtendedAddress*, the command will be sent to its coordinator in the CAP for a beacon-enabled PAN or immediately for a nonbeacon-enabled PAN. If the DeviceAddress parameter is not equal to *macCoordExtendedAddress* and this primitive was received by the MLME of a coordinator, the command will be sent using indirect transmission, i.e., the command frame is added to the list of pending transactions stored on the coordinator and extracted at the discretion of the concerned device using the method described in 7.5.6.3. Otherwise, the MLME issues the MLME-DISASSOCIATE.confirm primitive with a status of INVALID_PARAMETER.

If the disassociation notification command is to be sent using indirect transmission by the coordinator, the device address will be added to the address list field of the beacon, indicating a pending message. In this case, the coordinator attempts to add the information contained in the primitive to its list of pending transactions. If there is no capacity to store the transaction, the MLME will discard the MSDU and issue the MLME-DISASSOCIATE.confirm primitive with a status of TRANSACTION_OVERFLOW. If there is capacity to store the transaction, the coordinator will add the information to the list. If the transaction is not handled within *macTransactionPersistenceTime*, the transaction information will be discarded and the MLME will issue the MLME-DISASSOCIATE.confirm with a status of TRANSACTION_EXPIRED. The transaction handling procedure is described in 7.5.5.

The SecurityEnable parameter specifies whether security is to be applied to the disassociation notification command frame. If the SecurityEnable parameter is set to FALSE, the MLME will set the security enabled subfield of the frame control field to 0 (see 7.2.1.1.2) and not perform any security operations on the frame. If the SecurityEnable parameter is set to TRUE, the MLME will set the security enabled subfield of the frame control field to 1 and obtain the key and security information, corresponding to the DeviceAddress parameter, from the ACL entries in the MAC PIB, as described in 7.5.8.4.1. If an appropriate key could not be found in the ACL, the MLME will discard the frame and issue the MLME-DISASSOCIATE.confirm primitive with a status of UNAVAILABLE_KEY. If an appropriate key was found in the ACL, the MLME will use it to apply security to the frame, according to the security information found in the ACL (see 7.5.8.4). If any other error occurs during the secure processing of the frame, the MLME will discard the frame and issue the MLME-DISASSOCIATE.confirm primitive with a status of FAILED_SECURITY_CHECK.

If the disassociation notification command cannot be sent due to a CSMA algorithm failure, the MLME will issue the MLME-DISASSOCIATE.confirm primitive with a status of CHANNEL_ACCESS_FAILURE.

If the disassociation notification command frame is to be transmitted by a device, the MLME first enables the transmitter by issuing the PLME-SET-TRX-STATE.request primitive with a state of TX_ON to the PHY. On receipt of the PLME-SET-TRX-STATE.confirm primitive with a status of either SUCCESS or TX_ON, the disassociation notification command frame is then transmitted by issuing the PD-DATA.request primitive. Finally, on receipt of the PD-DATA.confirm primitive, the MLME enables the receiver by issuing the PLME-SET-TRX-STATE.request primitive with a state of RX_ON to the PHY in preparation for the acknowledgment.

If the MLME successfully transmits a disassociation notification command, the MLME will expect an acknowledgment in return. If this does not occur, the disassociation notification command frame will be retried. If an acknowledgment is not received after *aMaxFrameRetries* attempts, the MLME will issue the MLME-DISASSOCIATE.confirm primitive with a status of NO_ACK.

If the MLME successfully transmits a disassociation notification command and receives an acknowledgment in return, the MLME will issue the MLME-DISASSOCIATE.confirm primitive with a status of SUCCESS.

In the case where a device MLME receives this primitive from its next higher layer, the disassociation notification command is sent to the coordinator specified in the DeviceAddress parameter. Similarly, in the case where a coordinator MLME receives this primitive from its next higher layer, the disassociation notification command is sent to the device specified in the DeviceAddress parameter.

On receipt of the disassociation notification command, the MLME of the recipient issues the MLME-DISASSOCIATE.indication primitive.

If any parameter in the MLME-DISASSOCIATE.request primitive is not supported or is out of range, the MLME will issue the MLME-DISASSOCIATE.confirm primitive with a status of INVALID_PARAMETER.

7.1.4.2 MLME-DISASSOCIATE.indication

The MLME-DISASSOCIATE.indication primitive is used to indicate the reception of a disassociation notification command.

7.1.4.2.1 Semantics of the service primitive

The semantics of the MLME-DISASSOCIATE.indication primitive is as follows:

```
MLME-DISASSOCIATE.indication    (
                                DeviceAddress,
                                DisassociateReason,
                                SecurityUse,
                                ACLEntry
                                )
```

Table 38 specifies the parameters for the MLME-DISASSOCIATE.indication primitive.

7.1.4.2.2 When generated

The MLME-DISASSOCIATE.indication primitive is generated by the MLME and issued to its next higher layer on receipt of a disassociation notification command.

Table 38—MLME-DISASSOCIATE.indication parameters

Name	Type	Valid range	Description
DeviceAddress	Device address	An extended 64 bit IEEE address	The address of the device requesting disassociation.
DisassociateReason	Integer	0 x 00–0 x ff	The reason for the disassociation (see 7.3.1.3.2).
SecurityUse	Boolean	TRUE or FALSE	An indication of whether the received MAC command frame is using security. This value is set to TRUE if the security enable subfield was set to 1 or FALSE if the security enabled subfield was set to 0.
ACLEntry	Integer	0 x 00–0 x 08	The *macSecurityMode* parameter value from the ACL entry associated with the sender of the data frame. This value is set to 0x08 if the sender of the data frame was not found in the ACL.

7.1.4.2.3 Effect on receipt

The next higher layer is notified of the reason for the disassociation.

7.1.4.3 MLME-DISASSOCIATE.confirm

The MLME-DISASSOCIATE.confirm primitive reports the results of an MLME-DISASSOCIATE.request primitive.

7.1.4.3.1 Semantics of the service primitive

The semantics of the MLME-DISASSOCIATE.confirm primitive is as follows:

```
MLME-DISASSOCIATE.confirm        (
                                 status
                                 )
```

Table 39 specifies the parameters for the MLME-DISASSOCIATE.confirm primitive.

Table 39—MLME-DISASSOCIATE.confirm parameters

Name	Type	Valid range	Description
Status	Enumeration	SUCCESS, TRANSACTION_OVERFLOW, TRANSACTION_EXPIRED, NO_ACK, CHANNEL_ACCESS_FAILURE, UNAVAILABLE_KEY, FAILED_SECURITY_CHECK, or INVALID_PARAMETER	The status of the disassociation attempt.

7.1.4.3.2 When generated

The MLME-DISASSOCIATE.confirm primitive is generated by the initiating MLME and issued to its next higher layer in response to an MLME-DISASSOCIATE.request primitive. This primitive returns a status of either SUCCESS, indicating that the disassociation request was successful, or an error code of TRANSACTION_OVERFLOW, TRANSACTION_EXPIRED, NO_ACK, CHANNEL_ACCESS_ FAILURE, UNAVAILABLE_KEY, FAILED_SECURITY_CHECK, or INVALID_PARAMETER. The reasons for these status values are fully described in 7.1.4.1.3.

7.1.4.3.3 Effect on receipt

On receipt of the MLME-DISASSOCIATE.confirm primitive, the next higher layer of the initiating device is notified of the result of the disassociation attempt. If the disassociation attempt was successful, the status parameter will be set to SUCCESS. Otherwise, the status parameter indicates the error.

7.1.4.4 Disassociation message sequence charts

Figure 26 illustrates the sequence of messages necessary for successful disassociation from a PAN. The originating device may be either a device or the coordinator to which the device has associated.

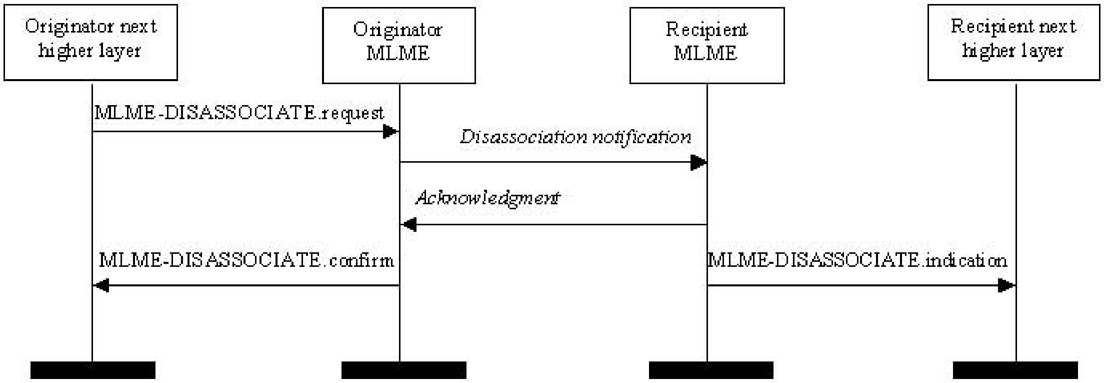

Originator next higher layer
Originator MLME
Recipient MLME
Recipient next higher layer
MLME-DISASSOCIATE.request
Disassociation notification
Acknowledgment
MLME-DISASSOCIATE.confirm
MLME-DISASSOCIATE.indication

Figure 26—Message sequence chart for disassociation

7.1.5 Beacon notification primitive

The MLME-SAP beacon notification primitive defines how a device may be notified when a beacon is received during normal operating conditions.

All devices shall provide an interface for the beacon notification primitive.

7.1.5.1 MLME-BEACON-NOTIFY.indication

The MLME-BEACON-NOTIFY.indication primitive is used to send parameters contained within a beacon frame received by the MAC sublayer to the next higher layer. The primitive also sends a measure of the LQ and the time the beacon frame was received.

7.1.5.1.1 Semantics of the service primitive

The semantics of the MLME-BEACON-NOTIFY.indication primitive is as follows:

```
MLME-BEACON-NOTIFY.indication    (
                                 BSN,
                                 PANDescriptor,
                                 PendAddrSpec,
                                 AddrList,
                                 sduLength,
                                 sdu
                                 )
```

Table 40 specifies the parameters for the MLME-BEACON-NOTIFY.indication primitive.

Table 40—MLME-BEACON-NOTIFY.indication parameters

Name	Type	Valid range	Description
BSN	Integer	0 x 00–0 x ff	The beacon sequence number.
PANDescriptor	PANDescriptor value	See Table 41	The PANDescriptor for the received beacon.
PendAddrSpec	Bitmap	See 7.2.2.1.6	The beacon pending address specification.
AddrList	List of device addresses	—	The list of addresses of the devices for which the beacon source has data.
sduLength	Integer	0 – *aMaxBeaconPayloadLength*	The number of octets contained in the beacon payload of the beacon frame received by the MAC sublayer.
sdu	Set of octets	—	The set of octets comprising the beacon payload to be transferred from the MAC sublayer entity to the next higher layer.

Table 41 describes the elements of the PANDescriptor type.

Table 41—Elements of PANDescriptor

Name	Type	Valid range	Description
CoordAddrMode	Integer	0 x 02–0 x 03	The coordinator addressing mode corresponding to the received beacon frame. This value can take one of the following values: 2 = 16 bit short address. 3 = 64 bit extended address.
CoordPANId	Integer	0 x 0000–0 x ffff	The PAN identifier of the coordinator as specified in the received beacon frame.

Table 41—Elements of PANDescriptor *(continued)*

Name	Type	Valid range	Description
CoordAddress	Device address	As specified by the CoordAddrMode parameter	The address of the coordinator as specified in the received beacon frame.
LogicalChannel	Integer	Selected from the available logical channels supported by the PHY	The current logical channel occupied by the network.
SuperframeSpec	Bitmap	See 7.2.2.1.2	The superframe specification as specified in the received beacon frame.
GTSPermit	Boolean	TRUE or FALSE	TRUE if the beacon is from a PAN coordinator that is accepting GTS requests.
LinkQuality	Integer	0 x 00–0 x ff	The LQ at which the network beacon was received. Lower values represent lower LQ (see 6.7.8).
TimeStamp	Integer	0 x 000000–0 x ffffff	The time at which the beacon frame was received, in symbols. This value is equal to the timestamp taken when the beacon frame was received, as described in 7.5.4.1. The precision of this value is a minimum of 20 bits, with the lowest 4 bits being the least significant.
SecurityUse	Boolean	TRUE or FALSE	An indication of whether the received beacon frame is using security. This value is set to TRUE if the security enable subfield was set to 1 or FALSE if the security enabled subfield was set to 0.
ACLEntry	Integer	0 x 00–0 x 08	The *macSecurityMode* parameter value from the ACL entry associated with the sender of the data frame. This value is set to 0 x 08 if the sender of the data frame was not found in the ACL.
SecurityFailure	Boolean	TRUE or FALSE	TRUE if there was an error in the security processing of the frame or FALSE otherwise.

7.1.5.1.2 When generated

The MLME-BEACON-NOTIFY.indication primitive is generated by the MLME and issued to its next higher layer upon receipt of a beacon frame either when *macAutoRequest* is set to FALSE or when the beacon frame contains one or more octets of payload.

7.1.5.1.3 Effect on receipt

On receipt of the MLME-BEACON-NOTIFY.indication primitive, the next higher layer is notified of the arrival of a beacon frame at the MAC sublayer.

7.1.6 Primitives for reading PIB attributes

The MLME-SAP get primitives define how to read values from the PIB.

All devices shall provide an interface for these get primitives.

7.1.6.1 MLME-GET.request

The MLME-GET.request primitive requests information about a given PIB attribute.

7.1.6.1.1 Semantics of the service primitive

The semantics of the MLME-GET.request primitive is as follows:

```
MLME-GET.request                  (
                                  PIBAttribute
                                  )
```

Table 42 specifies the parameters for the MLME-GET.request primitive.

Table 42—MLME-GET.request parameters

Name	Type	Valid range	Description
PIBAttribute	Integer	See Table 71 and Table 72	The identifier of the PIB attribute to read.

7.1.6.1.2 When generated

The MLME-GET.request primitive is generated by the next higher layer and issued to its MLME to obtain information from the MAC PIB.

7.1.6.1.3 Effect on receipt

On receipt of the MLME-GET.request primitive, the MLME attempts to retrieve the requested MAC PIB attribute from its database. If the identifier of the PIB attribute is not found in the database, the MLME will issue the MLME-GET.confirm primitive with a status of UNSUPPORTED_ATTRIBUTE.

If the requested MAC PIB attribute is successfully retrieved, the MLME will issue the MLME-GET.confirm primitive with a status of SUCCESS.

7.1.6.2 MLME-GET.confirm

The MLME-GET.confirm primitive reports the results of an information request from the MAC PIB.

7.1.6.2.1 Semantics of the service primitive

The semantics of the MLME-GET.confirm primitive is as follows:

```
MLME-GET.confirm                  (
                                  status,
                                  PIBAttribute,
                                  PIBAttributeValue
                                  )
```

Table 43 specifies the parameters for the MLME-GET.confirm primitive.

Table 43—MLME-GET.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS or UNSUPPORTED_ATTRIBUTE	The result of the request for MAC PIB attribute information.
PIBAttribute	Integer	See Table 71 and Table 72	The identifier of the MAC PIB attribute that was read.
PIBAttributeValue	Various	Attribute specific; see Table 71 and Table 72	The value of the indicated MAC PIB attribute that was read.

7.1.6.2.2 When generated

The MLME-GET.confirm primitive is generated by the MLME and issued to its next higher layer in response to an MLME-GET.request primitive. This primitive returns a status of either SUCCESS, indicating that the request to read a MAC PIB attribute was successful, or an error code of UNSUPPORTED_ATTRIBUTE. The reasons for these status values are fully described in 7.1.6.1.3.

7.1.6.2.3 Effect on receipt

On receipt of the MLME-GET.confirm primitive, the next higher layer is notified of the results of its request to read a MAC PIB attribute. If the request to read a MAC PIB attribute was successful, the status parameter will be set to SUCCESS. Otherwise, the status parameter indicates the error.

7.1.7 GTS management primitives

The MLME-SAP GTS management primitives define how GTSs are requested and maintained. Devices wishing to use these primitives and GTSs in general will already be tracking the beacons of their PAN coordinators.

These GTS management primitives are optional for an RFD.

7.1.7.1 MLME-GTS.request

The MLME-GTS.request primitive allows a device to send a request to the PAN coordinator to allocate a new GTS or to deallocate an existing GTS.

7.1.7.1.1 Semantics of the service primitive

The semantics of the MLME-GTS.request primitive is as follows:

```
MLME-GTS.request            (
                            GTSCharacteristics,
                            SecurityEnable
                            )
```

Table 44 specifies the parameters for the MLME-GTS.request primitive.

Table 44—MLME-GTS.request parameters

Name	Type	Valid range	Description
GTSCharacteristics	GTS characteristics	See 7.3.3.1.2	The characteristics of the GTS request.
SecurityEnable	Boolean	TRUE or FALSE	TRUE if security is enabled for this transfer or FALSE otherwise.

7.1.7.1.2 When generated

The MLME-GTS.request primitive is generated by the next higher layer and issued to its MLME to request the allocation of a new GTS or to request the deallocation of an existing GTS.

The GTSCharacteristics parameter specifies whether the request is for the allocation of a new GTS or for the deallocation of an existing GTS. If the characteristics type field of the GTSCharacteristics parameter is equal to 1, the remaining fields in the GTSCharacteristics will specify the desired characteristics of the new GTS, i.e., its length and direction. If the characteristics type field of the GTSCharacteristics parameter is equal to 0, the remaining fields in the GTSCharacteristics will specify the length and direction of the GTS that the device wishes to deallocate.

7.1.7.1.3 Effect on receipt

On receipt of the MLME-GTS.request primitive, the MLME of a device attempts to generate a GTS request command (see 7.3.3.1) with the information contained in this primitive and, if successful, sends it to the PAN coordinator.

If *macShortAddress* is equal to 0 x fffe or 0 x ffff, the device is not permitted to request a GTS. In this case, the MLME issues the MLME-GTS.confirm primitive containing a status of NO_SHORT_ADDRESS.

The SecurityEnable parameter specifies whether security is to be applied to the GTS request command frame. If the SecurityEnable parameter is set to FALSE, the MLME will set the security enabled subfield of the frame control field to 0 (see 7.2.1.1.2) and not perform any security operations on the frame. If the SecurityEnable parameter is set to TRUE, the MLME will set the security enabled subfield of the frame control field to 1 and obtain the key and security information, corresponding to the address of the PAN coordinator, *macCoordExtendedAddress*, from the ACL entries in the MAC PIB, as described in 7.5.8.4.1. If an appropriate key could not be found in the ACL, the MLME will discard the frame and issue the MLME-GTS.confirm primitive with a status of UNAVAILABLE_KEY. If an appropriate key was found in the ACL, the MLME will use it to apply security to the frame, according to the security information found in the ACL (see 7.5.8.4). If any other error occurs during the secure processing of the frame, the MLME will discard the frame and issue the MLME-GTS.confirm primitive with a status of FAILED_SECURITY_CHECK.

If the GTS request command cannot be sent due to a CSMA algorithm failure, the MLME will issue the MLME-GTS.confirm primitive with a status of CHANNEL_ACCESS_FAILURE.

To transmit the GTS request command frame, the MLME first enables the transmitter by issuing the PLME-SET-TRX-STATE.request primitive with a state of TX_ON to the PHY. On receipt of the PLME-SET-TRX-STATE.confirm primitive with a status of either SUCCESS or TX_ON, the GTS request command frame is then transmitted by issuing the PD-DATA.request primitive. Finally, on receipt of the PD-DATA.confirm primitive, the MLME enables the receiver by issuing the PLME-SET-TRX-STATE.request primitive with a state of RX_ON to the PHY in preparation for the acknowledgment.

If the MLME successfully transmits a GTS request command, the MLME will expect an acknowledgment in return. If this does not occur, the GTS request command frame will be retried. If an acknowledgment is not received after *aMaxFrameRetries* attempts, the MLME will issue the MLME-GTS.confirm primitive with a status of NO_ACK.

If a GTS is being allocated and the request has been acknowledged, the device will wait for a confirmation via a GTS descriptor specified in a beacon frame from its PAN coordinator. If the PAN coordinator can allocate the requested GTS, it will issue the MLME-GTS.indication primitive with the characteristics of the allocated GTS. It also generates a GTS descriptor with the characteristics of the allocated GTS and the short address of the requesting device. If the PAN coordinator cannot allocate the requested GTS, it will generate a GTS descriptor with a start slot of 0 and the short address of the requesting device. In either case, the descriptor exists in the beacon for *aGTSDescPersistenceTime* superframes.

If the device receives a beacon frame from its PAN coordinator with a GTS descriptor containing a short address that matches *macShortAddress* before *aGTSDescPersistenceTime* beacons have been received, the device will process the descriptor. If no descriptor for that device is received before *aGTSDescPersistenceTime* beacons have been received or an MLME-SYNC-LOSS.indication, with a loss reason of BEACON_LOST, is issued, the MLME will issue the MLME-GTS.confirm primitive with a status of NO_DATA.

If a descriptor appears that matches the characteristics requested, the device will assume that the GTS was successfully allocated. The MLME of the device requesting the GTS issues the MLME-GTS.confirm primitive with a status of SUCCESS and a GTSCharacteristics parameter with a characteristics type equal to 1. The device may now start using the GTS.

If the descriptor appears with a start slot of 0, the PAN coordinator has denied the request. In this case, the device requesting the GTS issues the MLME-GTS.confirm primitive with a status of DENIED, indicating that the GTSCharacteristics parameter is to be ignored.

If a GTS is being deallocated, the device will issue the MLME-GTS.confirm primitive with a status of SUCCESS and a GTSCharacteristics parameter with a characteristics type equal to 0. On receipt of a GTS request command with a request type indicating a GTS deallocation, the PAN coordinator acknowledges the frame and deallocates the GTS. The MLME of the PAN coordinator will then issue the MLME-GTS.indication primitive with the appropriate GTS characteristics. If the PAN coordinator does not receive the deallocation request, countermeasures can be applied by the coordinator to ensure consistency is maintained (see 7.5.7.6).

If any parameter in the MLME-GTS.request primitive is not supported or is out of range, the MLME will issue the MLME-GTS.confirm primitive with a status of INVALID_PARAMETER.

7.1.7.2 MLME-GTS.confirm

The MLME-GTS.confirm primitive reports the results of a request to allocate a new GTS or deallocate an existing GTS.

7.1.7.2.1 Semantics of the service primitive

The semantics of the MLME-GTS.confirm primitive is as follows:

```
MLME-GTS.confirm          (
                          GTSCharacteristics,
                          status
                          )
```

Table 45 specifies the parameters for the MLME-GTS.confirm primitive.

Table 45—MLME-GTS.confirm parameters

Name	Type	Valid range	Description
GTSCharacteristics	GTS characteristics	See 7.3.3.1.2	The characteristics of the GTS.
status	Enumeration	SUCCESS, DENIED, NO_SHORT_ADDRESS, CHANNEL_ACCESS_FAILURE, NO_ACK, NO_DATA, UNAVAILABLE_KEY, FAILED_SECURITY_CHECK, or INVALID_PARAMETER.	The status of the GTS request.

7.1.7.2.2 When generated

The MLME-GTS.confirm primitive is generated by the MLME and issued to its next higher layer in response to a previously issued MLME-GTS.request primitive.

If the request to allocate or deallocate a GTS was successful, this primitive will return a status of SUCCESS and the characteristics type field of the GTSCharacteristics parameter will have the value of 1 or 0, respectively. If the PAN coordinator denied the request, the primitive will return a status of DENIED. Otherwise, the status parameter indicates an error code of NO_SHORT_ADDRESS, CHANNEL_ACCESS_ FAILURE, NO_ACK, NO_DATA, UNAVAILABLE_KEY, FAILED_SECURITY_CHECK or INVALID_ PARAMETER. The reasons for these status values are fully described in 7.1.7.1.3.

7.1.7.2.3 Effect on receipt

On receipt of the MLME-GTS.confirm primitive the next higher layer is notified of the result of its request to allocate or deallocate a GTS. If the request was successful, the status parameter will indicate a successful GTS operation. Otherwise, the status parameter will indicate the error.

7.1.7.3 MLME-GTS.indication

The MLME-GTS.indication primitive indicates that a GTS has been allocated or that a previously allocated GTS has been deallocated.

7.1.7.3.1 Semantics of the service primitive

The semantics of the MLME-GTS.indication primitive is as follows:

```
MLME-GTS.indication          (
                             DevAddress,
                             GTSCharacteristics,
                             SecurityUse,
                             ACLEntry
                             )
```

Table 46 specifies the parameters for the MLME-GTS.indication primitive.

Table 46—MLME-GTS.indication parameters

Name	Type	Valid range	Description
DevAddress	Device address	0 x 0000–0 x fffd	The short address of the device that has been allocated or deallocated a GTS.
GTSCharacteristics	GTS characteristics	See 7.3.3.1.2	The characteristics of the GTS.
SecurityUse	Boolean	TRUE or FALSE	An indication of whether the received frame is using security. This value is set to TRUE if the security enable subfield was set to 1 or FALSE if the security enabled subfield was set to 0.
ACLEntry	Integer	0 x 00–0 x 08	The *macSecurityMode* parameter value from the ACL entry associated with the sender of the data frame. This value is set to 0x08 if the sender of the data frame was not found in the ACL.

7.1.7.3.2 When generated

The MLME-GTS.indication primitive is generated by the MLME of the PAN coordinator to its next higher layer whenever a GTS is allocated or deallocated following the reception of a GTS request command (see 7.3.3.1) by the MLME. The MLME of the PAN coordinator also generates this primitive when a GTS deallocation is initiated by the PAN coordinator itself. The characteristics type field in the GTSCharacteristics parameter will be equal to 1 if a GTS has been allocated or 0 if a GTS has been deallocated.

This primitive is generated by the MLME of a device and issued to its next higher layer when the PAN coordinator has deallocated one of its GTSs. In this case, the characteristics type field of the GTSCharacteristics parameter is equal to 0.

7.1.7.3.3 Effect on receipt

On receipt of the MLME-GTS.indication primitive the next higher layer is notified of the allocation or deallocation of a GTS.

7.1.7.4 GTS management message sequence charts

Figure 27 and Figure 28 illustrate the sequence of messages necessary for successful GTS management. The first depicts the message flow for the case in which the device initiates the GTS allocation. The second depicts the message flow for the two cases for which a GTS deallocation occurs, first, by a device (a) and, second, by the PAN coordinator (b).

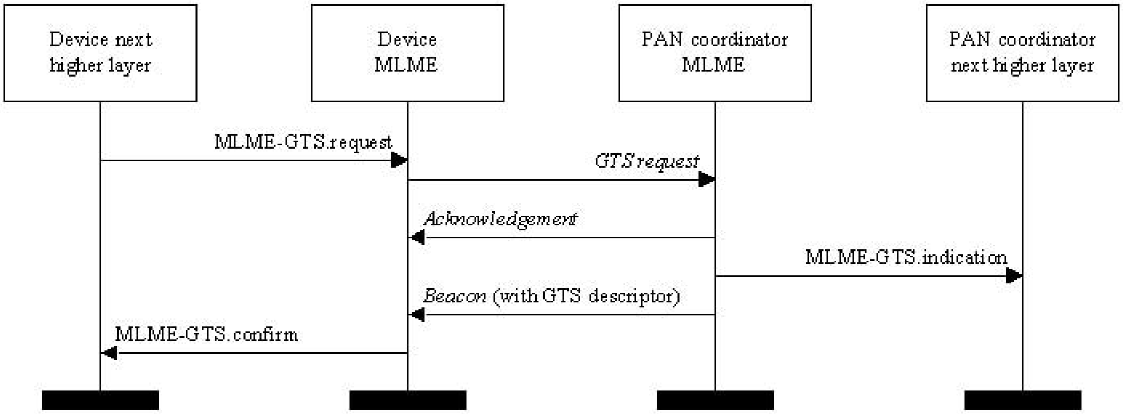


Figure 27—Message sequence chart for GTS allocation initiated by a device

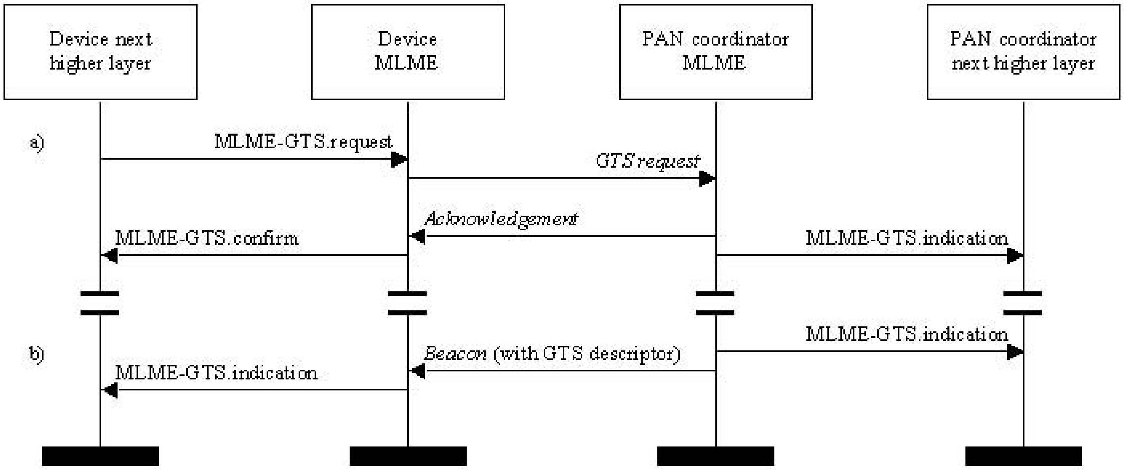


Figure 28—Message sequence chart for GTS deallocation initiated by a device (a) and the PAN coordinator (b)

7.1.8 Primitives for orphan notification

MLME-SAP orphan notification primitives define how a coordinator can issue a notification of an orphaned device.

These orphan notification primitives are optional for an RFD.

7.1.8.1 MLME-ORPHAN.indication

The MLME-ORPHAN.indication primitive allows the MLME of a coordinator to notify the next higher layer of the presence of an orphaned device.

7.1.8.1.1 Semantics of the service primitive

The semantics of the MLME-ORPHAN.indication primitive is as follows:

```
MLME-ORPHAN.indication        (
                              OrphanAddress,
                              SecurityUse,
                              ACLEntry
                              )
```

Table 47 specifies the parameters for the MLME-ORPHAN.indication primitive.

Table 47—MLME-ORPHAN.indication parameters

Name	Type	Valid range	Description
OrphanAddress	Device address	Extended 64 bit IEEE address	The address of the orphaned device.
SecurityUse	Boolean	TRUE or FALSE	An indication of whether the received MAC command frame is using security. This value is set to TRUE if the security enable subfield was set to 1 or FALSE if the security enabled subfield was set to 0.
ACLEntry	Integer	0 x 00–0 x 08	The *macSecurityMode* parameter value from the ACL entry associated with the sender of the data frame. This value is set to 0 x 08 if the sender of the data frame was not found in the ACL.

7.1.8.1.2 When generated

The MLME-ORPHAN.indication primitive is generated by the MLME of a coordinator and issued to its next higher layer on receipt of an orphan notification command (see 7.3.2.3).

7.1.8.1.3 Effect on receipt

The effect on receipt of the MLME-ORPHAN.indication primitive is that the next higher layer is notified of the orphaned device. The next higher layer then determines whether the device was previously associated and issues the MLME-ORPHAN.response primitive to the MLME with its decision.

The decision about whether the device was previously associated to the coordinator and the response occurs within a time of *aResponseWaitTime* symbols. If the device was previously associated with the coordinator, it will send the MLME-ORPHAN.response primitive with the AssociatedMember parameter set to TRUE and the ShortAddress parameter set to the corresponding short address allocated to the orphaned device. If the device was not previously associated with the coordinator, it will send the MLME-ORPHAN.response primitive with the AssociatedMember parameter set to FALSE.

7.1.8.2 MLME-ORPHAN.response

The MLME-ORPHAN.response primitive allows the next higher layer of a coordinator to respond to the MLME-ORPHAN.indication primitive.

7.1.8.2.1 Semantics of the service primitive

The semantics of the MLME-ORPHAN.response primitive is as follows:

```
MLME-ORPHAN.response          (
                              OrphanAddress,
                              ShortAddress,
                              AssociatedMember,
                              SecurityEnable
                              )
```

Table 48 specifies the parameters for the MLME-ORPHAN.response primitive.

Table 48—MLME-ORPHAN.response parameters

Name	Type	Valid range	Description
OrphanAddress	Device address	Extended 64 bit IEEE address	The address of the orphaned device.
ShortAddress	Integer	0 x 0000–0 x ffff	The short address allocated to the orphaned device if it is associated with this coordinator. The special short address 0 x fffe indicates that no short address was allocated, and the device will use its 64 bit extended address in all communications. If the device was not associated with this coordinator, this field will contain the value 0 x ffff and be ignored on receipt.
AssociatedMember	Boolean	TRUE or FALSE	TRUE if the orphaned device is associated with this coordinator or FALSE otherwise.
SecurityEnable	Boolean	TRUE or FALSE	TRUE if security is enabled for this transfer or FALSE otherwise.

7.1.8.2.2 When generated

The MLME-ORPHAN.response primitive is generated by the next higher layer and issued to its MLME when it reaches a decision about whether the orphaned device indicated in the MLME-ORPHAN.indication primitive is associated.

7.1.8.2.3 Effect on receipt

If the AssociatedMember parameter is set to TRUE, the orphaned device is associated with the coordinator. In this case, the MLME generates and sends the coordinator realignment command (see 7.3.2.5) to the orphaned device containing the value of the ShortAddress field. This command is sent in the CAP if the coordinator is on a beacon-enabled PAN or immediately otherwise. If the AssociatedMember parameter is set to FALSE, the orphaned device is not associated with the coordinator and this primitive will be ignored. If the orphaned device does not receive the coordinator realignment command following its orphan notification within *aResponseWaitTime* symbols, it will assume it is not associated to any coordinator in range.

The SecurityEnable parameter specifies whether security is to be applied to the coordinator realignment command frame. If the SecurityEnable parameter is set to FALSE, the MLME will set the security enabled subfield of the frame control field to 0 (see 7.2.1.1.2) and not perform any security operations on the frame. If the SecurityEnable parameter is set to TRUE, the MLME will set the security enabled subfield of the frame control field to 1 and obtain the key and security information, corresponding to the ExtendedAddress parameter, from the ACL entries in the MAC PIB, as described in 7.5.8.4.1. If an appropriate key could not be found in the ACL, the MLME will discard the frame and issue the MLME-COMM-STATUS.indication primitive with a status of UNAVAILABLE_KEY. If an appropriate key was found in the ACL, the MLME will use it to apply security to the frame, according to the security information found in the ACL (see 7.5.8.4). If any other error occurs during the secure processing of the frame, the MLME will discard the frame and issue the MLME-COMM-STATUS.indication primitive with a status of FAILED_SECURITY_CHECK.

If the CSMA-CA algorithm failed due to adverse conditions on the channel, the MAC sublayer will discard the MSDU and issue the MLME-COMM-STATUS.indication primitive with a status of CHANNEL_ACCESS_FAILURE.

To transmit the coordinator realignment command frame, the MLME first enables the transmitter by issuing the PLME-SET-TRX-STATE.request primitive with a state of TX_ON to the PHY. On receipt of the PLME-SET-TRX-STATE.confirm primitive with a status of either SUCCESS or TX_ON, the coordinator realignment command frame is then transmitted by issuing the PD-DATA.request primitive. Finally, if an acknowledgment was requested, the MLME enables the receiver on receipt of the PD-DATA.confirm primitive by issuing the PLME-SET-TRX-STATE.request primitive with a state of RX_ON to the PHY in preparation for the acknowledgment.

The MAC sublayer enables its receiver immediately following the transmission of the MPDU and waits for an acknowledgment from the recipient for at most *macAckWaitDuration* symbols. If the MAC sublayer does not receive an acknowledgment within this time, it will retry its transmission at most *aMaxFrameRetries* times. If the MAC sublayer still does not receive an acknowledgment from the recipient, it will discard the MSDU and issue the MLME-COMM-STATUS.indication primitive with a status of NO_ACK.

If the MPDU was successfully transmitted and an acknowledgment was received, if requested, the MAC sublayer will issue the MLME-COMM-STATUS.indication primitive with a status of SUCCESS.

If any parameter in the MLME-ORPHAN.response primitive is not supported or is out of range, the MAC sublayer will issue the MLME-COMM-STATUS.indication primitive with a status of INVALID_PARAMETER.

7.1.8.3 Orphan notification message sequence chart

Figure 30 illustrates the sequence of messages necessary for a coordinator to issue a notification of an orphaned device.

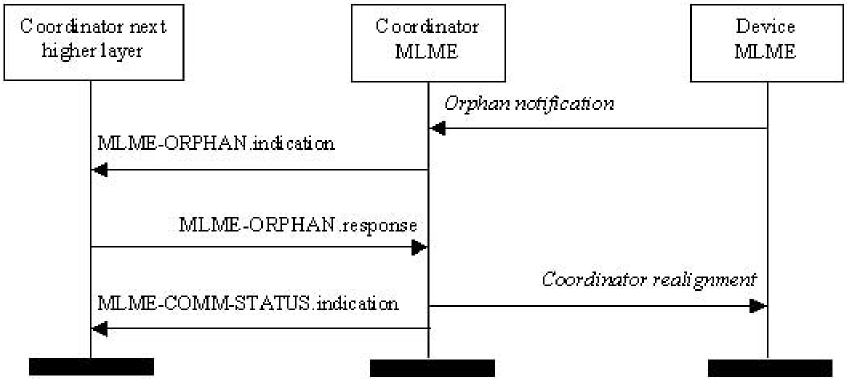


Figure 30—Message sequence chart for orphan notification

7.1.9 Primitives for resetting the MAC sublayer

MLME-SAP reset primitives specify how to reset the MAC sublayer to its default values.

All devices shall provide an interface for these reset primitives.

7.1.9.1 MLME-RESET.request

The MLME-RESET.request primitive allows the next higher layer to request that the MLME performs a reset operation.

7.1.9.1.1 Semantics of the service primitive

The semantics of the MLME-RESET.request primitive is as follows:

```
MLME-RESET.request          (
                            SetDefaultPIB
                            )
```

Table 49 specifies the parameter for the MLME-RESET.request primitive.

Table 49—MLME-RESET.request parameter

Name	Type	Valid range	Description
SetDefaultPIB	Boolean	TRUE or FALSE	If TRUE, the MAC sublayer is reset and all MAC PIB attributes are set to their default values. If FALSE, the MAC sublayer is reset but all MAC PIB attributes retain their values prior to the generation of the MLME-RESET.request primitive.

7.1.9.1.2 When generated

The MLME-RESET.request primitive is generated by the next higher layer and issued to its MLME to request a reset of the MAC sublayer to its initial conditions. The MLME-RESET.request primitive is issued prior to the use of the MLME-START.request or the MLME-ASSOCIATE.request primitives. If this primitive is issued to the MLME of an associated device or coordinator, any required disassociation attempts using the MLME-DISASSOCIATE.request primitive will be made a priori at the discretion of the next higher layer.

7.1.9.1.3 Effect on receipt

On receipt of the MLME-RESET.request primitive, the MLME issues the PLME-SET-TRX-STATE.request primitive with a state of TRX_OFF. On receipt of the PLME-SET-TRX-STATE.confirm primitive, the MAC sublayer is then set to its initial conditions, clearing all internal variables to their default values. If the SetDefaultPIB parameter is set to TRUE, the MAC PIB attributes are set to their default values.

If the PLME-SET-TRX-STATE.confirm primitive is successful, the MLME will issue the MLME-RESET.confirm primitive with the status of SUCCESS. Otherwise, the MLME issues the MLME-RESET.confirm primitive with the status of DISABLE_TRX_FAILURE.

7.1.9.2 MLME-RESET.confirm

The MLME-RESET.confirm primitive reports the results of the reset operation.

7.1.9.2.1 Semantics of the service primitive

The semantics of the MLME-RESET.confirm primitive is as follows:

```
MLME-RESET.confirm              (
                                status
                                )
```

Table 50 specifies the parameter for the MLME-RESET.confirm primitive.

Table 50—MLME-RESET.confirm parameter

Name	Type	Valid range	Description
status	Enumeration	SUCCESS or DISABLE_TRX_FAILURE	The result of the reset operation.

7.1.9.2.2 When generated

The MLME-RESET.confirm primitive is generated by the MLME and issued to its next higher layer in response to an MLME-RESET.request primitive and following the receipt of the PLME-SET-TRX-STATE.confirm primitive.

7.1.9.2.3 Effect on receipt

On receipt of the MLME-RESET.confirm primitive, the next higher layer is notified of its request to reset the MAC sublayer. This primitive returns a status of SUCCESS if the request to reset the MAC sublayer was successful. Otherwise, the status is set to DISABLE_TRX_FAILURE, indicating that the attempt to disable the transceiver was unsuccessful.

7.1.10 Primitives for specifying the receiver enable time

MLME-SAP receiver state primitives define how a device can enable or disable the receiver at a given time.

All devices shall provide an interface for these receiver state primitives.

7.1.10.1 MLME-RX-ENABLE.request

The MLME-RX-ENABLE.request primitive allows the next higher layer to request that the receiver is enable for a finite period of time.

7.1.10.1.1 Semantics of the service primitive

The semantics of the MLME-RX-ENABLE.request primitive is as follows:

```
MLME-RX-ENABLE.request          (
                                DeferPermit,
                                RxOnTime,
                                RxOnDuration
                                )
```

Table 51 specifies the parameters for the MLME-RX-ENABLE.request primitive.

Table 51—MLME-RX-ENABLE.request parameters

Name	Type	Valid range	Description
DeferPermit	Boolean	TRUE or FALSE	TRUE if the receiver enable can be deferred until during the next superframe if the requested time has already passed. FALSE if the receiver enable is only to be attempted in the current superframe. This parameter is ignored for nonbeacon-enabled PANs.
RxOnTime	Integer	0 x 000000–0 x ffffff	The number of symbols from the start of the superframe before the receiver is to be enabled. The precision of this value is a minimum of 20 bits, with the lowest 4 bits being the least significant. This parameter is ignored for nonbeacon-enabled PANs.
RxOnDuration	Integer	0 x 000000–0 x ffffff	The number of symbols for which the receiver is to be enabled.

7.1.10.1.2 When generated

The MLME-RX-ENABLE.request primitive is generated by the next higher layer and issued to the MLME to enable the receiver for a fixed duration, at a time relative to the start of the current or next superframe on a beacon-enabled PAN or immediately on a nonbeacon-enabled PAN. The receiver is enabled exactly once per primitive request.

7.1.10.1.3 Effect on receipt

On a nonbeacon-enabled PAN, the MLME ignores the DeferPermit and RxOnTime parameters and requests that the PHY enable the receiver immediately and disable it after RxOnDuration symbols.

On a beacon-enabled PAN, the MLME first determines whether (RxOnTime + RxOnDuration) is less than the beacon interval, defined by *macBeaconOrder*. If it is not less, the MLME issues the MLME-RX-ENABLE.confirm primitive with a status of INVALID_PARAMETER.

The MLME then determines whether the receiver can be enabled in the current superframe. If the current number of symbols measured from the start of the superframe is less than (RxOnTime – *aTurnaroundTime*), the MLME attempts to enable the receiver in the current superframe, as described below. If the current number of symbols measured from the start of the superframe is greater than or equal to (RxOnTime – *aTurnaroundTime*) and DeferPermit is equal to TRUE, the MLME defers until the next superframe and attempts to enable the receiver in that superframe as described below. Otherwise, the MLME issues the MLME-RX-ENABLE.confirm primitive with a status of OUT_OF_CAP.

The MLME then attempts to determine whether the receiver can be enabled within the CAP. If (RxOnTime – *aTurnaroundTime*) is greater than the current length of the CAP, the MLME issues the MLME-RX-ENABLE.confirm primitive with a status of OUT_OF_CAP. Otherwise, the MLME requests that the receiver is enabled RxOnTime symbols from the start of the superframe.

The MLME requests that the PHY enable the receiver by issuing the PLME-SET-TRX-STATE.request primitive with a state of RX_ON. If the PHY issues the PLME-SET-TRX-STATE.confirm primitive with a status of TX_ON, the MLME issues the MLME-RX-ENABLE.confirm primitive with a status of TX_ACTIVE. Otherwise, the MLME issues the MLME-RX-ENABLE.confirm primitive with a status of SUCCESS.

If (RxOnTime + RxOnDuration) does not extend beyond the CAP, the MLME disables the receiver RxOnDuration symbols after it was enabled by issuing the PLME-SET-TRX-STATE.request primitive with a state of TRX_OFF. If (RxOnTime + RxOnDuration) extends beyond the current CAP, the MLME ensures that the receiver does not conflict with any requirements that come into operation after the CAP (e.g., GTSs or receiving the next beacon).

If the RxOnDuration parameter is equal to 0, the MLME requests that the PHY disable its receiver.

7.1.10.2 MLME-RX-ENABLE.confirm

The MLME-RX-ENABLE.confirm primitive reports the results of the attempt to enable the receiver.

7.1.10.2.1 Semantics of the service primitive

The semantics of the MLME-RX-ENABLE.confirm primitive is as follows:

```
    MLME-RX-ENABLE.confirm          (
                                    status
                                    )
```

Table 52 specifies the parameter for the MLME-RX-ENABLE.confirm primitive.

Table 52—MLME-RX-ENABLE.confirm parameter

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, TX_ACTIVE, OUT_OF_CAP, or INVALID_PARAMETER	The result of the receiver enable request.

7.1.10.2.2 When generated

The MLME-RX-ENABLE.confirm primitive is generated by the MLME and issued to its next higher layer in response to an MLME-RX-ENABLE.request primitive.

7.1.10.2.3 Effect on receipt

On receipt of the MLME-RX-ENABLE.confirm primitive, the next higher layer is notified of its request to enable the receiver. This primitive returns a status of either SUCCESS, if the request to enable the receiver was successful, or an error code of TX_ACTIVE, OUT_OF_CAP, or INVALID_PARAMETER. The reasons for these status values are fully described in 7.1.10.1.3.

7.1.10.3 Message sequence chart for changing the state of the receiver

Figure 30 illustrates the sequence of messages necessary for enabling the receiver for a fixed duration. Part a) illustrates the case for a beacon-enabled PAN where it is assumed that the MLME-RX-ENABLE.request has been received by the MLME without sufficient time available to enable the receiver in the current superframe. Part b) illustrates the case for a nonbeacon-enabled PAN where the receiver is enabled immediately.

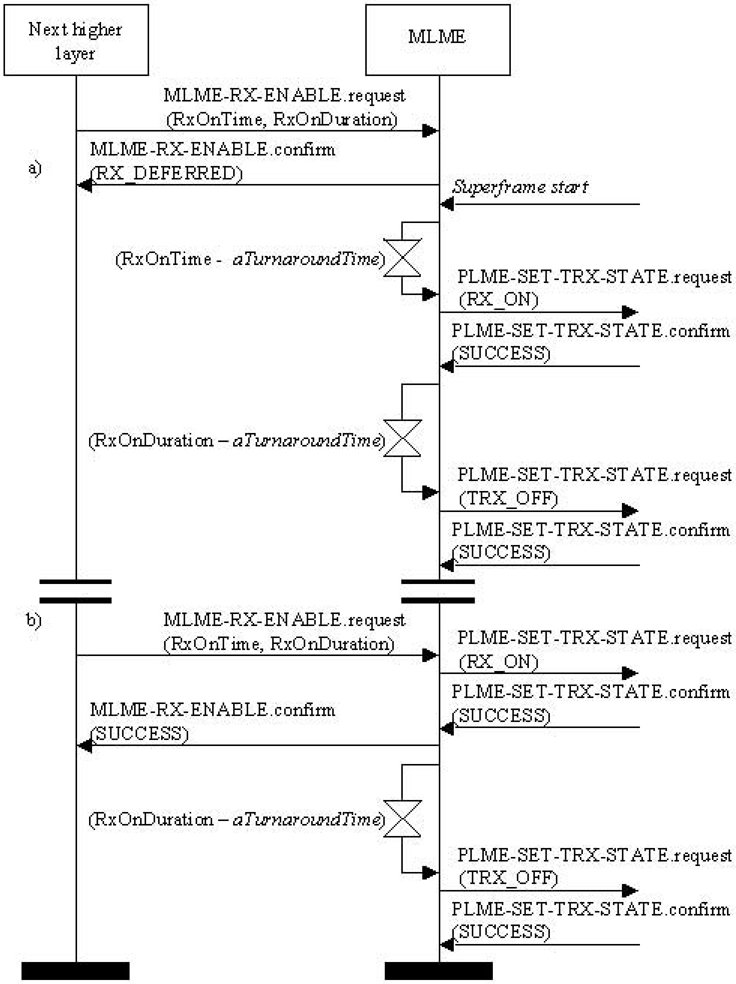

Next higher layer
MLME
MLME-RX-ENABLE.request (RxOnTime, RxOnDuration)
MLME-RX-ENABLE.confirm (RX_DEFERRED)
a)
Superframe start
(RxOnTime - aTurnaroundTime)
PLME-SET-TRX-STATE.request (RX_ON)
PLME-SET-TRX-STATE.confirm (SUCCESS)
(RxOnDuration – aTurnaroundTime)
PLME-SET-TRX-STATE.request (TRX_OFF)
PLME-SET-TRX-STATE.confirm (SUCCESS)
b)
MLME-RX-ENABLE.request (RxOnTime, RxOnDuration)
PLME-SET-TRX-STATE.request (RX_ON)
PLME-SET-TRX-STATE.confirm (SUCCESS)
MLME-RX-ENABLE.confirm (SUCCESS)
(RxOnDuration – aTurnaroundTime)
PLME-SET-TRX-STATE.request (TRX_OFF)
PLME-SET-TRX-STATE.confirm (SUCCESS)

Figure 30—Message sequence chart for changing the state of the receiver

7.1.11 Primitives for channel scanning

MLME-SAP scan primitives define how a device can determine the energy usage or the presence or absence of PANs in a communications channel.

All devices shall provide an interface for these scan primitives.

7.1.11.1 MLME-SCAN.request

The MLME-SCAN.request primitive is used to initiate a channel scan over a given list of channels. A device can use a channel scan to measure the energy on the channel, search for the coordinator with which it associated, or search for all coordinators transmitting beacon frames within the POS of the scanning device.

7.1.11.1.1 Semantics of the service primitive

The semantics of the MLME-SCAN.request primitive is as follows:

MLME-SCAN.request (
ScanType,
ScanChannels,
ScanDuration
)

Table 53 specifies the parameters for the MLME-SCAN.request primitive.

Table 53—MLME-SCAN.request parameters

Name	Type	Valid range	Description
ScanType	Integer	0 x 00–0 x 03	Indicates the type of scan performed: 0 x 00 = ED scan (FFD only). 0 x 01 = active scan (FFD only). 0 x 02 = passive scan. 0 x 03 = orphan scan.
ScanChannels	Bitmap	32 bit field	The 5 MSBs (b_{27}, ... , b_{31}) are reserved. The 27 LSBs (b_0, b_1, ... b_{26}) indicate which channels are to be scanned (1 = scan, 0 = do not scan) for each of the 27 valid channels (see 6.1.2).
ScanDuration	Integer	0–14	A value used to calculate the length of time to spend scanning each channel for ED, active, and passive scans. This parameter is ignored for orphan scans. The time spent scanning each channel is [*aBaseSuperframeDuration* * (2^n + 1)] symbols, where *n* is the value of the ScanDuration parameter.

7.1.11.1.2 When generated

The MLME-SCAN.request primitive is generated by the next higher layer and issued to its MLME to initiate a channel scan to search for activity within the POS of the device. This primitive can be used to perform an ED scan to determine channel usage, an active or passive scan to locate beacon frames containing any PAN identifier, or an orphan scan to locate a PAN to which the device is currently associated. See 7.5.2.1 for a description of each type of scan in detail.

ED or active scans can be performed before an FFD starts operation as a PAN coordinator. Active or passive scans can be performed prior to selecting a PAN for association. Orphan scans can be performed to attempt to locate a specific coordinator with which communication has been lost.

All devices shall be capable of performing passive scans and orphan scans; ED scans and active scans are optional for an RFD.

7.1.11.1.3 Effect on receipt

When the MLME receives the MLME-SCAN.request primitive, it initiates a scan in all channels specified in the ScanChannels parameter. The MLME suspends all beacon transmissions for the duration of the scan. During a scan, the MAC sublayer only accepts frames received over the PHY data service that are relevant to the scan being performed (see 7.5.2.1).

An ED scan allows a device to obtain a measure of the peak energy in each requested channel. The ED scan is performed on each channel by the MLME's repeatedly issuing the PLME-ED.request primitive to the PHY until [*aBaseSuperframeDuration* * (2^n + 1)] symbols, where n is the value of the ScanDuration parameter, have elapsed. The MLME notes the maximum energy measurement and moves on to the next channel in the channel list. A device will be able to store between one and an implementation-specified maximum number of channel ED measurements. The ED scan terminates when the number of channel ED measurements stored equals this implementation-specified maximum or when every channel specified in the channel list has been scanned.

An active scan is used by an FFD to locate all coordinators transmitting beacon frames within its POS. The active scan is performed on each channel by the MLME's first sending a beacon request command (see 7.3.2.4). The MLME then enables the receiver and records the information contained in each received beacon in a PAN descriptor structure (see Table 41 in 7.1.5.1.1). A device will be able to store between one and an implementation-specified maximum number of PAN descriptors. The active scan on a particular channel terminates when the number of PAN descriptors stored equals this implementation-specified maximum or when [*aBaseSuperframeDuration* * (2^n + 1)] symbols, where n is the value of the ScanDuration parameter, have elapsed. If the latter condition has been satisfied, the channel is considered to have been scanned. Where possible, the scan is repeated on each channel and terminates when the number of PAN descriptors stored equals the implementation-specified maximum or when every channel in the set of available channels has been scanned.

A passive scan, like an active scan, is used to locate all coordinators transmitting beacon frames within the POS of the scanning device; the difference is that the passive scan is a receive-only operation and does not transmit the beacon request command. The passive scan is performed on each channel by the MLME's enabling its receiver and recording the information contained in each received beacon in a PAN descriptor structure (see Table 41 in 7.1.5.1.1). A device will be able to store between one and an implementation-specified maximum number of PAN descriptors. The passive scan on a particular channel terminates when the number of PAN descriptors stored equals this implementation-specified maximum or when [*aBaseSuperframeDuration* * (2^n + 1)] symbols, where n is the value of the ScanDuration parameter, have elapsed. If the latter condition has been satisfied, the channel is considered to have been scanned. Where possible, the scan is repeated on each channel and terminates when the number of PAN descriptors stored equals the implementation-specified maximum or when every channel in the set of available channels has been scanned.

An orphan scan is used to locate the coordinator with which the scanning device had previously associated. The orphan scan is performed on each channel by the MLME first sending an orphan notification command (see 7.3.2.3). The MLME then enables its receiver for at most *aResponseWaitTime* symbols. If the device receives a coordinator realignment command within this time, the MLME will disable its receiver. Otherwise, the scan is repeated on the next channel in the channel list. The scan terminates when the device receives a coordinator realignment command (see 7.3.2.5) or when every channel in the set of available channels has been scanned.

The results of an ED scan are recorded in a list of ED values and reported to the next higher layer through the MLME-SCAN.confirm primitive with a status of SUCCESS.

The results of an active or passive scan are recorded in a set of PAN descriptor values and reported to the next higher layer through the MLME-SCAN.confirm primitive. If no beacons were found, the

MLME-SCAN.confirm primitive will contain a null set of PAN descriptor values and a status of NO_BEACON. Otherwise, the MLME-SCAN.confirm primitive will contain the set of PANDescriptor values found, a list of unscanned channels, and a status of SUCCESS.

The results of an orphan scan are reported to the next higher layer through the MLME-SCAN.confirm primitive. If successful, the MLME-SCAN.confirm primitive will contain a status of SUCCESS. If the device did not receive a coordinator realignment command, the MLME-SCAN.confirm primitive will contain a status of NO_BEACON. In either case, the PANDescriptorList and EnergyDetectList parameters of the MLME-SCAN.confirm primitive are null.

If any parameter in the MLME-SCAN.request primitive is not supported or is out of range, the MAC sublayer will issue the MLME-SCAN.confirm primitive with a status of INVALID_PARAMETER.

7.1.11.2 MLME-SCAN.confirm

The MLME-SCAN.confirm primitive reports the result of the channel scan request.

7.1.11.2.1 Semantics of the service primitive

The semantics of the MLME-SCAN.confirm primitive is as follows:

```
MLME-SCAN.confirm          (
                           status,
                           ScanType,
                           UnscannedChannels,
                           ResultListSize,
                           EnergyDetectList,
                           PANDescriptorList
                           )
```

Table 54 specifies the parameters for the MLME-SCAN.confirm primitive.

Table 54—MLME-SCAN.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, NO_BEACON, or INVALID_PARAMETER	The status of the scan request.
ScanType	Integer	0 x 00–0 x 03	Indicates if the type of scan performed: 0 x 00 = ED scan (FFD only). 0 x 01 = active scan (FFD only). 0 x 02 = passive scan. 0 x 03 = orphan scan.
UnscannedChannels	Bitmap	32 bit field	Indicates which channels given in the request were not scanned (1 = not scanned, 0 = scanned or not requested). This parameter is only valid for passive or active scans.

Table 54—MLME-SCAN.confirm parameters *(continued)*

Name	Type	Valid range	Description
ResultListSize	Integer	Implementation specific	The number of elements returned in the appropriate result lists. This value is 0 for the result of an orphan scan.
EnergyDetectList	List of integers	0 x 00–0 x ff for each integer	The list of energy measurements, one for each channel searched during an ED scan. This parameter is null for active, passive, and orphan scans.
PANDescriptorList	List of PAN descriptor values	See Table 41	The list of PAN descriptors, one for each beacon found during an active or passive scan. This parameter is null for ED and orphan scans.

7.1.11.2.2 When generated

The MLME-SCAN.confirm primitive is generated by the MLME and issued to its next higher layer when the channel scan initiated with the MLME-SCAN.request primitive has completed. If the MLME-SCAN.request primitive requested an active, passive, or orphan scan, the EnergyDetectList parameter will be null. If the MLME-SCAN.request primitive requested an ED or orphan scan, the PANDescriptorList parameter will be null. If the MLME-SCAN.request primitive requested an orphan scan, the ResultListSize parameter will be set to 0.

The MLME-SCAN.confirm primitive returns a status of either SUCCESS, indicating that the requested scan was successful, or an error code of NO_BEACON or INVALID_PARAMETER. The reasons for these status values are fully described in 7.1.11.1.3.

7.1.11.2.3 Effect on receipt

On receipt of the MLME-SCAN.confirm primitive, the next higher layer is notified of the results of the scan procedure. If the requested scan was successful, the status parameter will be set to SUCCESS. Otherwise, the status parameter indicates the error.

7.1.11.3 Channel scan message sequence charts

Figure 77 and Figure 80 (see 7.7) illustrate the sequence of messages necessary to perform an ED scan and a passive scan, respectively. These figures include steps taken by the PHY.

7.1.12 Communication status primitive

The MLME-SAP communication status primitive defines how the MLME communicates to the next higher layer about transmission status, when the transmission was not instigated by a .request primitive, and security errors on incoming packets.

All devices shall provide an interface for this communication status primitive.

7.1.12.1 MLME-COMM-STATUS.indication

The MLME-COMM-STATUS.indication primitive allows the MLME to indicate a communications status.

7.1.12.1.1 Semantics of the service primitive

The semantics of the MLME-COMM-STATUS.indication primitive is as follows:

```
MLME-COMM-STATUS.indication    (
                               PANId,
                               SrcAddrMode,
                               SrcAddr,
                               DstAddrMode,
                               DstAddr,
                               status
                               )
```

Table 55 specifies the parameters for the MLME-COMM-STATUS.indication primitive.

Table 55—MLME-COMM-STATUS.indication parameters

Name	Type	Valid range	Description
PANId	Integer	0 x 0000—0 ffff	The 16 bit PAN identifier of the device from which the frame was received or to which the frame was being sent.
SrcAddrMode	Integer	0 x 00—0 x 03	The source addressing mode for this primitive. This value can take one of the following values: 0 = no address (addressing fields omitted). 0 x 01 = reserved. 0 x 02 = 16 bit short address. 0 x 03 = 64 bit extended address.
SrcAddr	Device address	As specified by the SrcAddrMode parameter	The individual device address of the entity from which the frame causing the error originated.
DstAddrMode	Integer	0 x 00—0 x 03	The destination addressing mode for this primitive. This value can take one of the following values: 0 x 00 = no address (addressing fields omitted). 0 x 01 = reserved. 0 x 02 = 16 bit short address. 0 x 03 = 64 bit extended address.
DstAddr	Device address	As specified by the DstAddrMode parameter	The individual device address of the device for which the frame was intended.
Status	Enumeration	SUCCESS, TRANSACTION_OVERFLOW, TRANSACTION_EXPIRED, CHANNEL_ACCESS_FAILURE, NO_ACK, UNAVAILABLE_KEY, FRAME_TOO_LONG, FAILED_SECURITY_CHECK or INVALID_PARAMETER	The communications status.

7.1.12.1.2 When generated

The MLME-COMM-STATUS.indication primitive is generated by the MLME and issued to its next higher layer either following a transmission instigated through a .response primitive or on receipt of a frame that generates an error in its secure processing.

The MLME-COMM-STATUS.indication primitive is generated by the MAC sublayer entity following either the MLME-ASSOCIATE.response primitive or the MLME-ORPHAN.response primitive. This primitive returns a status of either SUCCESS, indicating that the request to transmit was successful, or an error code of TRANSACTION_OVERFLOW, TRANSACTION_EXPIRED, CHANNEL_ACCESS_FAILURE, NO_ACK, UNAVAILABLE_KEY, FRAME_TOO_LONG, FAILED_SECURITY_CHECK, or INVALID_PARAMETER. The reasons for these status values are fully described in 7.1.3.3.3 and 7.1.8.2.3, respectively.

If a secure frame is received and the appropriate key is not available in the ACL, the MAC sublayer will discard the MSDU and issue this primitive with a status of UNAVAILABLE_KEY. If the appropriate key is available in the ACL but the secure processing of the frame results in a MAC frame payload longer than *aMaxMACFrameSize*, the MLME will issue this primitive with a status of FRAME_TOO_LONG. If the appropriate key is available in the ACL, but an error occurs during the secure processing of the frame, the MAC sublayer will discard the MSDU and issue this primitive with a status of FAILED_SECURITY_CHECK.

7.1.12.1.3 Effect on receipt

On receipt of the MLME-COMM-STATUS.indication primitive, the next higher layer is notified of the communication status of a transmission or notified of an error that has occurred during the secure processing of incoming frame.

7.1.13 Primitives for writing MAC PIB attributes

MLME-SAP set primitives define how MAC PIB attributes may be written.

All devices shall provide an interface for these set primitives.

7.1.13.1 MLME-SET.request

The MLME-SET.request primitive attempts to write the given value to the indicated MAC PIB attribute.

7.1.13.1.1 Semantics of the primitive

The semantics of the MLME-SET.request primitive is as follows:

```
MLME-SET.request                  (
                                  PIBAttribute,
                                  PIBAttributeValue
                                  )
```

Table 56 specifies the parameters for the MLME-SET.request primitive.

7.1.13.1.2 When generated

The MLME-SET.request primitive is generated by the next higher layer and issued to its MLME to write the indicated MAC PIB attribute.

Table 56—MLME-SET.request parameters

Name	Type	Valid range	Description
PIBAttribute	Integer	See Table 71 and Table 72	The identifier of the MAC PIB attribute to write.
PIBAttributeValue	Various	Attribute specific; see Table 71 and Table 72	The value to write to the indicated MAC PIB attribute.

7.1.13.1.3 Effect on receipt

On receipt of the MLME-SET.request primitive, the MLME attempts to write the given value to the indicated MAC PIB attribute in its database. If the PIBAttribute parameter specifies an attribute that is not found in the database (see Table 71 and Table 72), the MLME will issue the MLME-SET.confirm primitive with a status of UNSUPPORTED_ATTRIBUTE. If the PIBAttibuteValue parameter specifies a value that is out of the valid range for the given attribute, the MLME will issue the MLME-SET.confirm primitive with a status of INVALID_PARAMETER.

If the requested MAC PIB attribute is successfully written, the MLME will issue the MLME-SET.confirm primitive with a status of SUCCESS.

7.1.13.2 MLME-SET.confirm

The MLME-SET.confirm primitive reports the results of an attempt to write a value to a MAC PIB attribute.

7.1.13.2.1 Semantics of the service primitive

The semantics of the MLME-SET.confirm primitive is as follows:

```
    MLME-SET.confirm                (
                                    status,
                                    PIBAttribute
                                    )
```

Table 57 specifies the parameters for the MLME-SET.confirm primitive.

Table 57—MLME-SET.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, UNSUPPORTED_ATTRIBUTE, or INVALID_PARAMETER	The result of the request to write the MAC PIB attribute.
PIBAttribute	Integer	See Table 71 and Table 72	The identifier of the MAC PIB attribute that was written.

7.1.13.2.2 When generated

The MLME-SET.confirm primitive is generated by the MLME and issued to its next higher layer in response to an MLME-SET.request primitive. The MLME-SET.confirm primitive returns a status of either

SUCCESS, indicating that the requested value was written to the indicated MAC PIB attribute, or an error code of UNSUPPORTED_ATTRIBUTE or INVALID_PARAMETER. The reasons for these status values are fully described in 7.1.13.1.3.

7.1.13.2.3 Effect on receipt

On receipt of the MLME-SET.confirm primitive, the next higher layer is notified of the result of its request to set the value of a MAC PIB attribute. If the requested value was written to the indicated MAC PIB attribute, the status parameter will be set to SUCCESS. Otherwise, the status parameter indicates the error.

7.1.14 Primitives for updating the superframe configuration

MLME-SAP start primitives define how an FFD can request to start using a new superframe configuration in order to initiate a PAN, begin transmitting beacons on an already existing PAN, facilitating device discovery, or to stop transmitting beacons.

These start primitives are optional for an RFD.

7.1.14.1 MLME-START.request

The MLME-START.request primitive makes a request for the device to start using a new superframe configuration.

7.1.14.1.1 Semantics of the service primitive

The semantics of the MLME-START.request primitive is as follows:

```
MLME-START.request          (
                            PANId,
                            LogicalChannel,
                            BeaconOrder,
                            SuperframeOrder,
                            PANCoordinator,
                            BatteryLifeExtension,
                            CoordRealignment,
                            SecurityEnable
                            )
```

Table 58 specifies the parameters for the MLME-START.request primitive.

Table 58—MLME-START.request parameters

Name	Type	Valid range	Description
PANId	Integer	0 x 0000–0 x ffff	The PAN identifier to be used by the beacon.
LogicalChannel	Integer	Selected from the available logical channels supported by the PHY	The logical channel on which to start transmitting beacons.

Table 58—MLME-START.request parameters *(continued)*

Name	Type	Valid range	Description
BeaconOrder	Integer	0–15	How often the beacon is to be transmitted. The beacon order, *BO*, and the beacon interval, *BI*, are related as follows: for $0 \leq BO \leq 14$, $BI = aBaseSuperframeDuration * 2^{BO}$ symbols. If *BO* = 15, the coordinator will not transmit a beacon, and the SuperframeOrder parameter value is ignored.
SuperframeOrder	Integer	0–*BO* or 15	The length of the active portion of the superframe, including the beacon frame. The superframe order, *SO*, and the superframe duration, *SD*, are related as follows: for $0 \leq SO \leq \mathrm{BO} \leq 14$, $SD = aBaseSuperframeDuration * 2^{SO}$ symbols. If *SO* = 15, the superframe will not be active after the beacon.
PANCoordinator	Boolean	TRUE or FALSE	If this value is TRUE, the device will become the PAN coordinator of a new PAN. If this value is FALSE, the device will begin transmitting beacons on the PAN with which it is associated.
BatteryLifeExtension	Boolean	TRUE or FALSE	If this value is TRUE, the receiver of the beaconing device is disabled *macBattLifeExtPeriods* full backoff periods after the interframe spacing (IFS) period of the beacon frame. If this value is FALSE, the receiver of the beaconing device remains enabled for the entire CAP.
CoordRealignment	Boolean	TRUE or FALSE	TRUE if a coordinator realignment command is to be transmitted prior to changing the superframe configuration or FALSE otherwise.
SecurityEnable	Boolean	TRUE or FALSE	TRUE if security is enabled for beacon transmissions or FALSE otherwise.

7.1.14.1.2 When generated

The MLME-START.request primitive is generated by the next higher layer and issued to its MLME to request that a device start using a new superframe configuration.

7.1.14.1.3 Effect on receipt

If the MLME-START.request primitive is received when *macShortAddress* is set to 0 x ffff, the MLME will issue the MLME-START.confirm primitive with a status of NO_SHORT_ADDRESS.

On receipt of the MLME-START.request primitive, the MLME sets *macBeaconOrder* to the value of the BeaconOrder parameter. If *macBeaconOrder* is equal to 15, the MLME will also set *macSuperframeOrder* to 15. In this case, this primitive configures a beaconless PAN. If *macBeaconOrder* is less than 15, the MLME will set *macSuperframeOrder* to the value of the SuperframeOrder parameter.

When the PANCoordinator parameter is set to TRUE, the MLME updates *macPANId* with the value of the PANId parameter and *phyCurrentChannel* with the value of the LogicalChannel parameter, by issuing the PLME-SET.request primitive. When the PANCoordinator parameter is set to FALSE, the MLME ignores the PANId and LogicalChannel parameters.

If the CoordRealignment parameter is set to TRUE, the MLME generates and broadcasts a coordinator realignment command containing the new PANId and LogicalChannel parameters.

When the device is already transmitting beacons, the new superframe configuration is put into operation at the next scheduled beacon. If the device is not already transmitting beacons, the new superframe configuration is put into operation immediately.

The address used by the coordinator in its beacon frames is determined by the current value of *macShort-Address*, which is set by the next higher layer before issuing this primitive. Also the MLME sets *macBatt-LifeExt* to the value of the BatteryLifeExtension parameter.

The SecurityEnable parameter specifies whether security is to be applied to all following beacon frames. If the SecurityEnable parameter is set to FALSE, the MAC sublayer will set the security enabled subfield of the frame control field to 0 (see 7.2.1.1.2) and not perform any security operations on the frame. If the SecurityEnable parameter is set to TRUE, the MAC sublayer will set the security enabled subfield of the frame control field to 1 and obtain the key and security information, corresponding to the broadcast address, from the ACL entries in the MAC PIB, as described in 7.5.8.4.1. If an appropriate key could not be found in the ACL, the MAC sublayer will discard the frame and issue the MLME-START.confirm primitive with a status of UNAVAILABLE_KEY. If an appropriate key was found in the ACL, the MAC sublayer will use it to apply security to the frame, according to the security information found in the ACL (see 7.5.8.4). If the length of the resulting frame is longer than *aMaxMACFrameSize*, the MLME will issue the MLME-START.confirm primitive with a status of FRAME_TOO_LONG. If any other error occurs during the secure processing of the frame, the MAC sublayer will discard the frame and issue the MLME-START.confirm primitive with a status of FAILED_SECURITY_CHECK.

To transmit any beacon frame, the MLME first enables the transmitter by issuing the PLME-SET-TRX-STATE.request primitive with a state of TX_ON to the PHY. On receipt of the PLME-SET-TRX-STATE.confirm primitive with a status of either SUCCESS or TX_ON, the beacon frame is then transmitted by issuing the PD-DATA.request primitive. Finally, on receipt of the PD-DATA.confirm primitive and if the active portion of the superframe extends beyond the beacon frame transmission (see 7.5.1.1), the MLME of the coordinator will enable the receiver by issuing the PLME-SET-TRX-STATE.request primitive with a state of RX_ON to the PHY. If the active portion of the superframe ends after the beacon frame transmission, the receiver will not be enabled.

On completion of this procedure, the MLME responds with the MLME-START.confirm primitive. If the attempt to start using a new superframe configuration was successful, the status parameter will be set to SUCCESS. If any parameter is not supported or is out of range, the status parameter will be set to INVALID_PARAMETER.

7.1.14.2 MLME-START.confirm

The MLME-START.confirm primitive reports the results of the attempt to start using a new superframe configuration.

7.1.14.2.1 Semantics of the service primitive

The semantics of the MLME-START.confirm primitive is as follows:

```
MLME-START.confirm            (
                              status
                              )
```

Table 59 specifies the parameters for the MLME-START.confirm primitive.

Table 59—MLME-START.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, NO_SHORT_ADDRESS, UNAVAILABLE_KEY, FRAME_TOO_LONG, FAILED_SECURITY_CHECK, or INVALID_PARAMETER	The result of the attempt to start using an updated superframe configuration.

7.1.14.2.2 When generated

The MLME-START.confirm primitive is generated by the MLME and issued to its next higher layer in response to an MLME-START.request primitive. The MLME-START.confirm primitive returns a status of either SUCCESS, indicating that the MAC sublayer has started using the new superframe configuration, or an error code of NO_SHORT_ADDRESS, UNAVAILABLE_KEY, FRAME_TOO_LONG, FAILED_SECURITY_CHECK, or INVALID_PARAMETER. The reasons for these status values are fully described in 7.1.14.1.3.

7.1.14.2.3 Effect on receipt

On receipt of the MLME-START.confirm primitive, the next higher layer is notified of the result of its request to start using a new superframe configuration. If the MAC sublayer has been successful, the status parameter will be set to SUCCESS. Otherwise, the status parameter indicates the error.

7.1.14.3 Message sequence chart for updating the superframe configuration

Figure 31 illustrates the sequence of messages necessary for initiating beacon transmissions in an FFD. Figure 76 (see 7.7) illustrates the sequence of messages necessary for a PAN coordinator to start beaconing on a new PAN; this figure includes steps taken by the PHY.

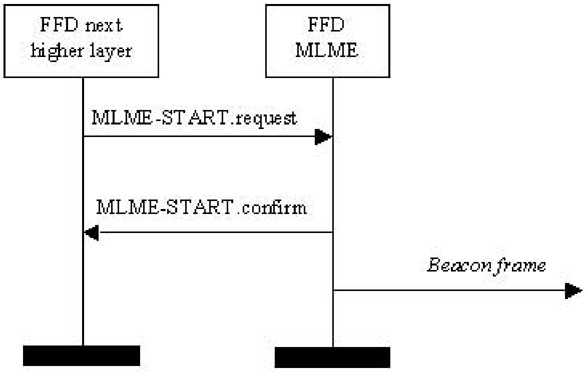


Figure 31—Message sequence chart for updating the superframe configuration

7.1.15 Primitives for synchronizing with a coordinator

MLME-SAP synchronization primitives define how synchronization with a coordinator may be achieved and how a loss of synchronization is communicated to the next higher layer.

All devices shall provide an interface for these synchronization primitives.

7.1.15.1 MLME-SYNC.request

The MLME-SYNC.request primitive requests to synchronize with the coordinator by acquiring and, if specified, tracking its beacons.

7.1.15.1.1 Semantics of the service primitive

The semantics of the MLME-SYNC.request primitive is as follows:

```
MLME-SYNC.request           (
                            LogicalChannel,
                            TrackBeacon
                            )
```

Table 60 specifies the parameters for the MLME-SYNC.request primitive.

Table 60—MLME-SYNC.request parameters

Name	Type	Valid range	Description
LogicalChannel	Integer	Selected from the available logical channels supported by the PHY	The logical channel on which to attempt coordinator synchronization.
TrackBeacon	Boolean	TRUE or FALSE	TRUE if the MLME is to synchronize with the next beacon and attempt to track all future beacons. FALSE if the MLME is to synchronize with only the next beacon.

7.1.15.1.2 When generated

The MLME-SYNC.request primitive is generated by the next higher layer of a device on a beacon-enabled PAN and issued to its MLME to synchronize with the coordinator.

7.1.15.1.3 Effect on receipt

If the MLME-SYNC.request primitive is received by the MLME on a beacon-enabled PAN, it will first set *phyCurrentChannel* equal to the LogicalChannel parameter by issuing the PLME-SET.request primitive to the PHY. The MLME then enables its receiver and searches for the current network beacon. If the TrackBeacon parameter is equal to TRUE, the MLME will track the beacon, i.e., enable its receiver just before the expected time of each beacon so that the beacon frame can be processed. If the TrackBeacon parameter is equal to FALSE, the MLME will locate the beacon, but not continue to track it.

If this primitive is received by the MLME while it is currently tracking the beacon, the MLME will not discard the primitive, but rather treat it as a new synchronization request.

If the beacon could not be located either on its initial search or during tracking, the MLME will issue the MLME-SYNC-LOSS.indication primitive with a loss reason of BEACON_LOST.

7.1.15.2 MLME-SYNC-LOSS.indication

The MLME-SYNC-LOSS.indication primitive indicates the loss of synchronization with a coordinator.

7.1.15.2.1 Semantics of the service primitive

The semantics of the MLME-SYNC-LOSS.indication primitive is as follows:

```
MLME-SYNC-LOSS.indication      (
                               LossReason
                               )
```

Table 61 specifies the parameters for the MLME-SYNC-LOSS.indication primitive.

Table 61—MLME-SYNC-LOSS.indication parameters

Name	Type	Valid range	Description
LossReason	Enumeration	PAN_ID_CONFLICT, REALIGNMENT, or BEACON_LOST	The reason that synchronization was lost.

7.1.15.2.2 When generated

The MLME-SYNC-LOSS.indication primitive is generated by the MLME of a device and issued to its next higher layer in the event of a loss of synchronization with the coordinator. It is also generated by the MLME of a PAN coordinator and issued to its next higher layer in the event of a PAN ID conflict.

If a device has detected a PAN identifier conflict and communicated it to the coordinator, the MLME will issue this primitive with a loss reason of PAN_ID_CONFLICT. Similarly, if a PAN coordinator receives a PAN ID conflict notification command (see 7.3.2.2), the MLME will issue this primitive with a loss reason of PAN_ID_CONFLICT.

If a device has received the coordinator realignment command (see 7.3.2.5) from the coordinator to which it associated and the MLME was not carrying out an orphan scan, the MLME will issue this primitive with a loss reason of REALIGNMENT.

If a device has not heard the beacon for *aMaxLostBeacons* consecutive superframes following an MLME-SYNC.request primitive, either initially or during tracking, the MLME will issue this primitive with a loss reason of BEACON_LOST. If the beacon was being tracked, the MLME will not attempt to track the beacon any further.

7.1.15.2.3 Effect on receipt

On receipt of the MLME-SYNC-LOSS.indication primitive, the next higher layer is notified of a loss of synchronization.

7.1.15.3 Message sequence chart for synchronizing with a coordinator

Figure 32 illustrates the sequence of messages necessary for a device to synchronize with a coordinator. In a), a single synchronization request is issued. The MLME then searches for a beacon and, if found, determines whether the coordinator has any data pending for the device. If so, the data are requested as described in 7.5.6.3. In b), a track synchronization request is issued. The MLME then searches for a beacon and, if found, attempts to keep track of it using a timer that expires just before the expected time of the next beacon. The MLME also checks for any data pending in the coordinator for the device when a beacon frame is received.

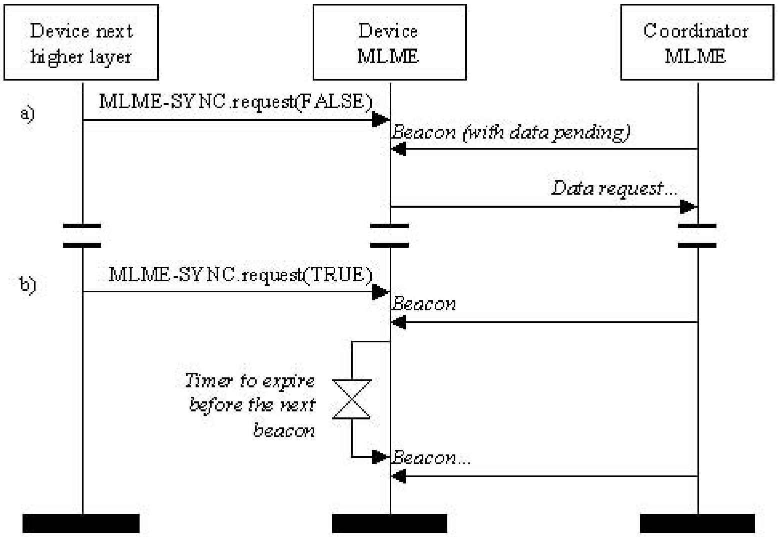


Figure 32—Message sequence chart for synchronizing to a coordinator in a beacon-enabled PAN

7.1.16 Primitives for requesting data from a coordinator

MLME-SAP polling primitives define how to request data from a coordinator.

All devices shall provide an interface for these polling primitives.

7.1.16.1 MLME-POLL.request

The MLME-POLL.request primitive prompts the device to request data from the coordinator.

7.1.16.1.1 Semantics of the service primitive

The semantics of the MLME-POLL.request primitive is as follows:

```
MLME-POLL.request          (
                           CoordAddrMode,
                           CoordPANId,
                           CoordAddress,
                           SecurityEnable
                           )
```

Table 62 specifies the parameter for the MLME-POLL.request primitive.

Table 62—MLME-POLL.request parameters

Name	Type	Valid range	Description
CoordAddrMode	Integer	0 x 02–0 x 03	The addressing mode of the coordinator to which the poll is intended. This parameter can take one of the following values: 2 = 16 bit short address, 3 = 64 bit extended address.
CoordPANId	Integer	0 x 0000–0 x fffe	The PAN identifier of the coordinator to which the poll is intended.
CoordAddress	Device-Address	As specified by the CoordAddrMode paramater	The address of the coordinator to which the poll is intended.
SecurityEnable	Boolean	TRUE or FALSE	TRUE if security is enabled for this transfer or FALSE otherwise.

7.1.16.1.2 When generated

The MLME-POLL.request primitive is generated by the next higher layer and issued to its MLME when data are to be requested from a coordinator.

7.1.16.1.3 Effect on receipt

On receipt of the MLME-POLL.request primitive, the MLME generates and sends a data request command (see 7.3.2.1). If the poll is directed to the PAN coordinator, the data request command is generated without any destination address information present. Otherwise, the data request command is generated with the destination address information in the CoordPANId and CoordAddress parameters.

The SecurityEnable parameter specifies whether security is to be applied to the data request command frame. If the SecurityEnable parameter is set to FALSE, the MLME will set the security enabled subfield of the frame control field to 0 (see 7.2.1.1.2) and not perform any security operations on the frame. If the SecurityEnable parameter is set to TRUE, the MLME will set the security enabled subfield of the frame control field to 1 and obtain the key and security information, corresponding to the CoordAddress parameter, from the ACL entries in the MAC PIB, as described in 7.5.8.4.1. If an appropriate key could not be found in the ACL, the MLME will discard the frame and issue the MLME-POLL.confirm primitive with a status of UNAVAILABLE_KEY. If an appropriate key was found in the ACL, the MLME will use it to apply security to the frame, according to the security information found in the ACL (see 7.5.8.4). If any other error occurs during the secure processing of the frame, the MLME will discard the frame and issue the MLME-POLL.confirm primitive with a status of FAILED_SECURITY_CHECK.

If the data request command cannot be sent due to a CSMA algorithm failure, the MLME will issue the MLME-POLL.confirm primitive with a status of CHANNEL_ACCESS_FAILURE.

To transmit the data request frame, the MLME first enables the transmitter by issuing the PLME-SET-TRX-STATE.request primitive with a state of TX_ON to the PHY. On receipt of the PLME-SET-TRX-STATE.confirm primitive with a status of either SUCCESS or TX_ON, the data request command frame is then transmitted by issuing the PD-DATA.request primitive. Finally, on receipt of the PD-DATA.confirm primitive, the MLME enables the receiver by issuing the PLME-SET-TRX-STATE.request primitive with a state of RX_ON to the PHY in preparation for the acknowledgment.

If the MLME successfully transmits a data request command, the MLME will expect an acknowledgment in return. If this does not occur, the data request command frame will be retried. If an acknowledgment is not received after *aMaxFrameRetries* attempts, the MLME will issue the MLME-POLL.confirm primitive with a status of NO_ACK.

If an acknowledgment is received, the MLME will request that the PHY enable its receiver if the frame pending subfield of the acknowledgment frame is set to 1. If the frame pending subfield of the acknowledgment frame is set to 0, the MLME will issue the MLME-POLL.confirm primitive with a status of NO_DATA.

If a frame is received from the coordinator with a zero length payload or if the frame is a MAC command frame, the MLME will issue the MLME-POLL.confirm primitive with a status of NO_DATA. If a frame is received from the coordinator with nonzero length payload, the MLME will issue the MLME-POLL.confirm primitive with a status of SUCCESS. In this case, the actual data are indicated to the next higher layer using the MCPS-DATA.indication primitive (see 7.1.1.3).

If a frame is not received within *aMaxFrameResponseTime* CAP symbols in a beacon-enabled PAN, or symbols in a nonbeacon-enabled PAN, even though the acknowledgment to the data request command has its frame pending subfield set to 1, the MLME will issue the MLME-POLL.confirm primitive with a status of NO_DATA.

If any parameter in the MLME-POLL.request primitive is not supported or is out of range, the MLME will issue the MLME-POLL.confirm primitive with a status of INVALID_PARAMETER.

7.1.16.2 MLME-POLL.confirm

The MLME-POLL.confirm primitive reports the results of a request to poll the coordinator for data.

7.1.16.2.1 Semantics of the service primitive

The semantics of the MLME-POLL.confirm primitive is as follows:

```
MLME-POLL.confirm          (
                           status
                           )
```

Table 63 specifies the parameters for the MLME-POLL.confirm primitive.

Table 63—MLME-POLL.confirm parameters

Name	Type	Valid range	Description
status	Integer	SUCCESS, CHANNEL_ACCESS_FAILURE, NO_ACK, NO_DATA, UNAVAILABLE_KEY, FAILED_SECURITY_CHECK, or INVALID_PARAMETER	The status of the data request.

7.1.16.2.2 When generated

The MLME-POLL.confirm primitive is generated by the MLME and issued to its next higher layer in response to an MLME-POLL.request primitive. If the request was successful, the status parameter will be equal to SUCCESS, indicating a successful poll for data. Otherwise, the status parameter indicates an error code of CHANNEL_ACCESS_FAILURE, NO_ACK, NO_DATA, UNAVAILABLE_KEY, FAILED_SECURITY_CHECK, or INVALID_PARAMETER. The reasons for these status values are fully described in 7.1.16.1.3.

7.1.16.2.3 Effect on receipt

On receipt of the MLME-POLL.confirm primitive, the next higher layer is notified of the status of the procedure to request data from the coordinator.

7.1.16.3 Message sequence chart for requesting data from a coordinator

Figure 33 illustrates the sequence of messages necessary for a device to request data from a coordinator. In both cases, a poll request is issued to the MLME, which then sends a data request command to the coordinator. In a) Figure 33, the corresponding acknowledgment has the frame pending (FP) subfield set to 0 and the MLME issues the poll request confirmation immediately. In b) Figure 33, the corresponding acknowledgment has the frame pending subfield set to 1 and the MLME enables the receiver in anticipation of the data frame from the coordinator. On receipt of this data frame, the MLME issues a poll request confirmation followed by a data indication containing the data of the received frame.

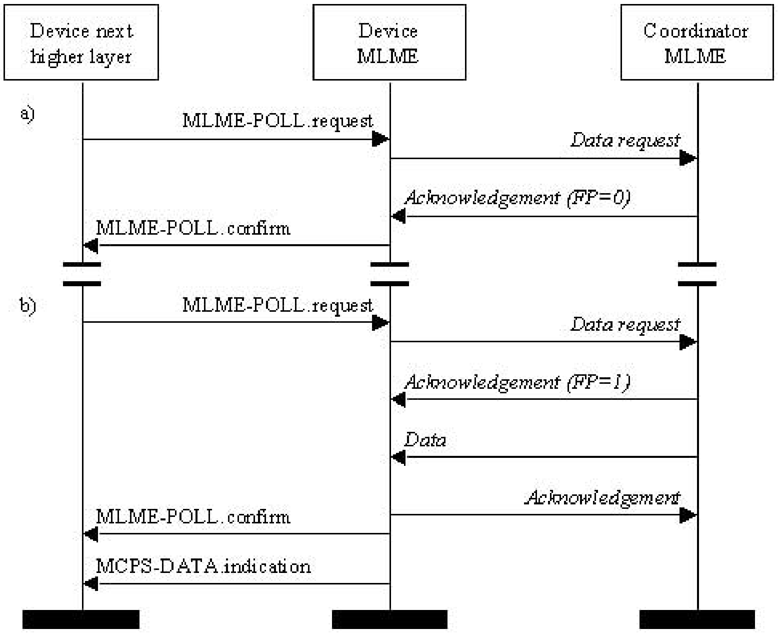


Figure 33—Message sequence chart for requesting data from the coordinator

7.1.17 MAC enumeration description

This subclause explains the meaning of the enumerations used in the primitives defined in the MAC sublayer specification. Table 64 shows a description of the MAC enumeration values.

Table 64—MAC enumerations description

Enumeration	Value	Description
SUCCESS	0 x 00	The requested operation was completed successfully. For a transmission request, this value indicates a successful transmission.
—	0 x 01—0 x df	Reserved for MAC command status and reason code values.
—	0 x 80—0 x df	Reserved.
BEACON_LOSS	0 x e0	The beacon was lost following a synchronization request.
CHANNEL_ACCESS_FAILURE	0 x e1	A transmission could not take place due to activity on the channel, i.e., the CSMA-CA mechanism has failed.
DENIED	0 x e2	The GTS request has been denied by the PAN coordinator.
DISABLE_TRX_FAILURE	0 x e3	The attempt to disable the transceiver has failed.
FAILED_SECURITY_CHECK	0 x e4	The received frame induces a failed security check according to the security suite.
FRAME_TOO_LONG	0 x e5	The frame resulting from secure processing has a length that is greater than *aMACMaxFrameSize.*

Table 64—MAC enumerations description *(continued)*

Enumeration	Value	Description
INVALID_GTS	0 x e6	The requested GTS transmission failed because the specified GTS either did not have a transmit GTS direction or was not defined.
INVALID_HANDLE	0 x e7	A request to purge an MSDU from the transaction queue was made using an MSDU handle that was not found in the transaction table.
INVALID_PARAMETER	0 x e8	A parameter in the primitive is out of the valid range.
NO_ACK	0 x e9	No acknowledgment was received after *aMaxFrameRetries*.
NO_BEACON	0 x ea	A scan operation failed to find any network beacons.
NO_DATA	0 x eb	No response data were available following a request.
NO_SHORT_ADDRESS	0 x ec	The operation failed because a short address was not allocated.
OUT_OF_CAP	0 x ed	A receiver enable request was unsuccessful because it could not be completed within the CAP.
PAN_ID_CONFLICT	0 x ee	A PAN identifier conflict has been detected and communicated to the PAN coordinator.
REALIGNMENT	0 x ef	A coordinator realignment command has been received.
TRANSACTION_EXPIRED	0 x f0	The transaction has expired and its information discarded.
TRANSACTION_OVERFLOW	0 x f1	There is no capacity to store the transaction.
TX_ACTIVE	0 x f2	The transceiver was in the transmitter enabled state when the receiver was requested to be enabled.
UNAVAILABLE_KEY	0 x f3	The appropriate key is not available in the ACL.
UNSUPPORTED_ATTRIBUTE	0 x f4	A SET/GET request was issued with the identifier of a PIB attribute that is not supported.
—	0 x f5–0 x ff	Reserved.

7.2 MAC frame formats

This subclause specifies the format of the MAC frame (MPDU). Each MAC frame consists of the following basic components:

a) A MHR, which comprises frame control, sequence number, and address information.
b) A MAC payload, of variable length, which contains information specific to the frame type. Acknowledgment frames do not contain a payload.
c) A MFR, which contains a FCS.

The frames in the MAC sublayer are described as a sequence of fields in a specific order. All frame formats in this subclause are depicted in the order in which they are transmitted by the PHY, from left to right, where the leftmost bit is transmitted first in time. Bits within each field are numbered from 0 (leftmost and least significant) to $k - 1$ (rightmost and most significant), where the length of the field is k bits. Fields that are longer than a single octet are sent to the PHY in the order from the octet containing the lowest numbered bits to the octet containing the highest numbered bits.

7.2.1 General MAC frame format

The MAC frame format is composed of a MHR, a MAC payload, and a MFR. The fields of the MHR appear in a fixed order, however, the addressing fields may not be included in all frames. The general MAC frame shall be formatted as illustrated in Figure 34.

Octets: 2	1	0/2	0/2/8	0/2	0/2/8	variable	2
Frame control	Sequence number	Destination PAN identifier	Destination address	Source PAN identifier	Source address	Frame payload	FCS
		Addressing fields					
MHR						MAC payload	MFR

Figure 34—General MAC frame format

7.2.1.1 Frame control field

The frame control field is 16 bits in length and contains information defining the frame type, addressing fields, and other control flags. The frame control field shall be formatted as illustrated in Figure 35.

Bits: 0–2	3	4	5	6	7–9	10–11	12–13	14–15
Frame type	Security enabled	Frame pending	Ack. request	Intra-PAN	Reserved	Dest. addressing mode	Reserved	Source addressing mode

Figure 35—Format of the frame control field

7.2.1.1.1 Frame type subfield

The frame type subfield is 3 bits in length and shall be set to one of the nonreserved values listed in Table 65.

Table 65—Values of the frame type subfield

Frame type value $b_2\ b_1\ b_0$	Description
000	Beacon
001	Data
010	Acknowledgment
011	MAC command
100─111	Reserved

7.2.1.1.2 Security enabled subfield

The security enabled subfield is 1 bit in length and shall be set to 0 if the frame is not cryptographically protected by the MAC sublayer. If the security enabled subfield is set to 1, the frame shall be protected using the keys stored in the MAC PIB for the security relationship indicated by the current frame. The cryptographic operations used to protect the frame are defined by the security suite selected for that security relationship. If no security suite is defined for that relationship, the security enabled subfield shall be set to 0.

7.2.1.1.3 Frame pending subfield

The frame pending subfield is 1 bit in length and shall be set to 1 if the device sending the frame has additional data to send to the recipient following the current transfer. If more data are pending, the recipient shall retrieve them by sending another data request command to the device. If the device sending the frame does not have any more data for the recipient, this subfield shall be set to 0.

The frame pending subfield shall be used only in frames transmitted either during the CAP by devices operating on a beacon-enabled PAN or at any time by devices operating on a nonbeacon-enabled PAN.

At all other times it shall be set to 0 on transmission and ignored on reception.

7.2.1.1.4 Acknowledgment request subfield

The acknowledgment request subfield is 1 bit in length and specifies whether an acknowledgment is required from the recipient device on receipt of a data or MAC command frame. If this subfield is set to 1, the recipient device shall send an acknowledgment frame after determining that the frame is valid (see 7.5.6.2 for a list of requirements necessary for frame validity). If this subfield is set to 0, the recipient device shall not send an acknowledgment frame after determining that the frame is valid.

7.2.1.1.5 Intra-PAN subfield

The intra-PAN subfield is 1 bit in length and specifies whether the MAC frame is to be sent within the same PAN (intra-PAN) or to another PAN (inter-PAN). If this subfield is set to 1 and both destination and source addresses are present, the frame shall not contain the source PAN identifier field. If this subfield is set to 0 and both destination and source addresses are present, the frame shall contain both destination and source PAN identifier fields.

7.2.1.1.6 Destination addressing mode subfield

The destination addressing mode subfield is 2 bits in length and shall be set to one of the values listed in Table 66.

If this subfield is equal to 0 and the frame type subfield does not specify that this frame is an acknowledgment or beacon frame, the source addressing mode subfield shall be nonzero, implying that the frame is directed to the PAN coordinator with the PAN identifier as specified in the source PAN identifier field.

7.2.1.1.7 Source addressing mode subfield

The source addressing mode subfield is 2 bits in length and shall be set to one of the values listed in Table 66.

If this subfield is equal to 0 and the frame type subfield does not specify that this frame is an acknowledgment frame, the destination addressing mode subfield shall be nonzero, implying that the frame has originated from the PAN coordinator with the PAN identifier as specified in the destination PAN identifier field.

Table 66—Possible values of the destination and source addressing mode subfields

Addressing mode value b_1 b_0	Description
00	PAN identifier and address field are not present.
01	Reserved.
10	Address field contains a 16 bit short address.
11	Address field contains a 64 bit extended address.

7.2.1.2 Sequence number field

The sequence number field is 8 bits in length and specifies a unique sequence identifier for the frame.

For a beacon frame, the sequence number field shall specify a BSN. Each coordinator shall store its current BSN value in the MAC PIB attribute *macBSN* and initialize it to a random value. The algorithm for choosing a random number is out of the scope of this standard. The coordinator shall copy the value of the *macBSN* attribute into the sequence number field of a beacon frame, each time one is generated, and shall then increment *macBSN* by one.

For a data, acknowledgment, or MAC command frame, the sequence number field shall specify a data sequence number (DSN) that is used to match an acknowledgment frame to the data or MAC command frame. Each device shall support exactly one DSN regardless of the number of unique devices with which it wishes to communicate. Each device shall store its current DSN value in the MAC PIB attribute *macDSN* and initialize it to a random value. The algorithm for choosing a random number is out of the scope of this standard. The device shall copy the value of the *macDSN* attribute into the sequence number field of a data or MAC command frame, each time one is generated, and shall then increment *macDSN* by one.

If an acknowledgment is requested, the recipient device shall copy the DSN received in the data or MAC command frame into the DSN field of the corresponding acknowledgment frame. If the acknowledgment was not received after *macAckWaitDuration* symbols, the MAC sublayer of the originating device shall retransmit the frame using the same DSN as was used in the original transmission.

7.2.1.3 Destination PAN identifier field

The destination PAN identifier field is 16 bits in length and specifies the unique PAN identifier of the intended recipient of the frame. A value of 0 x ffff in this field shall represent the broadcast PAN identifier, which shall be accepted as a valid PAN identifier by all devices currently listening to the channel.

This field shall be included in the MAC frame only if the destination addressing mode subfield of the frame control field is nonzero.

7.2.1.4 Destination address field

The destination address field is either 16 bits or 64 bits in length, according to the value specified in the destination addressing mode subfield of the frame control field (see 7.2.1.1.6), and specifies the address of the intended recipient of the frame. A 16 bit value of 0 x ffff in this field shall represent the broadcast short address, which shall be accepted as a valid short address by all devices currently listening to the channel.

This field shall be included in the MAC frame only if the destination addressing mode subfield of the frame control field is nonzero.

7.2.1.5 Source PAN identifier field

The source PAN identifier field is 16 bits in length and specifies the unique PAN identifier of the originator of the frame. This field shall be included in the MAC frame only if the source addressing mode and intra-PAN subfields of the frame control field are nonzero and equal to zero, respectively.

The PAN identifier of a device is initially determined during association on a PAN, but may change following a PAN identifier conflict resolution (see 7.5.2.2).

7.2.1.6 Source address field

The source address field is either 16 bits or 64 bits in length, according to the value specified in the destination addressing mode subfield of the frame control field (see 7.2.1.1.7), and specifies the address of the originator of the frame. This field shall be included in the MAC frame only if the source addressing mode subfield of the frame control field is nonzero.

7.2.1.7 Frame payload field

The frame payload field has a variable length and contains information specific to individual frame types. If the security enabled subfield is set to 1 in the frame control field, the frame payload is protected as defined by the security suite selected for that relationship.

7.2.1.8 FCS field

The FCS field is 16 bits in length and contains a 16 bit ITU-T CRC. The FCS is calculated over the MHR and MAC payload parts of the frame.

The FCS shall be calculated using the following standard generator polynomial of degree 16:

$$G_{16}(x) = x^{16} + x^{12} + x^{5} + 1 \tag{7}$$

The FCS shall be calculated for transmission using the following algorithm:

- Let $M(x) = b_0x^{k-1} + b_1x^{k-2} + \ldots + b_{k-2}x + b_{k-1}$ be the polynomial representing the sequence of bits for which the checksum is to be computed.
- Multiply $M(x)$ by x^{16}, giving the polynomial $x^{16} \times M(x)$.
- Divide $x^{16} \times M(x)$ modulo 2 by the generator polynomial, $G_{16}(x)$, to obtain the remainder polynomial, $R(x) = r_0x^{15} + r_1x^{14} + \ldots + r_{14}x + r_{15}$.
- The FCS field is given by the remainder coefficients.

As an example, consider an acknowledgment frame with no payload and the following 3 byte MHR:

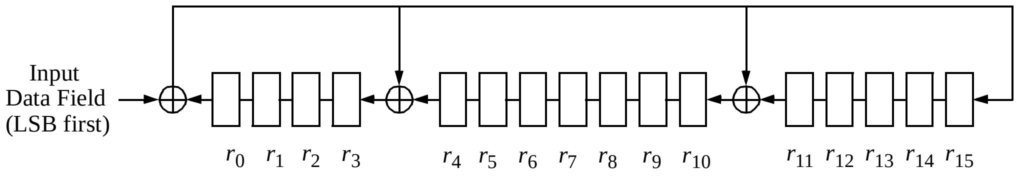
0100 0000 0000 0000 0101 0110 [leftmost bit (b_0) transmitted first in time]
b_0..b_{23}

The FCS for this case would be the following:

0010 0111 1001 1110 [leftmost bit (r_0) transmitted first in time]
r_0..............................r_{15}

A typical implementation is depicted in Figure 36.

CRC-16 Generator Polynomial: $G(x) = x^{16} + x^{12} + x^{5} + 1$


Input
Data Field
(LSB first)
r_0 r_1 r_2 r_3
r_4 r_5 r_6 r_7 r_8 r_9 r_{10}
r_{11} r_{12} r_{13} r_{14} r_{15}

1. Initialize the remainder register (r_0 through r_{15}) to zero.
2. Shift MHR and payload into the divider in the order of transmission (LSB first).
3. After the last bit of the data field is shifted into the divider, the remainder register contains the FCS.
4. The FCS is appended to the data field so that r_0 is transmitted first.

Figure 36—Typical FCS implementation

7.2.2 Format of individual frame types

Four frame types are defined: beacon, data, acknowledgment, and MAC command. These frame types are discussed in 7.2.2.1 through 7.2.2.4.

7.2.2.1 Beacon frame format

The beacon frame shall be formatted as illustrated in Figure 37.

Octets: 2	**1**	**4/10**	**2**	**variable**	**variable**	**variable**	**2**
Frame control	Sequence number	Addressing fields	Superframe specification	GTS fields (Figure 38)	Pending address fields (Figure 39)	Beacon payload	FCS
MHR			MAC payload				MFR

Figure 37—Beacon frame format

The GTS fields shall be formatted as illustrated in Figure 38, and the pending address fields shall be formatted as illustrated in Figure 39.

The order of the fields of the beacon frame shall conform to the order of the general MAC frame as illustrated in Figure 34.

Octets: 1	0/1	variable
GTS specification	GTS directions	GTS list

Figure 38—Format of the GTS information fields

Octets: 1	variable
Pending address specification	Address list

Figure 39—Format of the pending address information fields

7.2.2.1.1 Beacon frame MHR fields

The MHR for a beacon frame shall contain the frame control field, the sequence number field, the source PAN identifier field, and the source address field.

In the frame control field, the frame type field shall contain the value that indicates a beacon frame, as shown in Table 65, and the source addressing mode subfield shall be set as appropriate for the address of the coordinator transmitting the beacon frame. If security is used for the beacon, the security enabled subfield shall be set to 1. All other fields shall be set to 0 and ignored on reception.

The sequence number field shall contain the current value of *macBSN*.

The addressing fields shall comprise only the source address fields. The source PAN identifier and source address fields shall contain the PAN identifier and address, respectively, of the device transmitting the beacon.

7.2.2.1.2 Superframe specification field

The superframe specification field is 16 bits in length and shall be formatted as illustrated in Figure 40.

Bits: 0-3	4-7	8-11	12	13	14	15
Beacon order	Superframe order	Final CAP slot	Battery life extension	Reserved	PAN coordinator	Association permit

Figure 40—Format of the superframe specification field

The beacon order subfield is 4 bits in length and shall specify the transmission interval of the beacon. If *BO* is the value of the beacon order, the beacon interval, *BI*, shall be computed as follows: $BI = aBaseSuperframeDuration * 2^{BO}$ symbols, where $0 \le BO \le 14$. If $BO = 15$, the coordinator shall not transmit beacon frames except when requested to do so, such as on receipt of a beacon request command.

The superframe order subfield is 4 bits in length and shall specify the length of time during which the superframe is active (i.e., receiver enabled), including the beacon frame transmission time. The coordinator shall interact with its PAN only during the active superframe. If *SO* is the value of the superframe order, the superframe duration, *SD*, shall be computed as follows. For $0 \le SO \le BO \le 14$, $SD = aBaseSuperframeDuration * 2^{SO}$ symbols. If *SO* = 15, the superframe shall not be active following the transmission of the beacon.

The final CAP slot subfield is four bits in length and specifies the final superframe slot utilized by the CAP. The duration of the CAP, as implied by this subfield, shall be greater than or equal to the value specified by *aMinCAPLength*. However, an exception is allowed for the accommodation of the temporary increase in the beacon frame length needed to perform GTS maintenance (see 7.2.2.1.3).

The battery life extension subfield is 1 bit in length and shall be set to 1 if frames transmitted to the beaconing device during the CAP are required to start in or before the sixth full backoff period following the beacon's IFS period. Otherwise, the battery life extension subfield shall be set to 0.

The PAN coordinator subfield is 1 bit in length and shall be set to 1 if the beacon frame is being transmitted by the PAN coordinator. Otherwise, the PAN coordinator subfield shall be set to 0.

The association permit subfield is 1 bit in length and shall be set to 1 if *macAssociationPermit* is set to TRUE (i.e., the coordinator is accepting association on the PAN). The association permit bit shall be set to 0 if the coordinator is currently not accepting association requests on its network.

7.2.2.1.3 GTS specification field

The GTS specification field is 8 bits in length and shall be formatted as illustrated in Figure 41.

Bits: 0-2	**3-6**	**7**
GTS descriptor count	Reserved	GTS permit

Figure 41—Format of the GTS specification field

The GTS descriptor count subfield is 3 bits in length and specifies the number of 3 octet GTS descriptors contained in the GTS list field of the beacon frame. If the value of this subfield is greater than zero, the size of the CAP shall be allowed to dip below *aMinCAPLength* to accommodate the temporary increase in the beacon frame length caused by the inclusion of the subfield. If the value of this subfield is zero, the GTS directions field and GTS list field of the beacon frame are not present.

The GTS permit subfield is 1 bit in length and shall be set to 1 if *macGTSPermit* is equal to TRUE (i.e., the PAN coordinator is accepting GTS requests). Otherwise, the GTS permit field shall be set to 0.

7.2.2.1.4 GTS directions field

The GTS directions field is 8 bits in length and shall be formatted as illustrated in Figure 42.

Bits: 0-6	**7**
GTS directions mask	Reserved

Figure 42—Format of the GTS directions field

The GTS directions mask subfield is 7 bits in length and contains a mask identifying the directions of the GTSs in the superframe. The lowest bit in the mask corresponds to the direction of the first GTS contained in the GTS list field of the beacon frame, with the remainder appearing in the order that they appear in the list. Each bit shall be set to 1 if the GTS is a receive only GTS or to 0 if the GTS is a transmit-only GTS. GTS direction is defined relative to the direction of the data frame transmission by the device.

7.2.2.1.5 GTS list field

The size of the GTS list field is defined by the values specified in the GTS specification field of the beacon frame and contains the list of GTS descriptors that represents the GTSs that are being maintained. The maximum number of GTS descriptors shall be limited to seven.

Each GTS descriptor is 24 bits in length and shall be formatted as illustrated in Figure 43.

Bits: 0-15	16-19	20-23
Device short address	GTS starting slot	GTS length

Figure 43—Format of the GTS descriptor

The device short address subfield is 16 bits in length and shall contain the short address of the device for which the GTS descriptor is intended.

The GTS starting slot subfield is 4 bits in length and contains the superframe slot at which the GTS is to begin.

The GTS length subfield is 4 bits in length and contains the number of contiguous superframe slots over which the GTS is active.

7.2.2.1.6 Pending address specification field

The pending address specification field shall be formatted as illustrated in Figure 44.

Bits: 0–2	3	4–6	7
Number of short addresses pending	Reserved	Number of extended addresses pending	Reserved

Figure 44—Format of the pending address specification field

The number of short addresses pending subfield is 3 bits in length and indicates the number of short addresses contained in the address list field of the beacon frame.

The number of extended addresses pending subfield is 3 bits in length and indicates the number of 64 bit extended addresses contained in the address list field of the beacon frame.

7.2.2.1.7 Address list field

The size of the address list field is determined by the values specified in the pending address specification field of the beacon frame and contains the list of addresses of the devices that currently have messages pending with the coordinator. The address list shall not contain the broadcast short address 0 x ffff.

The maximum number of addresses pending shall be limited to seven and may comprise both short and extended addresses. All pending short addresses shall appear first in the list followed by any extended addresses. If the coordinator is able to store more than seven transactions, it shall indicate them in its beacon on a first-come-first-served basis, ensuring that the beacon frame contains at most seven addresses.

7.2.2.1.8 Beacon payload field

The beacon payload field is an optional sequence of up to *aMaxBeaconPayloadLength* octets specified to be transmitted in the beacon frame by the next higher layer. If *macBeaconPayloadLength* is nonzero, the set of octets contained in *macBeaconPayload* shall be copied into this field.

If security is required on an outgoing beacon frame, the sequence of octets in the beacon payload field shall be processed according to the security suite corresponding to *aExtendedAddress*.

If the security enabled subfield of the frame control field of an incoming frame is set to 0, the beacon payload field shall contain the intended sequence of octets to be passed to the next higher layer. If the security enabled subfield of the frame control field of an incoming frame is set to 1, the device shall process the beacon payload field according to the security suite corresponding to the source address of the incoming frame in order to determine the intended sequence of octets to be passed to the next higher layer.

If a device receives a beacon with a payload field present, it shall indicate it to the next higher layer and then process the information contained in the superframe specification field and address list field. If the MAC sublayer receives a beacon without a payload field present, it shall immediately interpret and process the information contained in the superframe specification field and address list field.

7.2.2.2 Data frame format

The data frame shall be formatted as illustrated in Figure 45.

Octets: 2	**1**	**(see 7.2.2.2.1)**	**variable**	**2**
Frame control	Sequence number	Addressing fields	Data payload	FCS
MHR			MAC payload	MFR

Figure 45—Data frame format

The order of the fields of the data frame shall conform to the order of the general MAC frame as illustrated in Figure 34.

7.2.2.2.1 Data frame MHR fields

The MHR for a data frame shall contain the frame control field, the sequence number field, the destination PAN identifier/address fields, and/or the source PAN identifier/address fields.

In the frame control field, the frame type subfield shall contain the value that indicates a data frame, as shown in Table 65. All other subfields shall be set appropriately according to the intended use of the data frame.

The sequence number field shall contain the current value of *macDSN*.

The addressing fields shall comprise the destination address fields and/or the source address fields, dependent on the settings in the frame control field.

7.2.2.2.2 Data payload field

The payload of a data frame shall contain the sequence of octets that the next higher layer has requested the MAC sublayer to transmit.

If security is required on an outgoing data frame, the sequence of octets in the data payload field shall be processed according to the security suite corresponding to either the destination address, if present, or *macCoordExtendedAddress*, if the destination address field is not present.

If the security enabled subfield of the frame control field of an incoming frame is set to 0, the data payload field shall contain the intended sequence of octets to be passed to the next higher layer. If the security enabled subfield of the frame control field of an incoming frame is set to 1, the device shall process the data payload field according to its selected security suite in order to determine the intended sequence of octets to be passed to the next higher layer.

7.2.2.3 Acknowledgment frame format

The acknowledgment frame shall be formatted as illustrated in Figure 46.

Octets: 2	**1**	**2**
Frame control	Sequence number	FCS
MHR		MFR

Figure 46—Acknowledgment frame format

The order of the fields of the acknowledgment frame shall conform to the order of the general MAC frame as illustrated in Figure 34.

7.2.2.3.1 Acknowledgment frame MHR fields

The MHR for an acknowledgment frame shall contain only the frame control field and the sequence number field.

In the frame control field, the frame type subfield shall contain the value that indicates an acknowledgment frame, as shown in Table 65. The frame pending subfield shall be set according to whether the device sending the acknowledgment frame has more data pending for the recipient. All other subfields shall be set to 0 and ignored on reception.

The sequence number field shall contain the value of the sequence number received in the frame for which the acknowledgment is to be sent.

7.2.2.4 MAC command frame format

The MAC command frame shall be formatted as illustrated in Figure 47.

Octets: 2	**1**	**(see 7.2.2.4.1)**	**1**	**variable**	**2**
Frame control	Sequence number	Addressing fields	Command frame identifier	Command payload	FCS
MHR			MAC payload		MFR

Figure 47—MAC command frame format

The order of the fields of the MAC command frame shall conform to the order of the general MAC frame as illustrated in Figure 34.

7.2.2.4.1 MAC command frame MHR fields

The MHR for a MAC command frame shall contain the frame control field, the sequence number field, the destination PAN identifier/address fields, and/or the source PAN identifier/address fields.

In the frame control field, the frame type subfield shall contain the value that indicates a MAC command frame, as shown in Table 65. All other subfields shall be set appropriately according to the intended use of the MAC command frame.

The sequence number field shall contain the current value of *macDSN*.

The addressing fields shall comprise the destination address fields and/or the source address fields, dependent on the settings in the frame control field.

7.2.2.4.2 Command frame identifier field

The command frame identifier field identifies the MAC command being used. This field shall be set to one of the nonreserved values listed in Table 67.

7.2.2.4.3 Command payload field

The command payload field contains the MAC command itself.

If security is required on an outgoing MAC command frame, the sequence of octets in the command payload field shall be processed according to the security suite corresponding to either the destination address, if present, or *macCoordExtendedAddress*, if the destination address field is not present.

If the security enabled subfield of the frame control field of an incoming frame is set to 0, the command payload field shall contain the intended MAC command. If the security enabled subfield of the frame control field of an incoming frame is set to 1, the device shall process the command payload field according to its selected security suite in order to determine the intended MAC command.

The formats of the individual commands are described in 7.3.

7.3 MAC command frames

The command frames defined by the MAC sublayer are listed in Table 67. An FFD shall be capable of transmitting and receiving all command frame types, while the requirements for an RFD are indicated in the table. MAC commands shall only be transmitted in the CAP for beacon-enabled PANs or at any time for nonbeacon-enabled PANs.

How the MLME shall construct the individual commands for transmission is detailed in 7.3.1 through 7.3.3. MAC command reception shall abide by the procedure described in 7.5.6.2.

Table 67—MAC command frames

Command frame identifier	Command name	RFD		Subclause
		Tx	Rx	
0 x 01	Association request	X		7.3.1.1
0 x 02	Association response		X	7.3.1.2
0 x 03	Disassociation notification	X	X	7.3.1.3
0 x 04	Data request	X		7.3.2.1
0 x 05	PAN ID conflict notification	X		7.3.2.2
0 x 06	Orphan notification	X		7.3.2.3
0 x 07	Beacon request			7.3.2.4
0 x 08	Coordinator realignment		X	7.3.2.5
0 x 09	GTS request			7.3.3.1
0 x 0a–0 x ff	Reserved			—

7.3.1 Association and disassociation

The set of association and dissassociation commands is used to allow devices to associate with or disassociate from a PAN.

7.3.1.1 Association request command

The association request command allows a device to request association with a coordinator.

This command shall only be sent by an unassociated device that wishes to associate with a PAN. A device shall only associate with a PAN allowing association, as determined through the scan procedure.

All devices shall be capable of transmitting this command, although an RFD is not required to be capable of receiving it.

The association request command shall be formatted as illustrated in Figure 48.

octets: 17/23	**1**	**1**
MHR fields	Command frame identifier (see Table 67)	Capability information

Figure 48—Association request command format

7.3.1.1.1 MHR fields

The fields of the MHR of the general MAC frame format (see Figure 34) shall be specified as indicated in this subclause.

The source addressing mode subfield of the frame control field shall be set to 3 (64 bit extended addressing). The destination addressing mode subfield shall be set to the same mode as indicated in the beacon frame to which the association request command refers.

If security is used for the association request command, the security enabled subfield shall be set to 1, and the frame shall be processed according to the method defined by the security suite corresponding to the destination address. Otherwise, the security enabled subfield shall be set to 0.

The frame pending subfield of the frame control field shall be set to 0 and ignored upon reception, and the acknowledgment request subfield shall be set to 1.

The destination PAN identifier field shall contain the identifier of the PAN to which to associate. The destination address field shall contain the address from the beacon frame that was transmitted by the coordinator to which the association request command is being sent. The source PAN identifier field shall contain the broadcast PAN identifier (i.e., 0 x ffff). The source address field shall contain the value of *aExtendedAddress*.

7.3.1.1.2 Capability information field

The capability information field shall be formatted as illustrated in Figure 49.

bits: 0	**1**	**2**	**3**	**4-5**	**6**	**7**
Alternate PAN coordinator	Device type	Power source	Receiver on when idle	Reserved	Security capability	Allocate address

Figure 49—Capability information field format

The alternate PAN coordinator subfield is 1 bit in length and shall be set to 1 if the device is capable of becoming a PAN coordinator. Otherwise, the alternate PAN coordinator subfield shall be set to 0.

The device type subfield is 1 bit in length and shall be set to 1 if the device is an FFD. Otherwise, the device type subfield shall be set to 0 to indicate an RFD.

The power source subfield is 1 bit in length and shall be set to 1 if the device is receiving power from the alternating current mains. Otherwise, the power source subfield shall be set to 0.

The receiver on when idle subfield is 1 bit in length and shall be set to 1 if the device does not disable its receiver to conserve power during idle periods. Otherwise, the receiver on when idle subfield shall be set to 0.

The security capability subfield is 1 bit in length and shall be set to 1 if the device is capable of sending and receiving MAC frames secured using the security suite specified in 7.6. Otherwise the security capability subfield shall be set to 0.

The allocate address subfield is one bit in length and shall be set to 1 if the device wishes the coordinator to allocate a short address as a result of the association procedure. If this subfield is set to 0, the special short address of 0 x fffe shall be allocated to the device and returned through the association response command. In this case, the device shall communicate on the PAN using only its 64 bit extended address.

7.3.1.2 Association response command

The association response command allows the coordinator to communicate the results of an association attempt back to the device requesting association.

This command shall only be sent by the coordinator to a device that is currently trying to associate.

All devices shall be capable of receiving this command, although an RFD is not required to be capable of transmitting it.

The association response command shall be formatted as illustrated in Figure 50.

octets: 23	**1**	**2**	**1**
MHR fields	Command frame identifier (see Table 67)	Short address	Association status

Figure 50—Association response command format

7.3.1.2.1 MHR fields

The fields of the MHR of the general MAC frame format (see Figure 34) shall be specified as indicated in this subclause.

The destination addressing mode and source addressing mode subfields of the frame control field shall each be set to 3 (i.e., 64 bit extended addressing).

If security is used for the association response command, the security enabled subfield shall be set to 1, and the frame shall be processed according to the method defined by the security suite corresponding to the destination address. Otherwise, the security enabled subfield shall be set to 0.

The frame pending subfield of the frame control field shall be set to 0 and ignored upon reception, and the acknowledgment request subfield shall be set to 1.

The destination and source PAN identifier fields shall contain the value of *macPANId*. The destination address field shall contain the extended address of the device requesting association. The source address field shall contain the value of *aExtendedAddress*.

7.3.1.2.2 Short address field

The short address field is 16 bits in length.

If the coordinator was not able to associate this device to its PAN, this field shall be set to 0 x ffff, and the association status field shall contain the reason for the failure. If the coordinator was able to associate the device to its PAN, this field shall contain the short address that the device may use in its communications on the PAN until it is disassociated.

A short address field value equal to 0 x fffe shall indicate that the device has been successfully associated with a PAN, but has not been allocated a short address. In this case, the device shall communicate on the PAN using only its 64 bit extended address.

7.3.1.2.3 Association status field

The association status field is 8 bits in length and shall contain one of the nonreserved values listed in Table 68.

Table 68—Valid values of the association status field

Association status	Description
0 x 00	Association successful.
0 x 01	PAN at capacity.
0 x 02	PAN access denied.
0 x 03—0 x 7f	Reserved.
0 x 80—0 x ff	Reserved for MAC primitive enumeration values.

7.3.1.3 Disassociation notification command

Either the coordinator or an associated device may send the disassociate notification command.

All devices shall implement this command.

The disassociation notification command shall be formatted as illustrated in Figure 51.

octets: 17	1	1
MHR fields	Command frame identifier (see Table 67)	Disassociation reason

Figure 51—Disassociation notification command format

7.3.1.3.1 MHR fields

The fields of the MHR of the general MAC frame format (see Figure 34) shall be specified as indicated in this subclause.

The destination addressing mode and source addressing mode subfields of the frame control field shall both be set to 3 (i.e., 64 bit extended addressing).

If security is used for the disassociation notification command, the security enabled subfield shall be set to 1, and the frame shall be processed according to the method defined by the security suite corresponding to the destination address. Otherwise, the security enabled subfield shall be set to 0.

The frame pending subfield of the frame control field shall be set to 0 and ignored upon reception, and the acknowledgment request subfield shall be set to 1.

The destination and source PAN identifier fields shall contain the value of *macPANId*. If the coordinator wants an associated device to leave the PAN, then the destination address field shall contain the extended address of the device being removed from the PAN. If an associated device wants to leave the PAN, then the destination address field shall contain the value of *macCoordExtendedAddress*. The source address field shall contain the value of *aExtendedAddress*.

7.3.1.3.2 Disassociation reason field

The disassociation reason field is 8 bits in length and shall contain one of the nonreserved values listed in Table 69.

Table 69—Valid disassociation reason codes

Disassociate reason	Description
0 x 00	Reserved.
0 x01	The coordinator wishes the device to leave the PAN.
0 x 02	The device wishes to leave the PAN.
0 x 03—0 x 7f	Reserved.
0 x 80—0 x ff	Reserved for MAC primitive enumeration values.

7.3.2 Coordinator interaction

The set of coordinator interaction commands is used to allow devices to interact with a coordinator.

7.3.2.1 Data request command

The data request command is sent by a device to request data from a coordinator.

On a beacon-enabled PAN, this command shall be sent by a device when *macAutoRequest* is equal to TRUE and a beacon frame indicating that data are pending for that device is received from its coordinator. The coordinator indicates pending data in its beacon frame by adding the address of the recipient of the data to the address list field. This command shall also be sent when instructed to do so by the next higher layer on reception of the MLME-POLL.request primitive. In addition, a device may send this command to the coordinator *aResponseWaitTime* symbols after the acknowledgment to a request command, such as an association or a GTS request.

All devices shall be capable of transmitting this command, although an RFD is not required to be capable of receiving it.

The data request command shall be formatted as illustrated in Figure 52.

octets: 7/11/13/17	1
MHR fields	Command frame identifier (see Table 67)

Figure 52—Data request command format

7.3.2.1.1 MHR fields

The fields of the MHR of the general MAC frame format (see Figure 34) shall be specified as indicated in this subclause.

The destination addressing mode subfield of the frame control field shall be set to 0 (i.e., destination addressing information not present) if the data request command is to be sent to the PAN coordinator or set otherwise according to the coordinator to which the data request command is directed. The source addressing mode subfield shall be set to 3 (i.e., 64 bit extended address) if the value of *macShortAddress* is equal to either 0 x fffe or 0 x ffff or set to 2 (i.e., 16 bit short addressing) otherwise.

If security is used for the data request command, the security enabled subfield shall be set to 1, and the frame shall be processed according to the method defined by the security suite corresponding to *macCoordExtendedAddress*. Otherwise, the security enabled subfield shall be set to 0.

The frame pending subfield of the frame control field shall be set to 0 and ignored upon reception, and the acknowledgment request subfield shall be set to 1.

If the destination addressing mode subfield of the frame control field is set to 2, the destination PAN identifier and destination address fields shall contain the value of *macPANId* and *macCoordShortAddress*, respectively. The source PAN identifier field shall contain the value of *macPANId*. The source address field shall contain the value of *aExtendedAddress* if the value of *macShortAddress* is equal to 0 x fffe. Otherwise, the source address field shall be set to the value of *macShortAddress*.

7.3.2.2 PAN ID conflict notification command

The PAN ID conflict notification command is sent by a device to the PAN coordinator when a PAN identifier conflict is detected.

All devices shall be capable of transmitting this command, although an RFD is not required to be capable of receiving it.

The PAN ID conflict notification command shall be formatted as illustrated in Figure 53.

octets: 23	1
MHR fields	Command frame identifier (see Table 67)

Figure 53—PAN ID conflict notification command format

7.3.2.2.1 MHR fields

The fields of the MHR of the general MAC frame format (see Figure 34) shall be specified as indicated in this subclause.

The destination addressing mode and source addressing mode subfields of the frame control field shall both be set to 3 (i.e., 64 bit extended addressing).

The frame shall be processed for security by the sender according to the method defined by the security suite corresponding to *macCoordExtendedAddress.* If the security suite identifier is 0 x 00, the security enabled subfield of the frame control field shall be set to 0. Otherwise, the security enabled subfield shall be set to 1.

The frame pending subfield of the frame control field shall be set to 0 and ignored upon reception, and the acknowledgment request subfield shall be set to 1.

The destination PAN identifier field and source PAN identifier field shall each contain the value of *macPANId*. The destination address field shall contain the value of *macCoordExtendedAddress*. The source address field shall contain the value of *aExtendedAddress*.

7.3.2.3 Orphan notification command

The orphan notification command is used by an associated device that has lost synchronization with its coordinator.

All devices shall be capable of transmitting this command, although an RFD is not required to be capable of receiving it.

The orphan notification command shall be formatted as illustrated in Figure 54.

octets: 17	1
MHR fields	Command frame identifier (see Table 67)

Figure 54—Orphan notification command format

7.3.2.3.1 MHR fields

The fields of the MHR of the general MAC frame format (see Figure 34) shall be specified as indicated in this subclause.

The source addressing mode subfield of the frame control field shall be set to 3 (i.e., 64 bit extended addressing). The destination addressing mode subfield shall be set to 2 (i.e., 16 bit short addressing).

The frame shall be processed for security by the sender according to the method defined by the security suite corresponding to *macCoordExtendedAddress*. If the security suite identifier is 0x00, the security enabled subfield of the frame control field shall be set to 0. Otherwise, the security enabled subfield shall be set to 1.

The frame pending subfield and acknowledgment request subfield of the frame control field shall be set to 0 and ignored upon reception.

The destination PAN identifier field and source PAN identifier field shall each contain the broadcast PAN identifier (0 x ffff). The destination address field shall contain the broadcast short address (i.e., 0 x ffff). The source address field shall contain the value of *aExtendedAddress*.

7.3.2.4 Beacon request command

The beacon request command is used by a device to locate all coordinators within its POS during an active scan.

This command is optional for an RFD.

The beacon request command shall be formatted as illustrated in Figure 55.

octets: 7	**1**
MHR fields	Command frame identifier (see Table 67)

Figure 55—Beacon request command format

7.3.2.4.1 MHR fields

The fields of the MHR of the general MAC frame format (see Figure 34) shall be specified as indicated in this subclause.

The destination addressing mode subfield of the frame control field shall be set to 2 (i.e., 16 bit short addressing), and the source addressing mode subfield shall be set to 0 (i.e., source addressing information not present).

The frame pending subfield of the frame control field shall be set to 0 and ignored upon reception, and the acknowledgment request subfield shall also be set to 0.

The destination PAN identifier field shall contain the broadcast PAN identifier (i.e., 0 x ffff). The destination address field shall contain the broadcast short address (i.e., 0 x ffff).

7.3.2.5 Coordinator realignment command

The coordinator realignment command is sent by a coordinator either following the reception of an orphan notification command from a device that is recognized to be on its PAN or when any of its PAN configuration attributes change.

If this command is sent following the reception of an orphan notification command, it is sent directly to the orphaned device. If this command is sent when any PAN configuration attributes (e.g., PAN identifier or logical channel) change, it is broadcast to the PAN as a courtesy to any devices currently able to receive.

All devices shall be capable of receiving this command, although an RFD is not required to be capable of transmitting it.

The coordinator realignment command shall be formatted as illustrated in Figure 56.

octets: 17/23	**1**	**2**	**2**	**1**	**2**
MHR fields	Command frame identifier (see Table 67)	PAN identifier	Coordinator short address	Logical channel	Short address

Figure 56—Coordinator realignment command format

7.3.2.5.1 MHR fields

The fields of the MHR of the general MAC frame format (see Figure 34) shall be specified as indicated in this subclause.

The destination addressing mode subfield of the frame control field shall be set to 3 (e.g., 64 bit extended addressing) if the command is directed to an orphaned device or set to 2 (e.g., 16 bit short addressing) if it is to be broadcast to the PAN. The source addressing mode subfield of the frame control field shall be set to 3 (e.g., 64 bit extended addressing).

If security is used for the coordinator realignment command directed to an orphaned device, the security enabled subfield of the frame control field shall be set to 1, and the frame shall be processed according to the method defined by the security suite corresponding to the destination address. Otherwise, the security enabled subfield shall be set to 0.

The frame pending subfield of the frame control field shall be set to 0 and ignored upon reception.

The acknowledgment request subfield of the frame control field shall be set to 1 if the command is directed to an orphaned device or set to 0 if the command is to be broadcast to the PAN.

The destination PAN identifier field shall contain the broadcast PAN identifier (e.g., 0 x ffff). The destination address field shall contain the extended address of the orphaned device if the command is directed to an orphaned device. Otherwise, the destination address field shall contain the broadcast short address (e.g., 0 x ffff). The source PAN identifier field shall contain the value of *macPANId*, and the source address field shall contain the value of *aExtendedAddress*.

7.3.2.5.2 PAN identifier field

The PAN identifier field is 16 bits in length and shall contain the PAN identifier that the coordinator intends to use for all future communications.

7.3.2.5.3 Coordinator short address field

The coordinator short address field is 16 bits in length and shall contain the value of *macShortAddress*.

7.3.2.5.4 Logical channel field

The logical channel field is 8 bits in length and shall contain the logical channel that the coordinator intends to use for all future communications.

7.3.2.5.5 Short address field

The short address field is 16 bits in length. If the coordinator realignment command is broadcast to the PAN, this field shall be set to 0 x ffff and ignored on reception.

If the coordinator realignment command is sent directly to an orphaned device, this field shall contain the short address that the device shall use to operate on the PAN. If the device does not have a short address, because it always uses its extended IEEE address, this field shall contain the value 0 x fffe.

7.3.3 GTS allocation and deallocation

The GTS request command is used to manage GTSs. A device can use this command to request the allocation of a new GTS or the deallocation of an existing GTS.

This command is optional for an RFD.

7.3.3.1 GTS request command

The GTS request command is used by an associated device that is requesting the allocation of a new GTS or the deallocation of an existing GTS from the PAN coordinator. Only devices that have a valid short address shall send this command, i.e., the value of *macShortAddress* is not equal to 0 x fffe or 0 x ffff.

The GTS request command shall be formatted as illustrated in Figure 57.

octets: 7	**1**	**1**
MHR fields	Command frame identifier (see Table 67)	GTS characteristics

Figure 57—GTS request command format

7.3.3.1.1 MHR fields

The fields of the MHR of the general MAC frame format (see Figure 34) shall be as indicated in this subclause.

The destination addressing mode subfield of the frame control field shall be set to 0 (e.g., destination addressing information not present), and the source addressing mode subfield shall be set to 2 (e.g., 16 bit short addressing).

The frame shall be processed for security by the sender according to the method defined by the security suite corresponding to *macCoordExtendedAddress*. If the security suite identifier is 0 x 00, the security enabled subfield of the frame control field shall be set to 0. Otherwise, the security enabled subfield shall be set to 1.

The frame pending subfield of the frame control field shall be set to 0 and ignored upon reception, and the acknowledgment request subfield shall be set to 1.

The source PAN identifier field shall contain the value of *macPANId*, and the source address field shall contain the value of *macShortAddress*.

7.3.3.1.2 GTS characteristics field

The GTS characteristics field is 8 bits in length and shall be formatted as illustrated in Figure 58.

bits: 0–3	**4**	**5**	**6–7**
GTS length	GTS direction	Characteristics type	Reserved

Figure 58—GTS characteristics field format

The GTS length subfield is 4 bits in length and shall contain the number of superframe slots being requested for the GTS.

The GTS direction subfield is 1 bit in length and shall be set to 1 if the GTS is to be a receive-only GTS. Conversely, this subfield shall be set to 0 if the GTS is to be a transmit-only GTS. GTS direction is defined relative to the direction of data frame transmissions by the device.

The characteristics type field is 1 bit in length and shall be set to 1 if the characteristics refers to a GTS allocation or 0 if the characteristics refers to a GTS deallocation.

7.4 MAC constants and PIB attributes

This subclause specifies the constants and attributes required by the MAC sublayer.

7.4.1 MAC constants

The constants that define the characteristics of the MAC sublayer are presented in Table 70.

7.4.2 MAC PIB attributes

The MAC PIB comprises the attributes required to manage the MAC sublayer of a device. Each of these attributes can be read or written using the MLME-GET.request and MLME-SET.request primitives, respectively. The attributes contained in the MAC PIB are presented in Table 71; attributes marked with a diamond (◆) are optional for an RFD. The MAC PIB security-related attributes are presented in Table 72 and Table 73.

Table 70—MAC sublayer constants

Constant	Description	Value
aBaseSlotDuration	The number of symbols forming a superframe slot when the superframe order is equal to 0.	60
aBaseSuperframeDuration	The number of symbols forming a superframe when the superframe order is equal to 0.	*aBaseSlotDuration* * aNumSuperframeSlots
aExtendedAddress	The 64 bit (IEEE) address assigned to the device.	Device specific
aMaxBE	The maximum value of the backoff exponent in the CSMA-CA algorithm.	5
aMaxBeaconOverhead	The maximum number of octets added by the MAC sublayer to the payload of its beacon frame.	75
aMaxBeaconPayloadLength	The maximum size, in octets, of a beacon payload.	*aMaxPHYPacketSize* – *aMaxBeaconOverhead*
aGTSDescPersistenceTime	The number of superframes in which a GTS descriptor exists in the beacon frame of a PAN coordinator.	4
aMaxFrameOverhead	The maximum number of octets added by the MAC sublayer to its payload without security. If security is required on a frame, its secure processing may inflate the frame length so that it is greater than this value. In this case, an error is generated through the appropriate .confirm or MLME-COMM-STATUS.indication primitives.	25
aMaxFrameResponseTime	The maximum number of CAP symbols in a beacon-enabled PAN, or symbols in a nonbeacon-enabled PAN, to wait for a frame intended as a response to a data request frame.	1220
aMaxFrameRetries	The maximum number of retries allowed after a transmission failure.	3
aMaxLostBeacons	The number of consecutive lost beacons that will cause the MAC sublayer of a receiving device to declare a loss of synchronization.	4
aMaxMACFrameSize	The maximum number of octets that can be transmitted in the MAC frame payload field.	*aMaxPHYPacketSize* – *aMaxFrameOverhead*
aMaxSIFSFrameSize	The maximum size of an MPDU, in octets, that can be followed by a short interframe spacing (SIFS) period.	18
aMinCAPLength	The minimum number of symbols forming the CAP. This ensures that MAC commands can still be transferred to devices when GTSs are being used. An exception to this minimum shall be allowed for the accommodation of the temporary increase in the beacon frame length needed to perform GTS maintenance (see 7.2.2.1.3).	440
aMinLIFSPeriod	The minimum number of symbols forming a long interframe spacing (LIFS) period.	40
aMinSIFSPeriod	The minimum number of symbols forming a SIFS period.	12
aNumSuperframeSlots	The number of slots contained in any superframe.	16

Table 70—MAC sublayer constants *(continued)*

Constant	Description	Value
aResponseWaitTime	The maximum number of symbols a device shall wait for a response command to be available following a request command.	32 * *aBaseSuperframeDuration*
aUnitBackoffPeriod	The number of symbols forming the basic time period used by the CSMA-CA algorithm.	20

Table 71—MAC PIB attributes

Attribute	Identifier	Type	Range	Description	Default
macAckWaitDuration	0 x 40	Integer	54 or 120	The maximum number of symbols to wait for an acknowledgment frame to arrive following a transmitted data frame. This value is dependent on the currently selected logical channel. For 0 ≤ *phyCurrentChannel* ≤ 10, this value is equal to 120. For 11 ≤ *phyCurrentChannel* ≤ 26, this value is equal to 54.	54
macAssociationPermit◆	0 x 41	Boolean	TRUE or FALSE	Indication of whether a coordinator is currently allowing association. A value of TRUE indicates that association is permitted.	FALSE
macAutoRequest	0 x 42	Boolean	TRUE or FALSE	Indication of whether a device automatically sends a data request command if its address is listed in the beacon frame. A value of TRUE indicates that the data request command is automatically sent.	TRUE
macBattLifeExt	0 x 43	Boolean	TRUE or FALSE	Indication of whether battery life extension, by reduction of coordinator receiver operation time during the CAP, is enabled. A value of TRUE indicates that it is enabled.	FALSE
macBattLifeExtPeriods	0 x 44	Integer	6 or 8	The number of backoff periods during which the receiver is enabled following a beacon in battery life extension mode. This value is dependent on the currently selected logical channel. For 0 ≤ *phyCurrentChannel* ≤ 10, this value is equal to 8. For 11 ≤ *phyCurrentChannel* ≤ 26, this value is equal to 6.	6
macBeaconPayload◆	0 x 45	Set of octets	—	The contents of the beacon payload.	NULL
macBeaconPayloadLength◆	0 x 46	Integer	0 — *aMaxBeaconPayloadLength*	The length, in octets, of the beacon payload.	0

Table 71—MAC PIB attributes *(continued)*

Attribute	Identifier	Type	Range	Description	Default
macBeaconOrder◆	0 x 47	Integer	0—15	Specification of how often the coordinator transmits a beacon. The *macBeaconOrder*, *BO*, and the beacon interval, *BI*, are related as follows: for $0 \leq BO \leq 14$, $BI = aBaseSuperframeDuration * 2^{BO}$ symbols. If *BO* = 15, the coordinator will not transmit a beacon.	15
macBeaconTxTime◆	0 x 48	Integer	0 x 000000 —0 x ffffff	The time that the device transmitted its last beacon frame, in symbol periods. The measurement shall be taken at the same symbol boundary within every transmitted beacon frame, the location of which is implementation specific. The precision of this value shall be a minimum of 20 bits, with the lowest four bits being the least significant.	0 x 000000
macBSN◆	0 x 49	Integer	0 x 00—0 x ff	The sequence number added to the transmitted beacon frame.	Random value from within the range.
macCoordExtended-Address	0 x 4a	IEEE address	An extended 64 bit IEEE address	The 64 bit address of the coordinator with which the device is associated.	—
macCoordShort-Address	0 x 4b	Integer	0 x 0000—0x ffff	The 16 bit short address assigned to the coordinator with which the device is associated. A value of 0 x fffe indicates that the coordinator is only using its 64 bit extended address. A value of 0 x ffff indicates that this value is unknown.	0 x ffff
macDSN	0 x 4c	Integer	0 x 00–0 x ff	The sequence number added to the transmitted data or MAC command frame.	Random value from within the range.
macGTSPermit◆	0 x 4d	Boolean	TRUE or FALSE	TRUE if the PAN coordinator is to accept GTS requests. FALSE otherwise.	TRUE
macMaxCSMABack-offs	0 x 4e	Integer	0—5	The maximum number of back-offs the CSMA-CA algorithm will attempt before declaring a channel access failure.	4

Table 71—MAC PIB attributes *(continued)*

Attribute	Identifier	Type	Range	Description	Default
macMinBE	0 x 4f	Integer	0—3	The minimum value of the backoff exponent in the CSMA-CA algorithm. Note that if this value is set to 0, collision avoidance is disabled during the first iteration of the algorithm. Also note that for the slotted version of the CSMA-CA algorithm with the battery life extension enabled, the minimum value of the backoff exponent will be the lesser of 2 and the value of *macMinBE*.	3
macPANId	0 x 50	Integer	0 x 0000—0 x ffff	The 16 bit identifier of the PAN on which the device is operating. If this value is 0 x ffff, the device is not associated.	0 x ffff
macPromiscuous-Mode◆	0 x 51	Boolean	TRUE or FALSE	This indicates whether the MAC sublayer is in a promiscuous (receive all) mode. A value of TRUE indicates that the MAC sublayer accepts all frames received from the PHY.	FALSE
macRxOnWhenIdle	0 x 52	Boolean	TRUE or FALSE	This indicates whether the MAC sublayer is to enable its receiver during idle periods.	FALSE
macShortAddress	0 x 53	Integer	0 x 0000—0 x ffff	The 16 bit address that the device uses to communicate in the PAN. If the device is a PAN coordinator, this value shall be chosen before a PAN is started. Otherwise, the address is allocated by a coordinator during association. A value of 0xfffe indicates that the device has associated but has not been allocated an address. A value of 0xffff indicates that the device does not have a short address.	0 x ffff
macSuperframe-Order◆	0 x 54	Integer	0—15	This specifies the length of the active portion of the superframe, including the beacon frame. The *macSuperframeOrder*, *SO*, and the superframe duration, *SD*, are related as follows: for $0 \leq SO \leq BO \leq 14$, $SD = aBaseSuperframeDuration * 2^{SO}$ symbols. If $SO = 15$, the superframe will not be active following the beacon.	15
macTransaction-PersistenceTime◆	0 x 55	Integer	0 x 0000—0 x ffff	The maximum time (in superframe periods) that a transaction is stored by a coordinator and indicated in its beacon.	0 x 01f4

Table 72—MAC PIB security attributes

Attribute	Identifier	Type	Range	Description	Default
macACLEntryDescriptorSet	0 x 70	Set of ACL descriptor values (see Table 73)	Variable	A set of ACL entries, each containing address information, security suite information, and security material to be used to protect frames between the MAC sublayer and the specified device.	Null set
macACLEntryDescriptorSetSize	0 x 71	Integer	0 x 00—0 x ff	The number of entries in the ACL descriptor set.	0 x 00
macDefaultSecurity	0 x 72	Boolean	TRUE or FALSE	Indication of whether the device is able to transmit secure frames to or accept secure frames from devices that are not explicitly listed in the ACL. It is also used to communicate with multiple devices at once. A value of TRUE indicates that such transmissions are permitted.	FALSE
macDefaultSecurityMaterialLength	0 x 73	Integer	0 x 00—0 x 1a	The number of octets contained in *ACLSecurityMaterial.*	0 x 15
macDefaultSecurityMaterial	0 x 74	octet string	Variable	The specific security material to be used to protect frames between the MAC sublayer and devices not in the ACL (see 7.6.1.8).	Empty string
macDefaultSecuritySuite	0 x 75	Integer	0 x 00—0 x 07	The unique identifier of the security suite to be used to protect communications between the MAC and devices not in the ACL as specified in Table 75.	0 x 00
macSecurityMode	0 x 76	Integer	0 x 00—0 x 02	The identifier of the security mode in use as specified in 7.5.8. 0 x 00 = Unsecured mode. 0 x 01 = ACL mode. 0 x 02 = Secured mode.	0 x 00

Table 73—Elements of ACL entry descriptor

Name	Type	Range	Description	Default
ACLExtendedAddress	IEEE address	Any valid 64 bit device address	The 64 bit IEEE extended address of the device in this ACL entry.	Device specific
ACLShortAddress	Integer	0 x 0000—0 x ffff	The 16 bit short address of the device in this ACL entry. A value of 0 x fffe indicates that the device is using only its 64 bit extended address. A value of 0 x ffff indicates that this value is unknown.	0 x ffff

Table 73—Elements of ACL entry descriptor *(continued)*

Name	Type	Range	Description	Default
ACLPANId	Integer	0 x 0000–0 x ffff	The 16 bit PAN identifier of the device in this ACL entry.	Device specific
ACLSecurityMaterial-Length	Integer	0–26	The number of octets contained in *ACL-SecurityMaterial*.	21
ACLSecurityMaterial	Octet string	Variable	The specific keying material to be used to protect frames between the MAC sublayer and the device indicated by the associated *ACLExtendedAddress* (see 7.6.1.8).	Empty string
ACLSecuritySuite	Integer	0 x 00–0 x 07	The unique identifier of the security suite to be used to protect communications between the MAC sublayer and the device indicated by the associated *ACLExtendedAddress* as specified in Table 75.	0 x 00

7.5 MAC functional description

This subclause provides a detailed description of the MAC functionality. Subclause 7.5.1 describes the following two mechanisms for channel access: contention based and contention free. Contention-based access allows devices to access the channel in a distributed fashion using a CSMA-CA backoff algorithm. Contention-free access is controlled entirely by the PAN coordinator through the use of GTSs.

The mechanisms used for starting and maintaining a PAN are described in 7.5.2. Channel scanning is used by a device to assess the current state of a channel (or channels), locate all beacons within its POS, or locate a particular beacon with which it has lost synchronization. Before starting a new PAN, the results of a channel scan can be used to select an appropriate logical channel and a PAN identifier that is not being used by any other PAN in the area. Because it is still possible for the POS of two PANs with the same PAN identifier to overlap, a procedure exists to detect and resolve this situation. Following a channel scan and suitable PAN identifier selection, an FFD can begin operating as the PAN coordinator. Also described in the subclause is a method to allow a beaconing FFD to discover other such devices during normal operations, i.e., when not scanning.

The mechanisms to allow devices to join or leave a PAN are defined in 7.5.3. The association procedure describes the conditions under which a device may join a PAN and the conditions necessary for a coordinator to permit devices to join. Also described is the disassociation procedure, which can be initiated by the associated device or its coordinator.

The mechanisms to allow devices to acquire and maintain synchronization with a coordinator are described in 7.5.4. Synchronization on a beacon-enabled PAN is described after first explaining how a coordinator generates beacon frames. Following this explanation, synchronization on a nonbeacon-enabled PAN is described. A procedure to reestablish communication between a device and its coordinator has been created, as it is possible that a device may lose synchronization in the case of either a beacon-enabled or a nonbeacon-enabled PAN.

IEEE Std 802.15.4-2003 has been designed so that application data transfers can be controlled by the devices on a PAN rather than by the coordinator. The procedures the coordinator uses to handle multiple transactions while preserving this requirement are described in 7.5.5.

The mechanisms for transmitting, receiving, and acknowledging frames, including frames sent using indirect transmission, are described in 7.5.6. In addition, methods for retransmitting frames are also described.

The mechanisms for allocating and deallocating a GTS are described in 7.5.7. The deallocation process may result in the fragmentation of the GTS space, i.e., an unused slot or slots. The subclause describes a mechanism to resolve fragmentation.

When the next higher layer requests that security be implemented prior to transmission of a frame or upon receipt of a frame, the MAC sublayer uses the mechanisms defined in 7.5.8.

Throughout this subclause, the receipt of a frame is defined as the successful receipt of the frame by the PHY and the successful verification of the FCS by the MAC sublayer, as described in 7.2.1.8.

7.5.1 Channel access

This subclause describes the mechanisms for accessing the physical radio channel.

7.5.1.1 Superframe structure

The coordinator on a PAN can optionally bound its channel time using a superframe structure. A superframe is bounded by the transmission of a beacon frame and can have an active portion and an inactive portion. The coordinator shall interact with its PAN only during the active portion of the superframe and, therefore, may enter a low power (sleep) mode during the inactive portion.

The structure of this superframe is described by the values of *macBeaconOrder* and *macSuperframeOrder*. The MAC PIB attribute *macBeaconOrder*, describes the interval at which the coordinator shall transmit its beacon frames. The value of *macBeaconOrder*, *BO*, and the beacon interval, *BI*, are related as follows: for $0 \leq BO \leq 14$, $BI = aBaseSuperframeDuration * 2^{BO}$ symbols. The value of *macSuperframeOrder* shall be ignored if $BO = 15$.

The MAC PIB attribute *macSuperframeOrder* describes the length of the active portion of the superframe, which includes the beacon frame. The value of *macSuperframeOrder*, *SO*, and the superframe duration, *SD*, are related as follows: for $0 \leq SO \leq BO \leq 14$, $SD = aBaseSuperframeDuration * 2^{SO}$ symbols. If $SO = 15$, the superframe shall not remain active after the beacon. If $BO = 15$, the superframe shall not exist (the value of macSuperframeOrder shall be ignored), and *macRxOnWhenIdle* shall define whether the receiver is enabled during periods of transceiver inactivity.

The active portion of each superframe shall be divided into *aNumSuperframeSlots* equally spaced slots of duration $2^{SO} * aBaseSlotDuration$ and is composed of three parts: a beacon, a CAP and a CFP. The beacon shall be transmitted, without the use of CSMA, at the start of slot 0, and the CAP shall commence immediately after the beacon. The CFP, if present, follows immediately after the CAP and extends to the end of the active portion of the superframe. Any allocated GTSs shall be located within the CFP.

PANs that wish to use the superframe structure shall set *macBeaconOrder* to a value between 0 and 14 and *macSuperframeOrder* to a value between 0 and the value of *macBeaconOrder*.

PANs that do not wish to use the superframe structure (referred to as a nonbeacon-enabled PAN) shall set both *macBeaconOrder* and *macSuperframeOrder* to 15. In this case, a coordinator shall not transmit beacons; and all transmissions, with the exception of acknowledgment frames and any data frame that quickly follows the acknowledgment of a data request command (see 7.5.6.3), shall use an unslotted CSMA-CA mechanism to access the channel. In addition, GTSs shall not be permitted.

An example of the superframe structure is shown in Figure 59. In this case, the beacon interval, *BI*, is twice as long as the active superframe duration, *SD*, and the CFP contains two GTSs.

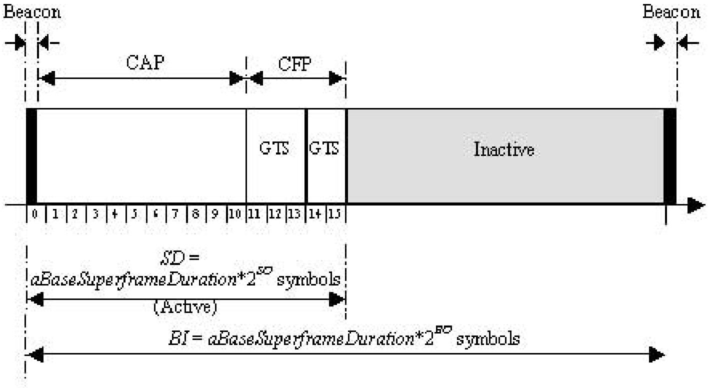

Beacon
Beacon
CAP
CFP
GTS
GTS
Inactive
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
SD = aBaseSuperframeDuration*2^SO symbols
(Active)
BI = aBaseSuperframeDuration*2^BO symbols

Figure 59—An example of the superframe structure

7.5.1.1.1 CAP

The CAP shall start immediately following the beacon and complete before the beginning of the CFP on a superframe slot boundary. If the CFP is zero length, the CAP shall complete at the end of the superframe. The CAP shall be at least *aMinCAPLength* symbols, unless additional space is needed to temporarily accommodate the increase in the beacon frame length needed to perform GTS maintenance (see 7.2.2.1.3), and shall shrink or grow dynamically to accommodate the size of the CFP.

All frames, except acknowledgment frames and any data frame that quickly follows the acknowledgment of a data request command (see 7.5.6.3), transmitted in the CAP shall use a slotted CSMA-CA mechanism to access the channel. A device transmitting within the CAP shall ensure that its transaction is complete (i.e., including the reception of any acknowledgment) one IFS period (see 7.5.1.2) before the end of the CAP. If this is not possible, the device shall defer its transmission until the CAP of the following superframe.

MAC command frames shall always be transmitted in the CAP.

7.5.1.1.2 CFP

The CFP shall start on a slot boundary immediately following the CAP and it shall complete before the start of the next beacon. If any GTSs have been allocated by the PAN coordinator, they shall be located within the CFP and occupy contiguous slots. The CFP shall therefore grow or shrink depending on the total length of all of the combined GTSs.

No transmissions within the CFP shall use a CSMA-CA mechanism to access the channel. A device transmitting in the CFP shall ensure that its transmissions are complete one IFS period (see 7.5.1.2) before the end of its GTS.

7.5.1.2 IFS

The MAC sublayer needs a finite amount of time to process data received by the PHY. To allow for this, transmitted frames shall be followed by an IFS period; if the transmission requires an acknowledgment, the IFS shall follow the acknowledgment frame. The length of the IFS period is dependent on the size of the frame that has just been transmitted. Frames (i.e., MPDUs) of up to *aMaxSIFSFrameSize* in length shall be followed by a SIFS period of a duration of at least *aMinSIFSPeriod* symbols. Frames (i.e., MPDUs) with lengths greater than *aMaxSIFSFrameSize* shall be followed by a LIFS of a duration of at least *aMinLIFSPeriod* symbols. These concepts are illustrated in Figure 60.

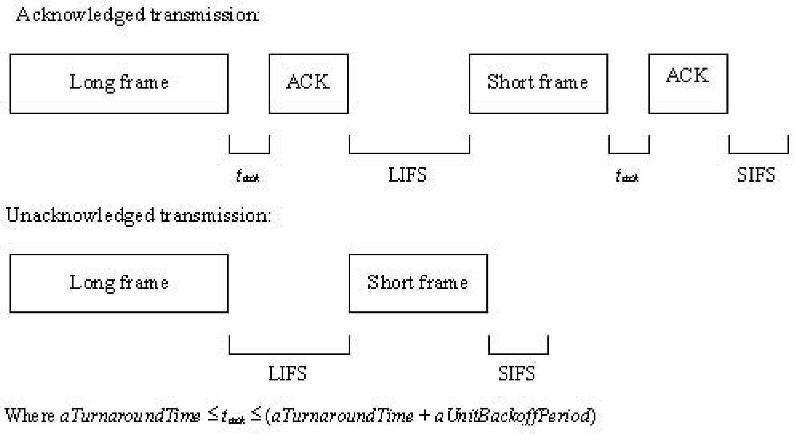

Acknowledged transmission:
Long frame
ACK
Short frame
ACK
t_{ack}
LIFS
t_{ack}
SIFS
Unacknowledged transmission:
Long frame
Short frame
LIFS
SIFS
Where *aTurnaroundTime* ≤ t_{ack} ≤ (*aTurnaroundTime* + *aUnitBackoffPeriod*)

Figure 60—IFS

The CSMA-CA algorithm shall take this requirement into account for transmissions in the CAP.

7.5.1.3 The CSMA-CA algorithm

The CSMA-CA algorithm shall be used before the transmission of data or MAC command frames transmitted within the CAP, unless the frame can be quickly transmitted following the acknowledgment of a data request command (see 7.5.6.3 for timing requirements). The CSMA-CA algorithm shall not be used for the transmission of beacon frames, acknowledgment frames, or data frames transmitted in the CFP.

If beacons are being used in the PAN, the MAC sublayer shall employ the slotted version of the CSMA-CA algorithm for transmissions in the CAP of the superframe. Conversely, if beacons are not being used in the PAN or if a beacon could not be located in a beacon-enabled PAN, the MAC sublayer shall transmit using the unslotted version of the CSMA-CA algorithm. In both cases, the algorithm is implemented using units of time called backoff periods, where one backoff period shall be equal to *aUnitBackoffPeriod* symbols.

In slotted CSMA-CA, the backoff period boundaries of every device in the PAN shall be aligned with the superframe slot boundaries of the PAN coordinator, i.e., the start of the first backoff period of each device is aligned with the start of the beacon transmission. In slotted CSMA-CA, the MAC sublayer shall ensure that the PHY commences all of its transmissions on the boundary of a backoff period. In unslotted CSMA-CA, the backoff periods of one device are not related in time to the backoff periods of any other device in the PAN.

Each device shall maintain three variables for each transmission attempt: *NB*, *CW* and *BE*. *NB* is the number of times the CSMA-CA algorithm was required to backoff while attempting the current transmission; this value shall be initialized to 0 before each new transmission attempt. *CW* is the contention window length, defining the number of backoff periods that need to be clear of channel activity before the transmission can commence; this value shall be initialized to 2 before each transmission attempt and reset to 2 each time the channel is assessed to be busy. The *CW* variable is only used for slotted CSMA-CA. *BE* is the backoff exponent, which is related to how many backoff periods a device shall wait before attempting to assess a channel. In unslotted systems, or slotted systems with *macBattLifeExt* set to FALSE, *BE* shall be initialized to the value of *macMinBE*. In slotted systems with *macBattLifeExt* set to TRUE, this value shall be initialized to the lesser of 2 and the value of *macMinBE*. Note that if *macMinBE* is set to 0, collision avoidance will be disabled during the first iteration of this algorithm.

Although the receiver of the device is enabled during the channel assessment portion of this algorithm, the device shall discard any frames received during this time.

Figure 61 illustrates the steps of the CSMA-CA algorithm. When using slotted CSMA-CA, the MAC sublayer shall first initialize *NB*, *CW*, and *BE* and then locate the boundary of the next backoff period [step (1)]. For unslotted CSMA-CA, the MAC sublayer shall initialize *NB* and *BE* and then proceed directly to step (2).

The MAC sublayer shall delay for a random number of complete backoff periods in the range 0 to $2^{BE} - 1$ [step (2)] and then request that the PHY perform a CCA [step (3)]. In a slotted CSMA-CA system, the CCA shall start on a backoff period boundary. In an unslotted CSMA-CA system, the CCA shall start immediately.

In a slotted CSMA-CA system with the battery life extension subfield (see Figure 42) set to 0, the MAC sublayer shall ensure that, after the random backoff, the remaining CSMA-CA operations can be undertaken and the entire transaction can be transmitted before the end of the CAP. If the number of backoff periods is greater than the remaining number of backoff periods in the CAP, the MAC sublayer shall pause the backoff countdown at the end of the CAP and resume it at the start of the CAP in the next superframe. If the number of backoff periods is less than or equal to the remaining number of backoff periods in the CAP, the MAC sublayer shall apply its backoff delay and then evaluate whether it can proceed. The MAC sublayer shall proceed if the remaining CSMA-CA algorithm steps (i.e., two CCA analyses), the frame transmission, and any acknowledgment can be completed before the end of the CAP. If the MAC sublayer can proceed, it shall request that the PHY perform the CCA in the current superframe. If the MAC sublayer cannot proceed, it shall wait until the start of the CAP in the next superframe and repeat the evaluation.

In a slotted CSMA-CA system with the battery life extension subfield set to 1, the MAC sublayer shall ensure that, after the random backoff, the remaining CSMA-CA operations can be undertaken and the entire transaction can be transmitted before the end of the CAP. The backoff countdown shall only occur during the first six full backoff periods after the end of the beacon’s IFS period. The MAC sublayer shall proceed if the remaining CSMA-CA algorithm steps (two CCA analyses), the frame transmission, and any acknowledgment can be completed before the end of the CAP, and the frame transmission will start in one of the first six full backoff periods after the beacon’s IFS period. If the MAC sublayer can proceed, it shall request that the PHY perform the CCA in the current superframe. If the MAC sublayer cannot proceed, it shall wait until the start of the CAP in the next superframe and repeat the evaluation.

If the channel is assessed to be busy [step (4)], the MAC sublayer shall increment both *NB* and *BE* by one, ensuring that *BE* shall be no more than *aMaxBE*. The MAC sublayer in a slotted CSMA-CA system shall also reset *CW* to 2. If the value of *NB* is less than or equal to *macMaxCSMABackoffs*, the CSMA-CA algorithm shall return to step (2). If the value of *NB* is greater than *macMaxCSMABackoffs*, the CSMA-CA algorithm shall terminate with a Channel Access Failure status.

If the channel is assessed to be idle [step (5)], the MAC sublayer in a slotted CSMA-CA system shall ensure that the contention window has expired before commencing transmission. To do this, the MAC sublayer shall first decrement *CW* by one and then determine whether it is equal to 0. If it is not equal to 0, the CSMA-CA algorithm shall return to step (3). If it is equal to 0, the MAC sublayer shall begin transmission of the frame on the boundary of the next backoff period. If the channel is assessed to be idle in an unslotted CSMA-CA system, the MAC sublayer shall begin transmission of the frame immediately.

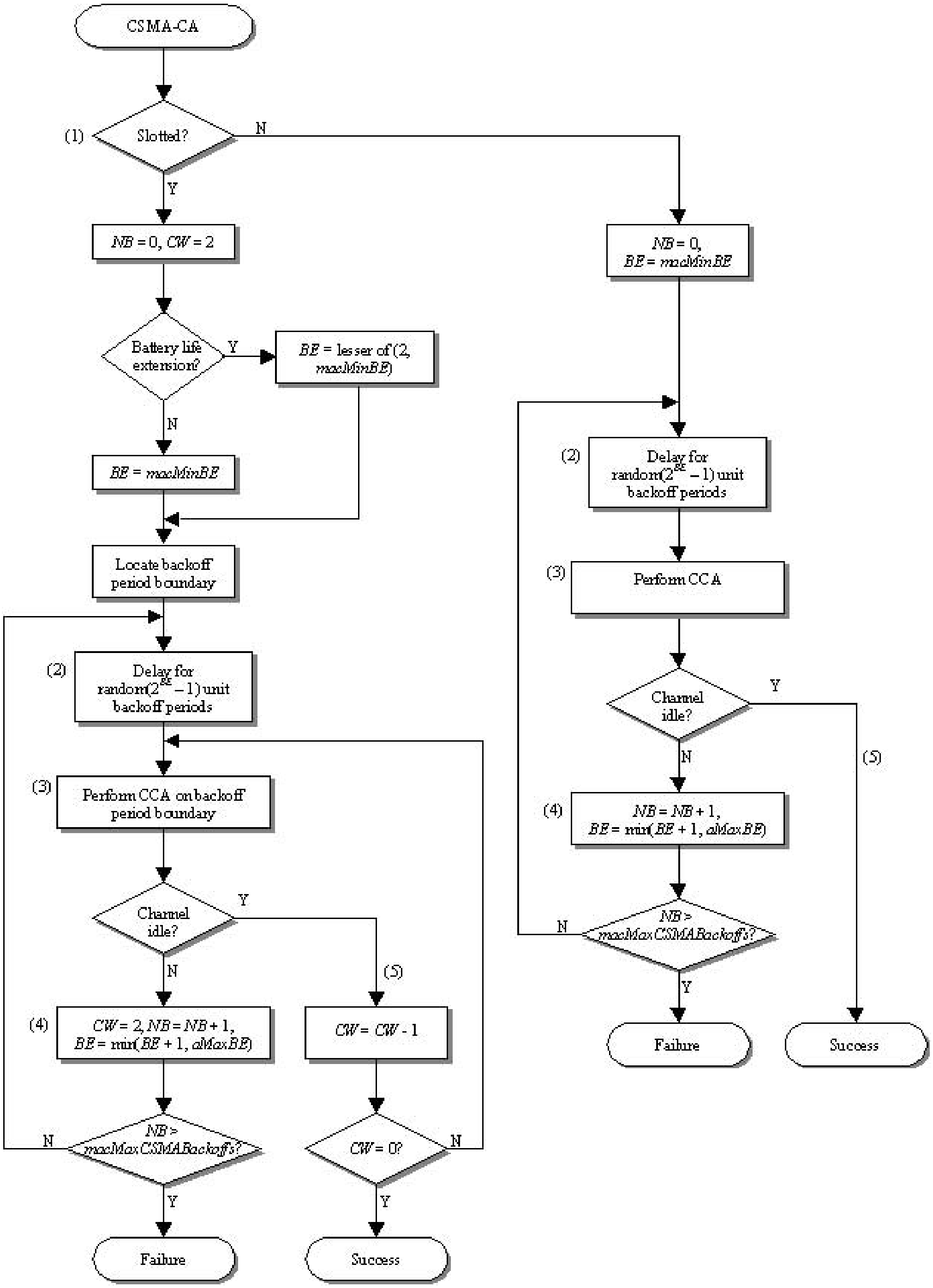


Figure 61—The CSMA-CA algorithm

7.5.2 Starting and maintaining PANs

This subclause specifies the procedures for scanning through channels, identifying PAN identifier conflicts, and starting PANs.

7.5.2.1 Scanning through channels

All devices shall be capable of performing passive and orphan scans across a specified list of channels. In addition, an FFD shall be able to perform ED and active scans. The next higher layer should submit a scan request containing a list of channels chosen only from the channels specified by *phyChannelsSupported*.

A device is instructed to begin a channel scan through the MLME-SCAN.request primitive. For the duration of the scan, the device shall suspend beacon transmissions, if applicable; and upon the conclusion of the scan, the device shall recommence beacon transmissions. The results of the scan shall be returned via the MLME-SCAN.confirm primitive.

7.5.2.1.1 ED channel scan

An ED scan allows an FFD to obtain a measure of the peak energy in each requested channel. This could be used by a prospective PAN coordinator to select a channel in which to operate prior to starting a new PAN. During an ED scan, the MAC sublayer shall discard all frames received over the PHY data service.

An ED scan over a specified set of logical channels is requested using the MLME-SCAN.request primitive with the ScanType parameter set to indicate an ED scan. For each logical channel, the MLME shall first switch to the channel, by setting *phyCurrentChannel* accordingly, and then repeatedly perform an ED measurement for [*aBaseSuperframeDuration* * $(2^n + 1)$] symbols, where *n* is the value of the ScanDuration parameter in the MLME-SCAN.request primitive. An ED measurement is performed by the MLME issuing the PLME-ED.request (see 6.2.2.3), which is guaranteed to return a value. The maximum ED measurement obtained during this period shall be noted before moving on to the next channel in the channel list. A device shall be able to store between one and an implementation-specified maximum number of channel ED measurements.

The ED scan shall terminate when either the number of channel ED measurements stored equals the implementation-specified maximum or energy has been measured on each of the specified logical channels.

7.5.2.1.2 Active channel scan

An active scan allows an FFD to locate any coordinator transmitting beacon frames within its POS. This could be used by a prospective PAN coordinator to select a PAN identifier prior to starting a new PAN, or it could be used by a device prior to association. During an active scan, the MAC sublayer shall discard all frames received over the PHY data service that are not beacon frames.

Before commencing an active scan, the MAC sublayer shall store the value of *macPANId* and then set it to 0xffff for the duration of the scan. This enables the receive filter to accept all beacons rather than just the beacons from its current PAN (see 7.5.6.2). On completion of the scan, the MAC sublayer shall restore the value of *macPANId* to the value stored before the scan began.

An active scan over a specified set of logical channels is requested using the MLME-SCAN.request primitive with the ScanType parameter set to indicate an active scan. For each logical channel, the device shall first switch to the channel, by setting *phyCurrentChannel* accordingly, and send a beacon request command (see 7.3.2.4). The device shall then enable its receiver for at most [*aBaseSuperframeDuration* * $(2^n + 1)$] symbols, where *n* is a value between 0 and 14. During this time, the device shall reject all nonbeacon frames and record the information contained in all unique beacons in a PAN descriptor structure (see Table 41 in 7.1.5.1.1). A device shall be able to store between one and an implementation-specified maximum number of PAN descriptors. A beacon frame shall be assumed to be unique if it contains both a PAN identifier and a source address that has not been seen before during the scan of the current channel.

If a beacon frame is received with the security enabled subfield in the frame control field set to 1, the device shall attempt to process the beacon frame for security, as described in 7.5.8. Any errors encountered during the secure processing of the beacon frame shall be ignored and the beacon information shall be recorded in a PAN descriptor with the SecurityUse, ACLEntry, and SecurityFailure fields (see Table 41) set accordingly.

If a coordinator of a beacon-enabled PAN receives the beacon request command, it shall ignore the command and continue transmitting its beacons as usual. If a coordinator of a nonbeacon-enabled PAN receives this command, it shall transmit a single beacon frame using unslotted CSMA-CA.

The active scan on a particular channel shall terminate when the number of beacons found equals the implementation-specified limit or the channel has been scanned for the full time, as specified in 7.5.2.1.2. If the latter condition has been satisfied, the channel shall be considered to have been scanned. Where possible, the scan shall be repeated on each channel. The entire scan shall terminate when the number of PAN descriptors stored equals the implementation-specified maximum or every channel in the set of available channels has been scanned.

7.5.2.1.3 Passive channel scan

A passive scan, like an active scan, allows a device to locate any coordinator transmitting beacon frames within its POS. The beacon request command, however, is not transmitted. This type of scan could be used by a device prior to association. During a passive scan, the MAC sublayer shall discard all frames received over the PHY data service that are not beacon frames.

Before commencing a passive scan, the MAC sublayer shall store the value of *macPANId* and then set it to 0 x ffff for the duration of the scan. This enables the receive filter to accept all beacons rather than just the beacons from its current PAN (see 7.5.6.2). On completion of the scan, the MAC sublayer shall restore the value of *macPANId* to the value stored before the scan began.

A passive scan over a specified set of logical channels is requested using the MLME-SCAN.request primitive with the ScanType parameter set to indicate a passive scan. For each logical channel, the device shall first switch to the channel, by setting *phyCurrentChannel* accordingly, and then enable its receiver for at most [*aBaseSuperframeDuration* * (2^n + 1)] symbols, where *n* is a value between 0 and 14. During this time, the device shall reject all nonbeacon frames and record the information contained in all unique beacons in a PAN descriptor structure (see Table 41 in 7.1.5.1.1). A device shall be able to store between one and an implementation- specified maximum number of PAN descriptors. A beacon frame shall be assumed to be unique if it contains both a PAN identifier and a source address that has not been seen before during the scan of the current channel.

If a beacon frame is received with the security enabled subfield set to 1, the device shall attempt to process the beacon frame for security, as described in 7.5.8. Any errors encountered during the secure processing of the beacon frame shall be ignored, and the beacon information shall be recorded in a PAN descriptor with the SecurityUse, ACLEntry, and SecurityFailure fields (see Table 41) set accordingly.

The passive scan on a particular channel shall terminate when the number of beacons found equals the implementation specified limit or the channel has been scanned for the full time, as specified in 7.5.2.1.3. If the latter condition has been satisfied, the channel shall be considered to have been scanned. Where possible, the scan shall be repeated on each channel. The entire scan shall terminate when the number of PAN descriptors stored equals the implementation-specified maximum or every channel in the set of available channels has been scanned.

7.5.2.1.4 Orphan channel scan

An orphan scan allows a device to attempt to relocate its coordinator following a loss of synchronization. During an orphan scan, the MAC sublayer shall discard all frames received over the PHY data service that are not coordinator realignment MAC command frames.

An orphan scan over a specified set of logical channels is requested using the MLME-SCAN.request primitive with the ScanType parameter set to indicate an orphan scan. For each logical channel, the device shall first switch to the channel, by setting *phyCurrentChannel* accordingly, and then send an orphan notification command (see 7.3.2.3). The device shall then enable its receiver for at most *aResponseWaitTime* symbols. If the device successfully receives a coordinator realignment command (see 7.3.2.5) within this time, the device shall disable its receiver.

If a coordinator receives the orphan notification command, it shall search its device list for the device sending the command. If the coordinator finds a record of the device, it shall send a coordinator realignment command to the orphaned device. The process of searching for the device and sending the coordinator realignment command shall occur within *aResponseWaitTime* symbols. The coordinator realignment command shall contain its current PAN identifier, *macPANId*, its current logical channel, and the short address of the orphaned device. If a coordinator finds no record of the device, it shall ignore the command and not send a coordinator realignment command.

The orphan scan shall terminate when the device receives a coordinator realignment command or the specified set of logical channels has been scanned.

7.5.2.2 PAN identifier conflict resolution

In some instances a situation could occur in which two PANs exist in the same POS with the same PAN identifier. If this conflict happens, the coordinator and its devices shall perform PAN identifier conflict resolution procedure.

This procedure is optional for an RFD.

7.5.2.2.1 Detection

The PAN coordinator shall conclude that a PAN identifier conflict is present if either of the following apply:

- A beacon frame is received by the PAN coordinator with the PAN coordinator subfield (see 7.2.2.1.2) set to 1 and the PAN identifier equal to *macPANId*.
- A PAN ID conflict notification command (see 7.3.2.2) is received by the PAN coordinator from a device on its PAN.

A device shall conclude that a PAN identifier conflict is present if the following applies:

- A beacon frame is received by the device with the PAN coordinator subfield set to 1, the PAN identifier equal to *macPANId*, and an address that is not equal to both *macCoordShortAddress* and *macCoordExtendedAddress*.

7.5.2.2.2 Resolution

On the detection of a PAN identifier conflict, the coordinator shall first perform an active scan and then, using the information from the scan, select a new PAN identifier. The algorithm for selecting a suitable PAN identifier is out of the scope of this standard. The coordinator shall then broadcast the coordinator realignment command containing the new PAN identifier with the source PAN identifier field equal to the value in *macPANId*. Once the coordinator realignment command has been sent, the coordinator shall set *macPANId* to the new PAN identifier.

On the detection of a PAN identifier conflict by a device, it shall generate the PAN ID conflict notification command (see 7.3.2.2) and send it to the PAN coordinator. If the PAN ID conflict notification command is received correctly, the PAN coordinator shall send an acknowledgment frame, thus confirming receipt. The PAN coordinator shall then resolve the conflict as described in this subclause.

7.5.2.3 Starting a PAN

A PAN shall be started by an FFD only after an active channel scan has been performed and a suitable PAN identifier selection has been made. The algorithm for selecting a suitable PAN identifier from the list of PAN descriptors returned from the active channel scan procedure is out of the scope of this standard. In addition, an FFD shall set *macShortAddress* to a value less than 0 x ffff.

An FFD is instructed to begin operating a PAN through the use of the MLME-START.request primitive (see 7.1.14.1) with the PANCoordinator parameter set to TRUE and the CoordRealigment parameter set to FALSE. On receipt of this primitive, the MAC sublayer shall set the logical channel in *phyCurrentChannel* and the PAN identifier in *macPANId*. After completing this, the MAC sublayer shall respond with the MLME-START.confirm primitive and begin operating as a PAN coordinator.

7.5.2.4 Beacon generation

A device shall be permitted to transmit beacon frames only if *macShortAddress* is not equal to 0 x ffff.

An FFD shall use the MLME-START.request primitive to begin transmitting beacons. The FFD may begin beacon transmission either as the PAN coordinator of a new PAN or as a device on a previously established PAN, depending upon the setting of the PANCoordinator parameter (see 7.1.14.1). On receipt of this primitive, the MAC sublayer shall set the PAN identifier in *macPANId* and use this value in the source PAN identifier field of the beacon frame. The address used in the source address field of the beacon frame shall contain the value of *aExtendedAddress* if *macShortAddress* is equal to 0xfffe or *macShortAddress* otherwise.

The time of transmission of the most recent beacon shall be recorded in *macBeaconTxTime* and shall be computed so that its value is taken at the same symbol boundary in each beacon frame, the location of which is implementation specific. The symbol boundary shall be chosen to be the same as that used in the timestamp of the incoming beacon frame, as described in 7.5.4.1.

All beacon frames shall be transmitted at the beginning of each superframe at an interval equal to *aBaseSuperframeDuration* * 2^n symbols, where *n* is the value of *macBeaconOrder*. The beacon frame shall be constructed as specified in 7.2.2.1.

Beacon transmissions shall be given priority over all other transmit and receive operations.

7.5.2.5 Device discovery

An FFD may indicate its presence on a PAN to other devices by transmitting beacon frames. This allows the other devices to perform device discovery.

An FFD that is not the PAN coordinator shall begin transmitting beacon frames only when it has successfully associated with a PAN. The transmission of beacon frames by the device is initiated through the use of the MLME-START.request primitive with the PANCoordinator parameter set to FALSE. On receipt of this primitive, the MLME shall begin transmitting beacons using the identifier of the PAN with which the device has associated, *macPANId*, and its short address, *macShortAddress*. A beacon frame shall be transmitted at a rate of one beacon frame every *aBaseSuperframeDuration* * 2^n symbols, where *n* is the value of *macBeaconOrder.*

7.5.3 Association and disassociation

This subclause specifies the procedures for association and disassociation.

7.5.3.1 Association

A device shall attempt to associate only after having first performed a MAC sublayer reset, by issuing the MLME-RESET.request primitive, and then having completed either an active channel scan (see 7.5.2.1.2) or a passive channel scan (see 7.5.2.1.3). The results of the channel scan would have then been used to choose a suitable PAN. The algorithm for selecting a suitable PAN with which to associate from the list of PAN descriptors returned from the channel scan procedure is out of the scope of this standard.

A coordinator shall allow association only if *macAssociationPermit* is set to TRUE. Similarly, a device shall attempt to associate only with a PAN that is currently allowing association, as indicated in the results of the scanning procedure. If a coordinator with *macAssociationPermit* set to FALSE receives an association request command from a device, the command shall be ignored.

Following the selection of a PAN with which to associate, the next higher layers shall request that the MLME configures the following PHY and MAC PIB attributes to the values necessary for association:

— *phyCurrentChannel* shall be set to an appropriate logical channel on which to associate.
— *macPANId* shall be set to the identifier of the PAN with which to associate.
— *macCoordExtendedAddress* or *macCoordShortAddress*, shall be set to the appropriate value according to the becon frame from the coordinator with which it wishes to associate.

In order to optimize the association procedure on a beacon-enabled PAN, a device may begin tracking the beacon of the coordinator with which it wishes to associate a priori. This is achieved by issuing the MLME-SYNC.request primitive with the TrackBeacon parameter set to TRUE.

A device that is instructed to associate with a PAN, through the MLME-ASSOCIATE.request primitive, shall try to associate only with an existing PAN and shall not attempt to start its own PAN.

An unassociated device shall initiate the association procedure by sending an associate request command (see 7.3.1.1) to the coordinator of an existing PAN. If the association request command is received correctly, the coordinator shall send an acknowledgment frame, thus confirming receipt.

The acknowledgment to an association request command does not mean that the device has associated. The coordinator needs time to determine whether the current resources available on the PAN are sufficient to allow another device to associate. The coordinator shall make this decision within *aResponseWaitTime* symbols. If the coordinator finds that the device was previously associated on its PAN, all previously obtained device-specific information shall be removed. If sufficient resources are available, the coordinator shall allocate a short address to the device and generate an association response command (see 7.3.1.2) containing the new address and a status indicating a successful association. If sufficient resources are not available, the coordinator shall generate an association response command containing a status indicating a failure (see Table 68). The association response command shall be sent to the device requesting association using indirect transmission, i.e., the association response command frame shall be added to the list of pending transactions stored on the coordinator and extracted at the discretion of the device concerned using the method described in 7.5.6.3.

If the allocate address subfield of the capability information field (see 7.3.1.1.2) of the association request command is set to 1, the coordinator shall allocate a 16 bit address with a range depending on the addressing mode supported by the coordinator, as described in Table 74. If the allocate address subfield of the association request command is set to 0, the address allocated by the coordinator shall be equal to 0 x fffe. A short address of 0 x fffe is a special case that indicates that the device has associated, but has not been allocated a short address. In this case, the device shall use only its 64 bit extended address to operate on the network.

On receipt of the acknowledgment to the association request command, the device shall wait for at most *aResponseWaitTime* symbols for the coordinator to make its association decision. If the device is tracking the beacon, it shall attempt to extract the association response command from the coordinator whenever it appears in the beacon frame. If the device is not tracking the beacon, it shall attempt to extract the association response command from the coordinator after *aResponseWaitTime* symbols. If the device does not extract an association response command frame from the coordinator within *aResponseWaitTime* symbols, it shall issue the MLME-ASSOCIATE.confirm primitive with a status of NO_DATA, and the association attempt shall be deemed a failure. In this case, the next higher layer shall terminate any tracking of the beacon. This is achieved by issuing the MLME-SYNC.request primitive with the TrackBeacon parameter set to FALSE.

On receiving the association response command, the device requesting association shall send an acknowledgment frame, thus confirming receipt. If the association status field of the command indicates that the association was successful, the device shall store the addresses of the coordinator with which it has associated. The short address of the coordinator, contained in the original beacon selected for association following a scan, shall be stored in *macCoordShortAddress* and the extended address of the coordinator, contained in the MHR of the association response command frame, shall be stored in *macCoordExtendedAddress*. The device shall also store the address contained in the short address field in *macShortAddress*. Communication on the PAN using this short address shall depend on its range, as described in Table 74.

Table 74—Usage of the short address

Value of *macShortAddress*	Description
0 x 0000–0 x fffd	The device shall use short addressing mode.
0 x fffe	The device shall use 64 bit extended addressing mode with an address consisting of *aExtendedAddress*.
0 x ffff	The device is not associated and shall not communicate on the PAN.

If the association status field of the command indicates that the association was unsuccessful, the device shall set *macPANId* to the default value (0 x ffff).

7.5.3.2 Disassociation

The disassociation procedure is initiated by the next higher layer by issuing the MLME-DISASSOCIATE.request primitive to the MLME.

When a coordinator wants one of its associated devices to leave the PAN, it shall send the disassociation notification command to the device using indirect transmission, i.e., the disassociation notification command frame shall be added to the list of pending transactions stored on the coordinator and extracted at the discretion of the device concerned using the method described in 7.5.6.3. If the device requests and correctly receives the disassociation notification command, it shall confirm its receipt by sending an acknowledgment frame. Even if the acknowledgment is not received, the coordinator shall consider the device disassociated.

If an associated device wants to leave the PAN, it shall send a disassociation notification command to its coordinator. If the disassociation notification command is received correctly by the coordinator, it shall confirm its receipt by sending an acknowledgment frame. Even if the acknowledgment is not received, the device shall consider itself disassociated.

If the source address contained in the disassociation notification command is equal to *macCoordExtendedAddress*, the recipient shall consider itself disassociated. If the command is received by a coordinator and the source is not equal to *macCoordExtendedAddress*, it shall verify that the source address corresponds to one of its associated devices; if so, the coordinator shall consider the device disassociated. If none of the above conditions is satisfied, the command shall be ignored.

An associated device shall disassociate itself by removing all references to the PAN. A coordinator shall disassociate a device by removing all references to that device.

The next higher layer of the requesting device shall be notified of the result of the disassociation procedure through the MLME-DISASSOCIATE.confirm primitive.

7.5.4 Synchronization

This subclause specifies the procedures for coordinators to generate beacon frames and for devices to synchronize with a coordinator. For PANs supporting beacons, synchronization is performed by receiving and decoding the beacon frames. For PANs not supporting beacons, synchronization is performed by polling the coordinator for data.

7.5.4.1 Synchronization with beacons

All devices operating on a beacon-enabled PAN (i.e., *macBeaconOrder* < 15) shall be able to acquire beacon sychronization in order to detect any pending messages or to track the beacon. Devices shall be permitted to acquire beacon synchronization only with beacons containing the PAN identifier specified in *macPANId*. If *macPANId* specifies the broadcast PAN identifier (0 x ffff), a device shall not attempt to acquire beacon synchronization.

A device is instructed to attempt to acquire the beacon through the MLME-SYNC.request primitive. If tracking is specified in the MLME-SYNC.request primitive, the device shall attempt to acquire the beacon and keep track of it by regular and timely activation of its receiver. If tracking is not specified, the device shall attempt to acquire the beacon only once or terminate the tracking after the next beacon if tracking was enabled through a previous request.

To acquire beacon synchronization, a device shall enable its receiver and search for at most [*aBaseSuperframeDuration* * (2^n + 1)] symbols, where *n* is the value of *macBeaconOrder*. If a beacon frame containing the current PAN identifier of the device is not received, the MLME shall repeat this search. Once the number of missed beacons reaches *aMaxLostBeacons*, the MLME shall notify the next higher layer by issuing the MLME-SYNC-LOSS.indication primitive with a loss reason of BEACON_LOSS.

The MLME shall timestamp each received beacon frame at the same symbol boundary within each frame, the location of which is implementation specific. The symbol boundary shall be chosen to be the same as that used in the timestamp of the outgoing beacon frame, stored in *macBeaconTxTime*.

If the security enabled subfield is set to 1, the MLME shall process the received beacon frame for security, as described in 7.5.8. If the secure processing fails, the frame shall be discarded, and the MLME shall issue the MLME-COMM-STATUS.indication primitive indicating the error.

If a beacon frame is received, the device shall verify that the beacon frame came from the coordinator with which it associated. Consequently, if the source address and the source PAN identifier fields of the MAR of the beacon frame do not match the coordinator source address (*macCoordShortAddress* or *macCoordExtendedAddress*, depending on the addressing mode) and the PAN identifier of the device (*macPANId*), the MLME shall discard the beacon frame.

If a valid beacon frame is received and *macAutoRequest* is set to FALSE, the MLME shall indicate the beacon parameters to the next higher layer by issuing the MLME-BEACON-NOTIFY.indication primitive. If a beacon frame is received and *macAutoRequest* is set to TRUE, the MLME shall first issue the MLME-BEACON-NOTIFY.indication primitive, if the beacon contains any payload. The MLME shall then compare its address with those addresses in the address list field of the beacon frame. If the address list field contains the short or extended address of the device and the source PAN identifier matches *macPANId*, the MLME shall follow the procedure for extracting pending data from the coordinator (see 7.5.6.3).

If beacon tracking is activated, the MLME shall enable its receiver at a time prior to the next expected beacon frame transmission, i.e., just before the known start of the next superframe. If the number of consecutive beacons missed by the MLME reaches *aMaxLostBeacons*, the MLME shall respond with the MLME-SYNC-LOSS.indication primitive with a loss reason of BEACON_LOST.

7.5.4.2 Synchronization without beacons

All devices operating on a nonbeacon-enabled PAN (*macBeaconOrder* = 15) shall be able to poll the coordinator for data at the discretion of the next higher layer.

A device is instructed to poll the coordinator when the MLME receives the MLME-POLL.request primitive. On receipt of this primitive, the MLME shall follow the procedure for extracting pending data from the coordinator (see 7.5.6.3).

7.5.4.3 Orphaned device realignment

If the next higher layer receives repeated communications failures following its requests to transmit data, it may conclude that it has been orphaned. A single communications failure occurs when a device transaction fails to reach the coordinator, i.e., an acknowledgment is not received after *aMaxFrameRetries* attempts at sending the data. If the next higher layer concludes that it has been orphaned, it may instruct the MLME to either perform the orphaned device realignment procedure, or to reset the MAC sublayer and then perform the association procedure.

If the decision has been made by the next higher layer to perform the orphaned device realignment procedure, it shall issue an MLME-SCAN.request with the ScanType parameter set to orphan scan and the ScanChannel parameter containing the list of channels to be scanned. Upon receiving this primitive, the MAC sublayer shall begin an orphan scan, as described in 7.5.2.1.4.

If the orphan scan is successful (i.e., its PAN has been located), the device shall update its MAC PIB with the PAN information contained in the coordinator realignment command (see 7.3.2.5). If the orphan scan was unsuccessful, the next higher layer shall decide what further action needs be taken, e.g., to retry the orphan scan or attempt to reassociate.

7.5.5 Transaction handling

Because IEEE Std 802.15.4-2003 favors very low cost devices that, in general, will be battery powered, transactions can be instigated from the devices themselves rather than from the coordinator. In other words, either the coordinator needs to indicate in its beacon when messages are pending for devices or the devices themselves need to poll the coordinator to determine whether they have any messages pending. Such transfers are called indirect transmissions.

The coordinator shall begin handling a transaction on receipt of an indirect transmission request either via the MCPS-DATA.request primitive or via a request from the MLME to send a MAC command instigated by a primitive from the next higher layer, such as the MLME-ASSOCIATE.response primitive (see 7.1.3.3). On completion of the transaction, the MAC sublayer shall indicate a status value to the next higher layer. If a request primitive instigated the indirect transmission, the corresponding confirm primitive shall be used to convey the appropriate status value. Conversely, if a response primitive instigated the indirect transmission, the MLME-COMM-STATUS.indication primitive shall be used to convey the appropriate status value.

The information contained in the indirect transmission request forms a transaction, and the coordinator shall be capable of storing at least one transaction. On receipt of an indirect transmission request, if there is no capacity to store another transaction, the MAC sublayer shall indicate to the next higher layer a status of TRANSACTION_OVERFLOW in the MLME-COMM-STATUS.indication primitive.

If the coordinator is capable of storing more than one transaction, it shall ensure that all the transactions for the same device are sent in the order in which they arrived at the MAC sublayer. For each transaction sent, if another exists for the same device, the MAC sublayer shall set its frame pending subfield to 1, indicating the additional pending data.

Each transaction shall persist in the coordinator for at most *macTransactionPersistenceTime*. If the transaction is not extracted by the appropriate device within this time, the transaction information shall be discarded and the MAC sublayer shall indicate to the next higher layer a status of TRANSACTION_EXPIRED in the MLME-COMM-STATUS.indication primitive.

If the coordinator transmits beacons, it shall list the addresses of the devices to which each transaction is associated in the address list field and indicate the number of addresses in the pending address specification field of the beacon frame. If the coordinator is able to store more than seven pending transactions, it shall indicate them in its beacon on a first-come-first-served basis, ensuring that the beacon frame contains at most seven addresses. For transactions requiring a GTS, the PAN coordinator shall not add the address of the recipient to its list of pending addresses in the beacon frame. Instead it shall transmit the transaction in the GTS allocated for the device (see 7.5.7.3).

On a beacon-enabled PAN, a device that receives a beacon containing its address in the list of pending addresses shall attempt to extract the data from the coordinator. On a nonbeacon-enabled PAN, a device shall attempt to extract the data from the coordinator on receipt of the MLME-POLL.request primitive. The procedure for extracting pending data from the coordinator is described in clause 7.5.6.3.

When the transaction is complete, its data shall be discarded and an indication of the result of the data transfer shall be sent to the next higher layer. If the transaction required an acknowledgment and an acknowledgment was not received, the MAC sublayer shall indicate a status of NO_ACK. If the transaction was successful, the MAC sublayer shall indicate a status of SUCCESS.

7.5.6 Transmission, reception, and acknowledgment

This subclause describes the fundamental procedures for transmission, reception, and acknowledgment.

7.5.6.1 Transmission

Each time a data or a MAC command frame is generated, the MAC sublayer shall copy the value of *macDSN* into the sequence number field of the MHR of the outgoing frame and then increment it by one. Similarly, each time a beacon frame is generated, the MAC sublayer shall copy the value of *macBSN* into the sequence number field of the MHR of the outgoing frame and then increment it by one.

The source address field, if present, shall contain the address of the device sending the frame. When a device has associated and has been allocated a short address (i.e., *macShortAddress* is not equal to 0 x fffe or 0 x ffff), it shall use that address in preference to its 64 bit extended address (i.e., *aExtendedAddress*) wherever possible. When a device has not yet associated to a PAN or *macShortAddress* is equal to 0 x ffff, it shall use its 64 bit extended address in all communications requiring the source address field. If the source address field is not present, the originator of the frame shall be assumed to be the PAN coordinator, and the destination address field shall contain the address of the recipient.

The destination address field, if present, shall contain the address of the intended recipient of the frame, which may be either a 16 bit short address or a 64 bit extended address. If the destination address field is not present, the recipient of the frame shall be assumed to be the PAN coordinator, and the source address field shall contain the address of the originator.

If both destination and source addressing information is present, the MAC sublayer shall compare the destination and source PAN identifiers. If the PAN identifiers are identical, the intra-PAN subfield of the frame control field shall be set to 1, and the source PAN identifier shall be omitted from the transmitted frame. If the PAN identifiers are different, the intra-PAN subfield of the frame control field shall be set to 0, and both destination and source PAN identifier fields shall be included in the transmitted frame.

If the frame is to be transmitted on a beacon-enabled PAN, the transmitting device shall attempt to find the beacon before transmitting. If the beacon is not being tracked (see 7.5.4.1) and hence the device does not know where the beacon will appear, it shall enable its receiver and search for at most [*aBaseSuperframeDuration* * (2^n + 1)] symbols, where *n* is the value of *macBeaconOrder*, in order to find the beacon. If the beacon is not found after this time, the device shall transmit the frame following the successful application of the unslotted version of the CSMA-CA algorithm (see 7.5.1.3). Once the beacon has been found, either after a search or due to its being tracked, the frame shall be transmitted in the appropriate portion of the superframe. Transmissions in the CAP shall follow a successful application of the slotted version of the CSMA-CA algorithm (see 7.5.1.3), and transmissions in a GTS shall not use CSMA-CA.

If the frame is to be transmitted on a nonbeacon-enabled PAN, the frame shall be transmitted following the successful application of the unslotted version of the CSMA-CA algorithm (see 7.5.1.3).

7.5.6.2 Reception and rejection

Each device may choose whether the MAC sublayer is to enable its receiver during idle periods. During these idle periods, the MAC sublayer shall still service transceiver task requests from the next higher layer. A transceiver task shall be defined as a transmission request with acknowledgment reception, if required, or a reception request. On completion of each transceiver task, the MAC sublayer shall request that the PHY enables or disables its receiver, depending on whether *macRxOnWhenIdle* is set to TRUE or FALSE, respectively. If *macBeaconOrder* is less than 15, the value of *macRxOnWhenIdle* shall be considered only during idle periods of the CAP.

Due to the nature of radio communications, a device with its receiver enabled will be able to receive and decode transmissions from all devices complying with IEEE Std 802.15.4-2003 that are currently operating on the same channel and are in its POS, along with interference from other sources. The MAC sublayer shall, therefore, be able to filter incoming frames and present only the frames that are of interest to the upper layers.

For the first level of filtering, the MAC sublayer shall discard all received frames that do not contain a correct value in their FCS field in the MFR, according to the algorithm described in 7.2.1.8. The next level of filtering shall be dependent on whether the MAC sublayer is currently operating in promiscuous mode. In promiscuous mode, the MAC sublayer shall pass all frames received after the first filter directly to the upper layers without applying any more filtering. The MAC sublayer shall be in promiscuous mode if *macPromiscuousMode* is set to TRUE.

If the MAC sublayer is not in promiscuous mode (i.e., *macPromiscuousMode* is set to FALSE), it shall accept only frames that satisfy all of the following requirements:

- — The frame type subfield of the frame control field shall not contain an illegal frame type.
- — If the frame type indicates that the frame is a beacon frame, the source PAN identifier shall match *macPANId* unless *macPANId* is equal to 0 x ffff, in which case the beacon frame shall be accepted regardless of the source PAN identifier.
- — If a destination PAN identifier is included in the frame, it shall match *macPANId* or shall be the broadcast PAN identifier (0 x ffff).
- — If a short destination address is included in the frame, it shall match either *macShortAddress* or the broadcast address (0 x ffff). Otherwise, if an extended destination address is included in the frame, it shall match *aExtendedAddress*.
- — If only source addressing fields are included in a data or MAC command frame, the frame shall be accepted only if the device is a PAN coordinator and the source PAN identifier matches *macPANId*.

If any of the requirements listed above are not satisfied, the MAC sublayer shall discard the incoming frame. If all of the requirements listed above are satisfied, the frame shall be considered valid and processed further. For valid frames, if the frame type subfield indicates a data or MAC command frame and the acknowledgment request subfield of the frame control field is set to 1, the MAC sublayer shall send an acknowledgment frame. Prior to the transmission of the acknowledgment frame, the sequence number included in the received data or MAC command frame shall be copied into the sequence number field of the acknowledgment frame. This step will allow the transaction originator to know that it has received the appropriate acknowledgment frame.

If the security enabled subfield is set to 1, the MAC sublayer shall process the received frame for security, as described in 7.5.8. During either an active or a passive scan for beacons, information contained in the received beacon frames that fail secure processing will still be put into a PAN descriptor, with the Security-Use, ACLEntry, and SecurityFailure fields (see Table 41) set accordingly, and passed to the next higher layer (see 7.5.2.1).

If the intra-PAN subfield of the frame control field is set to 1 and both destination and source addressing information is included in the frame, the MAC sublayer shall assume that the omitted source PAN identifier field is identical to the destination PAN identifier field.

If the frame was successfully processed, the MAC sublayer shall pass the frame to the next higher layer. This is achieved by issuing the MCPS-DATA.indication primitive containing the frame information.

7.5.6.3 Extracting pending data from a coordinator

A device on a beacon-enabled PAN can determine whether any frames are pending for it by examining the the contents of the received beacon frame, as described in 7.5.4.1. If the address of the device is contained in the address list field of the beacon frame, the MLME of the device shall send a data request command (see 7.3.2.1) to the coordinator during the CAP with the acknowledgment request subfield of the frame control field set to 1. There are two other cases for which the MLME shall send a data request command to the coordinator. The first case is when the MLME receives the MLME-POLL.request primitive. In the second case, a device may send a data request command *aResponseWaitTime* symbols after the acknowledgment to a request command, such as during the association procedure. The data request command shall contain the destination address information only if it is not intended for the PAN coordinator.

On successfully receiving a data request command, the coordinator shall send an acknowledgment frame, thus confirming its receipt. If the coordinator has enough time to determine whether the device has a frame pending and is still able to send the acknowledgment frame within *macAckWaitDuration* symbols, it shall set the frame pending subfield of the frame control field of the acknowledgment frame accordingly to indicate whether a frame is actually pending for the device. If this is not possible, the coordinator shall set the frame pending subfield of the acknowledgment frame to 1.

On receipt of the acknowledgment frame with the frame pending subfield set to 0, the device shall conclude that there are no data pending at the coordinator.

On receipt of the acknowledgment frame with the frame pending subfield set to 1, a device shall enable its receiver for at most *aMaxFrameResponseTime* CAP symbols in a beacon-enabled PAN, or symbols in a nonbeacon-enabled PAN, to receive the corresponding data frame from the coordinator. If there is an actual data frame pending within the coordinator for the requesting device, the coordinator shall send the frame to the device using one of the mechanisms described below in 7.5.6.3. If there is no data frame pending for the requesting device, the coordinator shall send a data frame without requesting acknowledgment to the device containing a zero length payload, indicating that no data are present, using one of the mechanisms described below in 7.5.6.3.

The data frame following the acknowledgment of the data request command shall be transmitted using one of the following mechanisms:

— Without using CSMA-CA, if the MAC sublayer can commence transmission of the data frame between *aTurnaroundTime* and (*aTurnaroundTime* + *aUnitBackoffPeriod*) symbols, on a backoff slot boundary, and there is time remaining in the CAP for the message, appropriate IFS, and acknowledgment. If a requested acknowledgment frame is not received following this data frame, all subsequent retransmissions shall be transmitted using CSMA-CA. The constant *aTurnaroundTime* is defined in Table 18 (in 6.4.1).
— Using CSMA-CA, otherwise.

If the requesting device does not receive a data frame from the coordinator within *aMaxFrameResponseTime* CAP symbols in a beacon-enabled PAN, or symbols in a nonbeacon-enabled PAN, or if the requesting device receives a data frame from the coordinator with a zero length payload, it shall conclude that there are no data pending at the coordinator. If the requesting device does receive a data frame from the coordinator, it shall send an acknowledgment frame, if requested, thus confirming receipt.

If the frame pending subfield of the frame control field of the data frame received from the coordinator is set to 1, the device still has more data pending with the coordinator. In this case it may extract the data by sending a new data request command to the coordinator, using the same procedure described above in 7.5.6.3.

7.5.6.4 Use of acknowledgments

A data or MAC command frame shall be sent with the acknowledgment request subfield of its frame control field set appropriately for the frame. A beacon or acknowledgment frame shall always be sent with the acknowledgment request subfield set to 0. Similarly, any frame that is broadcast shall be sent with its acknowledgment request subfield set to 0.

7.5.6.4.1 No acknowledgment

A frame transmitted with its acknowledgment request subfield set to 0 shall not be acknowledged by its intended recipient. The originating device shall assume that the transmission of the frame was successful.

The message sequence chart in Figure 62 shows the scenario for transmitting a single frame of data from an originator to a recipient without requiring an acknowledgment. In this case, the originator transmits the data frame with the acknowledgement request (AR) subfield equal to 0.

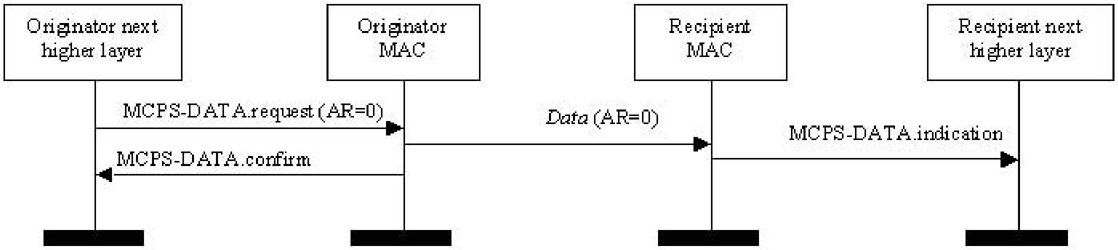


Figure 62—Successful data transmission without an acknowledgment

7.5.6.4.2 Acknowledgment

A frame transmitted with the acknowledgment request subfield of its frame control field set to 1 shall be acknowledged by the recipient. If the intended recipient correctly receives the frame, it shall generate and send an acknowledgment frame containing the same DSN from the data or MAC command frame that is being acknowledged.

The transmission of an acknowledgment frame in a nonbeacon-enabled PAN or in the CFP shall commence *aTurnaroundTime* symbols after the reception of the last symbol of the data or MAC command frame. The transmission of an acknowledgment frame in the CAP shall commence at a backoff slot boundary. In this case, the transmission of an acknowledgment frame shall commence between *aTurnaroundTime* and (*aTurnaroundTime* + *aUnitBackoffPeriod*) symbols after the reception of the last symbol of the data or MAC command frame. The constant *aTurnaroundTime* is defined in Table 18 (in 6.4.1).

The message sequence chart in Figure 63 shows the scenario for transmitting a single frame of data from an originator to a recipient with an acknowledgment. In this case, the originator indicates to the recipient that it requires an acknowledgment by transmitting the data frame with the acknowledgement request (AR) subfield set to 1.

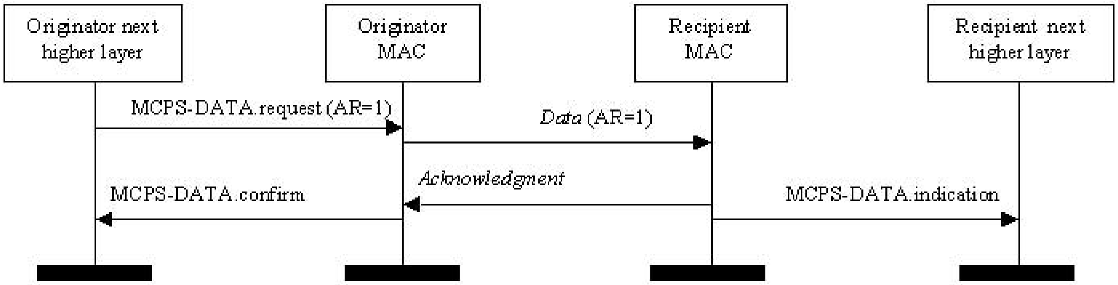


Figure 63—Successful data transmission with an acknowledgment

7.5.6.5 Retransmissions

A device that sends a frame with the acknowledgment request subfield of its frame control field set to 0 shall assume that the transmission was successfully received and shall hence not perform the retransmission procedure.

A device that sends a data or MAC command frame with its acknowledgment request subfield set to 1 shall wait for at most *macAckWaitDuration* symbols for the corresponding acknowledgment frame to be received. If an acknowledgment frame is received within *macAckWaitDuration* symbols and contains the same DSN as the original transmission, the transmission is considered successful, and no further action shall be taken by the device. If an acknowledgment is not received within *macAckWaitDuration* symbols or an acknowledgment is received containing a DSN that was not the same as the original transmission, the device shall conclude that the single transmission attempt has failed.

If a single transmission attempt has failed and the transmission was indirect, the coordinator shall not retransmit the data or MAC command frame. Instead, the frame shall remain in the transaction queue of the coordinator. If a single transmission attempt has failed and the transmission was direct, the device shall repeat the process of transmitting the data or MAC command frame and waiting for the acknowledgment, up to a maximum of *aMaxFrameRetries* times. Each retransmission shall only be attempted if it can be completed within the same portion of the superframe, i.e., the CAP or a GTS in which the original transmission was attempted. If this timing is not possible, the retransmission shall be deferred until the same portion in the next superframe.

If an acknowledgment is still not received after *aMaxFrameRetries* retransmissions, the MAC sublayer shall assume the transmission has failed and notify the next higher layer of the failure. This situation eventuality is referred to as a communications failure.

7.5.6.6 Promiscuous mode

A device may activate promiscuous mode by setting *macPromiscuousMode*. If the MLME is requested to set *macPromiscuousMode* to TRUE, the MLME shall also set *macRxOnWhenIdle* to TRUE and then request that the PHY enable its receiver. This request is achieved when the MLME issues the PLME-SET-TRX-STATE.request primitive with a state of RX_ON.

If the MLME is requested to set *macPromiscuousMode* to FALSE, the MLME shall also set *macRxOnWhenIdle* to FALSE and then request that the PHY disable its receiver. This is achieved by the MLME issuing the PLME-SET-TRX-STATE.request primitive (see 6.2.2.7) with a state of TRX_OFF.

7.5.6.7 Transmission scenarios

Due to the imperfect nature of the radio medium, a transmitted frame does not always reach its intended destination. Figure 64 illustrates three different data transmission scenarios:

— *Successful data transmission*. The originator MAC sublayer transmits the data frame to the recipient via the PHY data service. In waiting for an acknowledgment, the originator MAC sublayer starts a timer that will expire after *macAckWaitDuration* symbols. The recipient MAC sublayer receives the data frame, sends an acknowledgment back to the originator, and passes the frame to the next higher layer. The originator MAC sublayer receives the acknowledgment from the recipient before its timer expires and then disables and resets the timer. The data transfer is now complete, and the originator MAC sublayer issues a success confirmation to the next higher layer.

— *Lost data frame*. The originator MAC sublayer transmits the data frame to the recipient via the PHY data service. In waiting for an acknowledgment, the originator MAC sublayer starts a timer that will expire after *macAckWaitDuration* symbols. The recipient MAC sublayer does not receive the data frame and so does not respond with an acknowledgment. The timer of the originator MAC sublayer expires before an acknowledgment is received. The data transfer has failed and the originator retransmits the data. This sequence may be repeated up to a maximum of *aMaxFrameRetries* times. If a data transfer attempt fails a total of (1 + *aMaxFrameRetries*) times, the originator MAC sublayer will issue a failure confirmation to the next higher layer.

— *Lost acknowledgment frame*. The originator MAC sublayer transmits the data frame to the recipient via the PHY data service. In waiting for an acknowledgment, the originator MAC sublayer starts a timer that will expire after *macAckWaitDuration* symbols. The recipient MAC sublayer receives the data frame, sends an acknowledgment back to the originator, and passes the frame to the next higher layer. The originator MAC sublayer does not receive the acknowledgment frame and its timer expires. The data transfer has failed, and the originator will retransmit the data. This sequence may be repeated up to a maximum of *aMaxFrameRetries* times. If a data transfer attempt fails a total of (1 + *aMaxFrameRetries*) times, the MAC sublayer will issue a failure confirmation to the next higher layer.

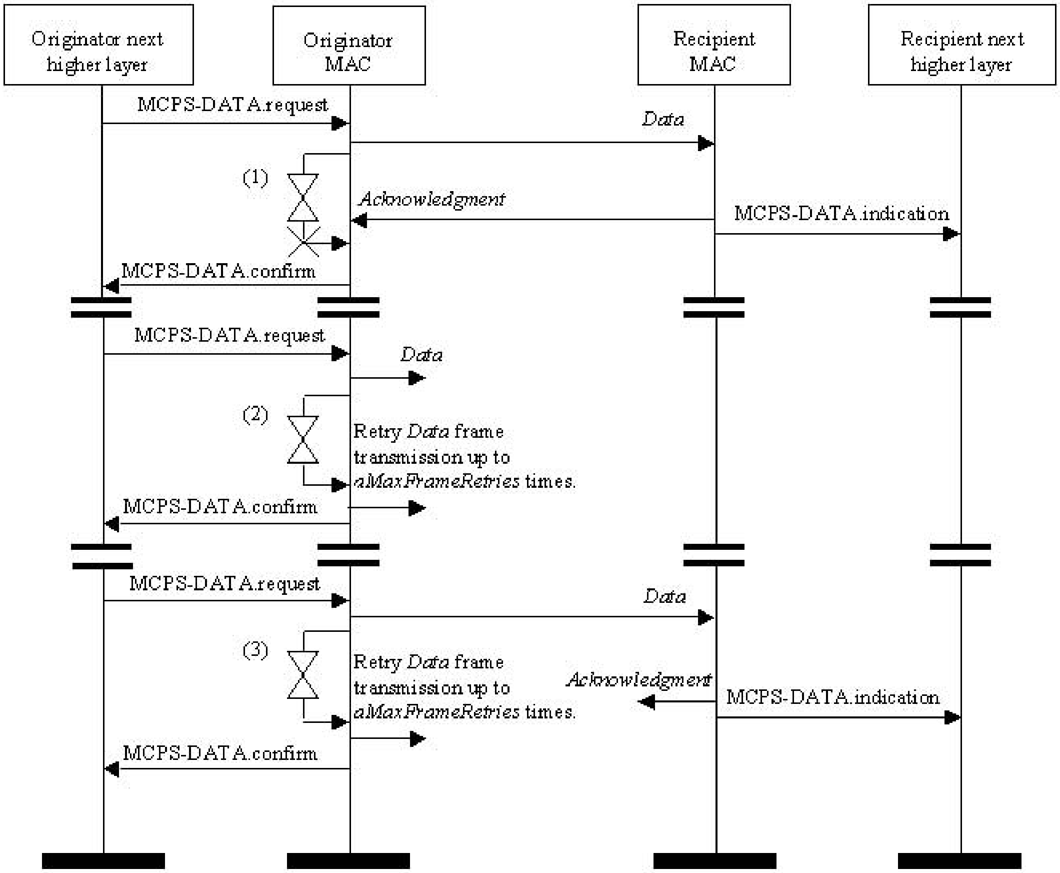


Figure 64—Transmission scenarios for frame reliability

7.5.7 GTS allocation and management

A GTS allows a device to operate on the channel within a portion of the superframe that is dedicated (on the PAN) exclusively to that device. A GTS shall be allocated only by the PAN coordinator, and it shall be used only for communications between the PAN coordinator and a device. A single GTS may extend over one or more superframe slots. The PAN coordinator may allocate up to seven GTSs at the same time, provided there is sufficient capacity in the superframe.

A GTS shall be allocated before use, with the PAN coordinator deciding whether to allocate a GTS based on the requirements of the GTS request and the current available capacity in the superframe. GTSs shall be allocated on a first-come-first-served basis, and all GTSs shall be placed contiguously at the end of the superframe and after the CAP. Each GTS shall be deallocated when the GTS is no longer required, and a GTS can be deallocated at any time at the discretion of the PAN coordinator or by the device that originally requested the GTS. A device that has been allocated a GTS may also operate in the CAP.

A data frame transmitted in an allocated GTS shall use only short addressing.

The management of GTSs shall be undertaken by the PAN coordinator only. To facilitate GTS management, the PAN coordinator shall be able to store all the information necessary to manage seven GTSs. For each GTS, the PAN coordinator shall be able to store its starting slot, length, direction, and associated device address.

The GTS direction, which is relative to the data flow from the device that owns the GTS, is specified as either transmit or receive. The device address and direction shall, therefore, uniquely identify each GTS. Each device may request one transmit GTS and/or one receive GTS. For each allocated GTS, the device shall be able to store its starting slot, length, and direction. If a device has been allocated a receive GTS, it shall enable its receiver for the entirety of the GTS. In the same way, a PAN coordinator shall enable its receiver for the entirety of the GTS if a device has been allocated a transmit GTS. If a data frame is received during a receive GTS and an acknowledgment is requested, the device shall transmit the acknowledgment frame as usual. Similarly, a device shall be able to receive an acknowledgment frame during a transmit GTS.

A device shall attempt to allocate and use a GTS only if it is currently tracking the beacons. The MLME is instructed to track beacons by issuing the MLME-SYNC.request primitive with the TrackBeacon parameter set to TRUE. If a device loses synchronization with the PAN coordinator, all its GTS allocations shall be lost.

The use of GTSs by an RFD is optional.

7.5.7.1 CAP maintenance

The PAN coordinator shall preserve the minimum CAP length of *aMinCAPLength* and take preventative action if the minimum CAP is not satisfied. However, an exception shall be allowed for the accommodation of the temporary increase in the beacon frame length needed to perform GTS maintenance. If preventative action becomes necessary, the action chosen is left up to the implementation, but may include one or more of the following:

- Limiting the number of pending addresses included in the beacon.
- Not including a payload field in the beacon frame.
- Deallocating one or more of the GTSs

7.5.7.2 GTS allocation

A device is instructed to request the allocation of a new GTS through the MLME-GTS.request primitive, with GTS characteristics set according to the requirements of the intended application.

To request the allocation of a new GTS, the MLME shall send the GTS request command (see 7.3.3.1) to the PAN coordinator. The characteristics type subfield of the GTS characteristics field of the request shall be set to 1 (GTS allocation), and the length and direction subfields shall be set according to the desired characteristics of the required GTS. If the GTS request command is received correctly, the PAN coordinator shall send an acknowledgment frame, thus confirming receipt.

On receipt of a GTS request command indicating a GTS allocation request, the PAN coordinator shall first check if there is available capacity in the current superframe, based on the remaining length of the CAP and the desired length of the requested GTS. The superframe shall have available capacity if the maximum number of GTSs has not been reached and allocating a GTS of the desired length would not reduce the length of the CAP to less than *aMinCAPLength*. GTSs shall be allocated on a first-come-first-served basis by the PAN coordinator provided there is sufficient bandwidth available. The PAN coordinator shall make this decision within *aGTSDescPersistenceTime* superframes.

On receipt of the acknowledgment to the GTS request command, the device shall continue to track beacons and wait for at most *aGTSDescPersistenceTime* superframes. If no GTS descriptor for the device appears in the beacon within this time, the MLME of the device shall notify the next higher layer of the failure. This notification is achieved when the MLME issues the MLME-GTS.confirm primitive (see 7.1.7.2) with a status of NO_DATA.

When the PAN coordinator determines whether capacity is available for the requested GTS, it shall generate a GTS descriptor with the requested specifications and the short address of the requesting device. If the GTS was allocated successfully, the PAN coordinator shall set the start slot in the GTS descriptor to the superframe slot at which the GTS begins and the length in the GTS descriptor to the length of the GTS. In addition, the PAN coordinator shall notify the next higher layer of the new GTS. This notification is achieved when the MLME of the PAN coordinator issues the MLME-GTS.indication primitive (see 7.1.7.3) with the characteristics of the allocated GTS. If there was not sufficient capacity to allocate the requested GTS, the start slot shall be set to 0 and the length to the largest GTS length that can currently be supported. The PAN coordinator shall then include this GTS descriptor in its beacon and update the GTS specification field of the beacon frame accordingly. The PAN coordinator shall also update the final CAP slot subfield of the superframe specification field of the beacon frame, indicating the final superframe slot utilized by the decreased CAP. The GTS descriptor shall remain in the beacon frame for *aGTSDescPersistenceTime* superframes, after which it shall be removed automatically. The PAN coordinator shall be allowed to reduce its CAP below *aMinCAPLength* to accommodate the temporary increase in the beacon frame length due to the inclusion of the GTS descriptor.

On receipt of a beacon frame containing a GTS descriptor corresponding to *macShortAddress*, the device shall process the descriptor. The MLME of the device shall then notify the next higher layer of whether the GTS allocation request was successful. This notification is achieved when the MLME issues the MLME-GTS.confirm primitive with a status of SUCCESS (if the start slot in the GTS descriptor was greater than zero) or DENIED (if the start slot was equal to zero or if the length did not match the requested length).

7.5.7.3 GTS usage

When the MAC sublayer of a device receives an MCPS-DATA.request primitive (see 7.1.1.1) with the TxOptions parameter indicating a GTS transmission, it shall first determine whether it has a valid GTS. If the device is a PAN coordinator, it shall determine whether it has a receive GTS corresponding to the device with the requested destination address. If the device is not a PAN coordinator, it shall determine whether a transmit GTS has been allocated. If a valid GTS is found, the MAC sublayer shall transmit the data during the GTS, i.e., between its starting slot and its starting slot plus its length. At this time, the MAC sublayer shall transmit the MPDU immediately without using CSMA-CA, provided the requested transaction can be completed before the end of the GTS. If the requested transaction cannot be completed before the end of the current GTS, the MAC sublayer shall defer the transmission until the specified GTS in the next superframe.

If the device has any receive GTSs, the MAC sublayer of the device shall ensure that the receiver is enabled at a time prior to the start of the GTS and for the duration of the GTS, as indicated by its starting slot and its length. The PAN coordinator shall send all frames within a receive GTS with the acknowledgment request subfield of the frame control field set to 1.

When the MAC sublayer of the PAN coordinator receives a MCPS-DATA.request primitive with the TxOptions parameter indicating a GTS transmission, it shall defer the transmission until the start of the receive GTS of the intended recipient. In this case, the address of the device with the message requiring a GTS transmission shall not be added to the list of pending addresses in the beacon frame (see 7.5.5). For all allocated transmit GTSs (relative to the device), the MAC sublayer of the PAN coordinator shall ensure that its receiver is enabled at a time prior to the start and for the duration of each GTS.

Before commencing transmission in a GTS, each device shall ensure that the data transmission, the acknowledgment, if requested, and the IFS, suitable to the size of the data frame, can be completed before the end of the GTS.

If a device misses the beacon at the beginning of a superframe, it shall not use its GTSs until it receives a subsequent beacon correctly. If a loss of synchronization occurs due to the loss of the beacon, the device shall consider all of its GTSs deallocated.

7.5.7.4 GTS deallocation

A device is instructed to request the deallocation of an existing GTS through the MLME-GTS.request primitive (see 7.1.7.1), using the characteristics of the GTS it wishes to deallocate. From this point onward, the GTS to be de-allocated shall not be used by the device, and its stored characteristics shall be reset.

To request the deallocation of an existing GTS, the MLME shall send the GTS request command (see 7.3.3.1) to the PAN coordinator. The characteristics type subfield of the GTS characteristics field of the request shall be set to 0 (i.e., GTS deallocation), and the length and direction subfields shall be set according to the characteristics of the GTS to deallocate. If the GTS request command is received correctly, the PAN coordinator shall send an acknowledgment frame, thus confirming receipt. On receipt of the acknowledgment to the GTS request command, the MLME shall notify the next higher layer of the deallocation. This notification is achieved when the MLME issues the MLME-GTS.confirm primitive (see 7.1.7.2) with a status of SUCCESS and a GTSCharacteristics parameter with its characteristics type subfield set to 0. If the GTS request command is not received correctly by the PAN coordinator, it shall determine that the device has stopped using its GTS by the procedure described in 7.5.7.6.

On receipt of a GTS request command with the characteristics type subfield of the GTS characteristics field set to 0 (GTS deallocation), the PAN coordinator shall attempt to deallocate the GTS. If the GTS characteristics contained in the GTS request command do not match the characteristics of a known GTS, the PAN coordinator shall ignore the request. If the GTS characteristics contained in the GTS request command match the characteristics of a known GTS, the MLME of the PAN coordinator shall deallocate the specified GTS and notify the next higher layer of the change. This notification is achieved when the MLME issues the MLME-GTS.indication primitive (see 7.1.7.3) with a GTSCharacteristics parameter containing the characteristics of the deallocated GTS and a characteristics type subfield set to 0. The PAN coordinator shall also update the final CAP slot subfield of the superframe specification field of the beacon frame, indicating the final superframe slot utilized by the increased CAP. It shall not add a descriptor to the beacon frame to describe the deallocation.

When a GTS deallocation is initiated by the PAN coordinator, the MLME shall notify the next higher layer of the change. This notification is achieved when the MLME issues the MLME-GTS.indication primitive with a GTSCharacteristics parameter containing the characteristics of the deallocated GTS and a characteristics type subfield set to 0. The PAN coordinator shall then deallocate the GTS and add a GTS descriptor into its beacon frame corresponding to the deallocated GTS, but with its starting slot set to 0. The descriptor shall remain in the beacon frame for *aGTSDescPersistenceTime* superframes. The PAN coordinator shall be allowed to reduce its CAP below *aMinCAPLength* to accommodate the temporary increase in the beacon frame length due to the inclusion of the GTS descriptor.

On receipt of a beacon frame containing a GTS descriptor corresponding to *macShortAddress* and a start slot equal to 0, the device shall immediately stop using the GTS. The MLME of the device shall then notify the next higher layer of the deallocation. This notification is achieved when the MLME issues the MLME-GTS.indication primitive with a GTSCharacteristics parameter containing the characteristics of the deallocated GTS and a characteristics type subfield set to 0.

7.5.7.5 GTS reallocation

The deallocation of a GTS may result in the superframe becoming fragmented. For example, Figure 65 shows three stages of a superframe with allocated GTSs. In stage 1, three GTSs are allocated starting at slots 14, 10, and 8, respectively. If GTS 2 is now deallocated (stage 2), there will be a gap in the superframe during which nothing can happen. To solve this, GTS 3 will have to be shifted to fill the gap, thus increasing the size of the CAP (stage 3).

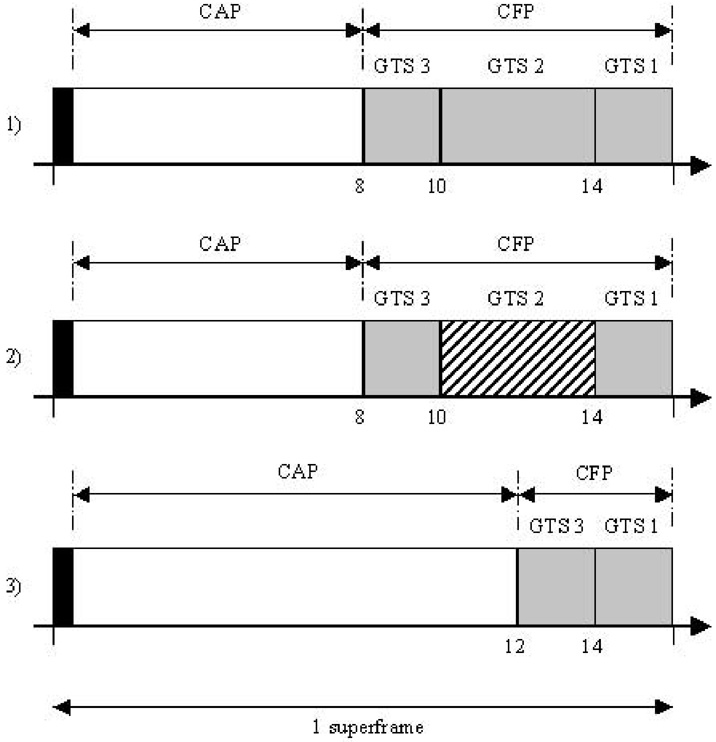


Figure 65—CFP defragmentation on GTS deallocations

The PAN coordinator shall ensure that any gaps occurring in the CFP, appearing due to the deallocation of a GTS, are removed to maximize the length of the CAP.

When a GTS is deallocated by the PAN coordinator, it shall add a GTS descriptor into its beacon frame indicating that the GTS has been deallocated. If the deallocation is initiated by a device, the PAN coordinator shall not add a GTS descriptor into its beacon frame to indicate the deallocation. For each device with an allocated GTS having a starting slot lower than the GTS being deallocated, the PAN coordinator shall update the GTS with the new starting slot and add a GTS descriptor to its beacon corresponding to this adjusted GTS. The new starting slot is computed so that no space is left between this GTS and either the end of the CFP, if the GTS appears at the end of the CFP, or the start of the next GTS in the CFP.

In situations where multiple reallocations occur at the same time, the PAN coordinator may choose to perform the reallocation in stages. The PAN coordinator shall keep each GTS descriptor in its beacon for *aGTSDescPersistenceTime* superframes.

On receipt of a beacon frame containing a GTS descriptor corresponding to *macShortAddress* and a direction and length corresponding to one of its GTSs, the device shall adjust the starting slot of the GTS corresponding to the GTS descriptor and start using it immediately.

In cases where it is necessary for the PAN coordinator to include a GTS descriptor in its beacon, it shall be allowed to reduce its CAP below *aMinCAPLength* to accommodate the temporary increase in the beacon frame length. After *aGTSDescPersistenceTime* superframes, the PAN coordinator shall remove the GTS descriptor from the beacon.

7.5.7.6 GTS expiration

The MLME of the PAN coordinator shall attempt to detect when a device has stopped using a GTS using the following rules:

— For a transmit GTS, the MLME of the PAN coordinator shall assume that a device is no longer using its GTS if a data frame is not received from the device in the GTS at least every 2**n* superframes, where *n* is defined below.

— For receive GTSs, the MLME of the PAN coordinator shall assume that a device is no longer using its GTS if an acknowledgment frame is not received from the device at least every 2**n* superframes, where *n* is defined below.

The value of *n* is defined as follows:

$$n = 2^{(8\text{-macBeaconOrder})} \qquad 0 \leq \text{macBeaconOrder} \leq 8$$

$$n = 1 \qquad 9 \leq \text{macBeaconOrder} \leq 14$$

7.5.8 Frame security

The MAC sublayer is responsible for providing security services on specified incoming and outgoing frames when requested to do so by the higher layers. IEEE Std 802.15.4-2003 supports the following security services (see 5.4.6.1 for definitions):

— Access control
— Data encryption
— Frame integrity
— Sequential freshness

The protocol also provides the following security modes (see 5.4.6.2 for definitions):

— Unsecured mode
— ACL mode
— Secured mode

The information determining how to provide the security is found in the MAC PIB (see Table 72 in 7.4.2).

7.5.8.1 ACL entries

The MAC PIB security attributes contain a single default ACL entry and a number of additional ACL entries. The default ACL entry is known by every device in the PAN and is used in situations in which the device needs to communicate with a second device or with multiple devices that it may not know individually. Individual ACL entries are used in situations in which the device shares a key with a specific known device.

The default ACL entry consists of *macDefaultSecurity*, which indicates whether security is in use for devices not in the ACL; *macDefaultSecuritySuite*, which indicates the default security suite to use for frames to be sent or received from devices not in the ACL; and *macDefaultSecurityMaterial*, which indicates the keying material to use in secure communications involving frames to be sent or received from devices not in the ACL. If *macDefaultSecurity* is set to FALSE, *macDefaultSecuritySuite* and *macDefaultSecurityMaterial* shall not be used.

The additional ACL entries are contained in *macACLEntryDescriptorSet.* Each ACL entry corresponds to a trusted device and consists of its PAN identifier, its 64 bit extended address, its short address (or 0xffff if this address is not known), and its security suite and related keying material.

7.5.8.2 Functional description of unsecured mode

Unsecured mode is the default security mode for the MAC sublayer and provides no MAC sublayer security. A device operating in unsecured mode shall not utilize the ACL entries and shall not perform any security related operations on incoming frames. When a device receives a frame while in this mode, the MAC sublayer shall perform its filtering operations on the incoming frame, as described in 7.5.6.2, before checking the security enabled subfield. If the security enabled subfield in the frame is set to 1 and the device is not performing an active or passive scan (see 7.1.11.1), the MAC sublayer shall pass the frame to the next higher layer. This is achieved by issuing the MCPS-DATA.indication primitive (see 7.1.1.3) with the SecurityUse field set to TRUE and the ACLEntry field set to 0x08. If the device is performing an active or passive scan, the device shall accept beacon frames with the security enabled subfield set to 1 and set the SecurityUse, ACLEntry, and SecurityFailure fields of the PAN descriptor corresponding to that beacon to TRUE, 0x08, and TRUE, respectively (see Table 41). If the MAC sublayer receives a data frame with the security enabled subfield set to 0, it shall pass the frame to the next higher layer. This is achieved by issuing the MCPS-DATA.indication primitive (see 7.1.1.3) with the SecurityUse field set to FALSE and the ACLEntry field set to 0x08.

7.5.8.3 Functional description of ACL mode

ACL mode provides a mechanism for the MAC sublayer to indicate to the higher layer if a received frame purportedly originated from a device in the ACL. A device operating in ACL mode shall not make any modifications to the MAC frames or perform any cryptographic operations on the frames. As a result, ACL mode provides only a means for the device to filter received frames according to the source address in the frame, but not a means to securely determine the originator of the frame. When a device receives a frame while in this mode, the MAC sublayer shall perform its filtering operations on the incoming frame, as described in 7.5.6.2, before checking the security enabled subfield. If the security enabled subfield in the frame is set to 1 and the device is not performing an active or passive scan (see 7.1.11.1), the MAC sublayer shall pass the frame to the next higher layer. This is achieved by issuing the MCPS-DATA.indication primitive (see 7.1.1.3) with the SecurityUse field set to TRUE and the ACLEntry field set to the *macSecurityMode* parameter value from the ACL entry associated with the sender of the data frame, if present. If the sender of the data frame was not found in the ACL, the ACLEntry field shall be set to 0 x 08. If the device is performing an active or passive scan, the device shall accept beacon frames with the security enabled subfield set to 1 and set the SecurityUse and SecurityFailure fields of the PAN descriptor corresponding to that beacon to TRUE, and the ACLEntry field to the *macSecurityMode* parameter value from the ACL entry associated with the sender of the data frame, if present. If the sender of the data frame was not found in the ACL, the value 0x08 shall be used in the ACLEntry field (see Table 41). If the MAC sublayer receives a data frame with the security enabled subfield set to 0, it shall pass the frame to the next higher layer. This is achieved by issuing the MCPS-DATA.indication primitive (see 7.1.1.3) with the SecurityUse field set to FALSE and the ACLEntry field set to the *macSecurityMode* parameter value from the ACL entry associated with the sender of the data frame, if present. If the sender of the data was not found in the ACL, the ACLEntry field shall be set to 0x08. The device shall determine whether a device is in the ACL by searching *macACLEntryDescriptorSet* for an individual ACL entry with a value of ACLPANId that matches the received PAN identifier and a value of ACLExtendedAddress or ACLShortAddress that matches the received source address. If no source address is present in the frame, the ACLEntry field shall be set to 0 x 08.

7.5.8.4 Functional description of secured mode

Secured mode provides a mechanism for the MAC sublayer to both use the ACL functionality and provide cryptographic protection on incoming and outgoing frames. While in this mode, if the MAC sublayer receives an incoming frame or a request from the higher layer to transmit a frame, the MAC sublayer shall process the frame as specified in 7.5.8.4.1 and 7.5.8.4.2.

7.5.8.4.1 Processing outgoing frames in secured mode

While in secured mode, if the MLME receives a message from a higher layer to prepare a secure frame for transmission (i.e., the security enabled bit in the TxOptions is set to 1), the MLME shall scan the entries in the ACL for the correct entry to use. The MLME shall first search through *macACLEntryDescriptorSet* to find an entry in which the ACLPANId field and the ACLExtendedAddress or ACLShortAddress field match the destination address information of the frame to be created. If a match is found, the MLME shall select the security suite from the associated ACLSecuritySuite field and the security material from the associated ACLSecurityMaterial field for use on the outgoing frame.

If the MLME is unable to locate an ACLPANId and an ACLExtendedAddress or ACLShortAddress that matches the destination address information of the frame to be created, the MLME shall examine *macDefaultSecurity*. If *macDefaultSecurity* is equal to TRUE, the MLME shall select the security suite indicated in *macDefaultSecuritySuite* and the security material from *macDefaultSecurityMaterial* for use on the outgoing frame.

If the MLME is unable to locate an ACLPANId and an ACLExtendedAddress or ACLShortAddress that matches the destination address information of the frame to be created and *macDefaultSecurity* is equal to FALSE, the MLME shall inform the next higher layer. This notification is achieved by issuing the MLME-COMM-STATUS.indication primitive with a status of UNAVAILABLE_KEY.

After the MLME obtains the appropriate security suite and security material from the ACL, the MLME shall first set the security enabled subfield in the frame control field to 1 before performing the cryptographic operations on the frame.

If the security suite specifies the use of encryption, the encryption operation shall be applied only to the data in the payload field within the MAC payload, i.e., the beacon payload field (see 7.2.2.1.8), command payload field (see 7.2.2.4.3), or data payload field (see 7.2.2.2.2), depending on the frame type. The remaining fields shall be left unencrypted. If a frame does not contain a payload field, encryption shall not be used. The result of the encryption operation shall be inserted into the payload field of the frame in the place of the original data.

If the security suite specifies the use of an integrity code, the integrity code shall be applied to the MHR concatenated with the MAC payload (see 7.2 for information on the MAC frame format). The result of the integrity code computation shall be placed in the payload field within the MAC payload of the frame in addition to any other data in the payload field. If the payload field does not contain any data, then it shall contain only the integrity code. Integrity codes shall not be used for acknowledgment frames.

The ordering and exact manner of performing the encryption and integrity operations and the placement of the resulting encrypted data or integrity code within the beacon payload, command payload, or data payload field are defined by the selected security suite (see 7.6).

If any of the security operations fail, the MLME shall not transmit the requested frame, but shall inform the next higher layer. This notification is achieved by issuing the MLME-COMM-STATUS.indication primitive with a status of FAILED_SECURITY_CHECK. If the length of the resulting frame is longer than *aMaxMACFrameSize*, the MLME shall not transmit the requested frame, but shall inform the next higher layer. This notification is achieved by issuing the MLME-COMM-STATUS.indication primitive with a status of FRAME_TOO_LONG.

If the security operations have been successfully performed and the payload field within the MAC payload has been modified appropriately, the device shall compute the FCS over the modified frame.

7.5.8.4.2 Processing incoming frames in secured mode

Any incoming frame may be protected by security. If the MLME receives a frame while in secured mode, it shall perform its filtering operations on the incoming frame, as described in 7.5.6.2, before checking the security enabled subfield to determine whether security is used for that frame.

If the security enabled subfield of the frame control field is set to 0 and the incoming frame is an association request command frame, the MLME of the coordinator shall pass the frame information to the next higher layer. This is achieved by issuing the MLME-ASSOCIATE.indication primitive (see 7.1.3.2). If the security enabled subfield of the frame control field is set to 0 and the incoming frame is a beacon request command frame, the coordinator shall process the frame as described in 7.5.2.1.2. If the security enabled subfield of the frame control field is set to 0 and the device is performing an active or passive scan, the device shall accept beacon frames and set the SecurityUse and SecurityFailure fields of the PAN descriptor corresponding to each received beacon to FALSE and TRUE, respectively, and the ACLEntry field to the *macSecurityMode* parameter value from the ACL entry associated with the sender of the data frame, if present. If the sender of the data was not found in the ACL, the value 0x08 shall be used in the ACLEntry field (see Table 41). Otherwise, if the security enabled subfield of the frame control field is set to 0, the device shall pass the frame to the next higher layer. This is achieved by issuing the MCPS-DATA.indication primitive (see 7.1.1.3) with the SecurityUse field set to FALSE and the ACLEntry field set to the *macSecurityMode* parameter value from the ACL entry associated with the sender of the data frame, if present. If the sender of the data frame was not found in the ACL, the value 0 x 08 shall be used in the ACLEntry field.

If the security enabled subfield in the frame control field is set to 1, the MLME shall scan the entries in the MAC PIB security attributes for the correct entry to use. The MLME shall first search through *macACLEntryDescriptorSet* to find an entry in which the ACLPANId, ACLExtendedAddress, and ACLShortAddress fields match the source address information of the frame received. If a match is found, the MLME shall select the security suite from the associated ACLSecuritySuite field and the security material from the associated ACLSecurityMaterial field for use on the incoming frame.

If the MLME is unable to locate an ACLPANId and an ACLExtendedAddress or ACLShortAddress that matches the source address information of the received frame, the MLME shall examine *macDefault-Security*. If *macDefaultSecurity* is equal to TRUE, the MLME shall select the security suite from *mac-DefaultSecuritySuite* and the security material from *macDefaultSecurityMaterial* for use on the incoming frame.

If the MLME is unable to locate an ACLPANId and an ACLExtendedAddress or ACLShortAddress that matches the source address information of the received frame and *macDefaultSecurity* is equal to FALSE and the device is not performing an active or passive scan (see 7.1.11.1), the MLME shall pass the frame to the next higher layer. This is achieved by issuing the MCPS-DATA.indication primitive (see 7.1.1.3) with the SecurityUse field set to TRUE and the ACLEntry field set to 0x08. If the device is performing an active or passive scan, the device shall accept secure beacon frames for which the corresponding security material cannot be found and set the SecurityUse, ACLEntry, and SecurityFailure fields of the PAN descriptor corresponding to that beacon to TRUE, 0x08, and TRUE, respectively (see Table 41).

After the MLME obtains the appropriate security suite and security material values from the ACL, the MAC sublayer shall apply the security operations, defined by these values, to the frame.

If the security suite specifies the use of encryption, the decryption operation shall be applied only to the data in the payload field within the MAC payload, i.e., the beacon payload field (see 7.2.2.1.8), command payload field (see 7.2.2.4.3), or data payload field (see 7.2.2.2.2), depending on the frame type. If a frame does not contain any data in the payload field, decryption shall not be used. The result of the decryption operation shall be inserted into the payload field of the frame in the place of the original encrypted data.

If the security suite specifies the use of an integrity code, the integrity code shall be checked by first removing the integrity code and any other security suite specific data (e.g., the frame counter and key sequence counter) from the payload field within the MAC payload and then verifying the integrity code on the MHR concatenated with the MAC payload.

The ordering and exact manner of performing the decryption and integrity operations and the location of the security data within the payload field are defined by the security suite in use (see 7.6).

If at least one of the security operations fails and the device is not performing an active or passive scan (see 7.1.11.1), the MLME shall discard the frame and inform the next higher layer. This notification is achieved by issuing the MLME-COMM-STATUS.indication primitive with a status of FAILED_SECURITY_ CHECK. If the device is performing an active or passive scan, the device shall accept beacon frames that cause a security operation failure and set the SecurityUse and SecurityFailure fields of the PAN descriptor to TRUE and set the ACLEntry field to TRUE if the key used in the operations was found in *macACLEntryDescriptorSet* or FALSE if the key used in the operations was found in *macDefaultSecurityMaterial* (see Table 41).

If the security operations have been successfully performed and the payload field within the MAC payload has been modified appropriately, the device shall then continue to process the frame. In the indication of the frame to the higher layer, the MAC shall set the SecurityUse field to TRUE and set the ACLEntry field to TRUE if the key used in the operations was found in *macACLEntryDescriptorSet* or FALSE if the key used in the operations was found in *macDefaultSecurityMaterial*.

7.6 Security suite specifications

Security suites may be used when a device is operating in secured mode. A security suite consists of a set of operations to perform on MAC frames that provide security services. The security suite name indicates the symmetric cryptography algorithm, mode, and integrity code bit length. The bit length of the integrity code is less than or equal to the block size of the symmetric algorithm and determines the probability that a random guess of the integrity code would be correct. This bit length does not correspond to the strength of the underlying algorithm. For all security suites in this standard, the algorithm used shall be advanced encryption standard (AES) (see 7.6.1.7). Each device that implements security shall support the AES-CCM-64 security suite (see 7.6.3) and zero or more additional security suites. Each security suite is specified by a 1 octet value as shown in Table 75; an identifier of 0x00 indicates that secured mode is not to be used.

Table 75—Security suite list

Identifier	Security suite name	Security services				Subclause
		Access control	Data encryption	Frame integrity	Sequential freshness (optional)	
0 x 00	None					
0 x 01	AES-CTR	X	X		X	7.6.2
0 x 02	AES-CCM-128	X	X	X	X	7.6.3

Table 75—Security suite list *(continued)*

Identifier	Security suite name	Security services				Subclause
		Access control	Data encryption	Frame integrity	Sequential freshness (optional)	
0 x 03	AES-CCM-64	X	X	X	X	7.6.3
0 x 04	AES-CCM-32	X	X	X	X	7.6.3
0 x 05	AES-CBC-MAC-128	X		X		7.6.4
0 x 06	AES-CBC-MAC-64	X		X		7.6.4
0 x 07	AES-CBC-MAC-32	X		X		7.6.4

7.6.1 Security suite building blocks

The following methods are defined for use in the security suites specified in this standard:

- Bit ordering
- Concatenation
- Integer encoding and counter incrementing
- CTR encryption
- CBC-MAC authentication
- CCM combined encryption and authentication
- AES encryption
- PIB security material

7.6.1.1 Bit ordering

For security operations in this standard, a bit is defined to be an element of the set {0, 1}. An octet (also called a byte) is defined to be a bit string of length 8 arranged in the order in which it would be transmitted, where bit 7 is the first bit in the octet and bit 0 is the last bit in the octet. An octet string (also called a byte string) is an array of octets arranged in order with the most significant octet first. The terms *first* and *last* and *leftmost* and *rightmost* are used to distinguish the ends of octet strings (first and leftmost are equivalent; last and rightmost are equivalent). Within an octet, the terms *first* and *last*, *leftmost* and *rightmost*, and *high-order* and *low-order* are used for the order of the bits (first, leftmost, and high-order are equivalent; last, rightmost, and low-order are equivalent).

Note that the first bit in an octet string is indexed as bit 7 of octet 0 and represents the high-order bit of the first octet. The 16th bit in an octet string is indexed as bit 0 of octet 2 and represents the low-order bit of the second octet.

7.6.1.2 Concatenation

In this standard, concatenation of two octet strings a and b of length m and n, respectively, denoted $a||b$, consists of the octet string of length $m + n$ with the leftmost m octets equal to a and the rightmost n octets equal to b.

7.6.1.3 Integer encoding and counter incrementing

Unless otherwise stated, for security operations in 7.6 and in Annex Annex B of this standard, when an integer is represented as an octet string, the first octet (octet 0) corresponds to the most significant octet and the first bit (bit 7) within the first octet corresponds to the MSB. An octet string A of length n, is written as a bit string $A = a_{0,7}a_{0,6}a_{0,5} \dots a_{n-1,1}a_{n-1,0}$. An octet string or bit string is converted to an integer I by assigning each bit $a_{i,j}$ the value $a_{i,j}*2^{8i+j}$ and setting I to be the sum of all of the values.

As an example of integer encoding, consider the integer 11146, which corresponds to the octet string composed of the two octets 0x2b 8a. This integer can be represented as the bit string 0010 1011 1000 1010. In this case, the most significant octet (octet 0) is 0x2b and the least significant octet (octet 1) is 0 x 8a. The LSB (bit 0) of octet 0 has the value 1, as does the MSB (bit 7) of octet 1. The integer 1 can be represented as a 4 octet string by 0x00 00 00 01 and as a bit string by 0000 0000 0000 0000 0000 0000 0000 0001.

The counter incrementing operation in this standard takes as input an integer that is encoded as an octet string of length n. When the counter incrementing operation is invoked, the integer shall be incremented (increased by 1). If the incremented integer is less than 2^{8n}, the operation shall return the octet string encoding of the new integer. Otherwise the operation shall set the counter to $2^{8n} - 1$ and return an error (i.e., the counter incrementing operation would have caused the integer value to roll over).

7.6.1.4 CTR encryption

The counter mode (CTR) symmetric encryption algorithm used in this standard consists of the generation of a key stream using a block cipher in CTR, with a given key and nonce, and performing an exclusive OR (XOR) of the key stream with the plaintext and integrity code. A nonce is a time stamp, a counter, or a special marker intended to prevent unauthorized message replay. The decryption operation consists of the generation of the key stream and the XOR of the key stream with the ciphertext to obtain the plaintext.

All of the operations (listed in the paragraph above) shall be performed as specified in Annex Annex B, which defines CTR encryption in the general sense. A security suite implementing CTR encryption will specify any parameters left unspecified in Annex Annex B according to the requirements of that particular security suite.

7.6.1.5 CBC-MAC authentication

The cipher block chaining message authentication code (CBC-MAC) symmetric authentication algorithm used in this standard consists of the generation of an integrity code using a block cipher in CBC mode computed on a message that includes the length of the authenticated data at the beginning of the data themselves. The verification operation consists of the computation of this integrity code and comparison to the received integrity code.

All of the above operations shall be performed as specified in Annex Annex B, which defines CBC-MAC mode in the general sense. A security suite implementing CBC-MAC authentication will specify any parameters left unspecified in Annex Annex B according to the requirements of that particular security suite.

7.6.1.6 CCM combined encryption and authentication

The CTR encryption plus CBC-MAC (CCM) combined symmetric encryption and authentication mechanism used in this standard consists of the generation of an integrity code followed by the encryption of plaintext data and the integrity code. The output consists of the encrypted data and the encrypted integrity code.

The symmetric authentication operation used in this security suite consists of the generation of an integrity code using a block cipher in CBC mode computed on a nonce followed by padded authentication data followed by padded plaintext data, if present. The verification operation consists of the computation of this integrity code and comparison to the received integrity code.

The symmetric encryption operation used in this security suite consists of the generation of a key stream using a block cipher in CTR with a given key and nonce and performing an XOR of the key stream with the integrity code and plaintext, if present. The decryption operation consists of the generation of the key stream and the XOR of the key stream with the ciphertext to obtain the plaintext and integrity code.

All of the operations (listed in the paragraph above) shall be performed as specified in Annex Annex B, which defines CCM mode in the general sense. A security suite implementing CCM combined encryption and authentication will specify any parameters left unspecified in Annex Annex B according to the requirements of that particular security suite.

7.6.1.7 AES encryption

The AES encryption algorithm used in this standard shall be performed as specified in NIST FIPS Pub 197 (see 2.4). This encryption algorithm is parameterized by the use of 128 bit block size; the key length selected shall be 128 bits.

7.6.1.8 PIB security material

This subclause describes the formats of the security information stored in the MAC PIB. This information is dependent on the individual security suite selected and is referenced in this clause.

The symmetric key is the AES key for this ACL entry that shall be used to perform exactly one of (CTR encryption) or (CCM encryption and authentication) or (CBC-MAC authentication). An AES key shall not be used with different security suites.

The frame counter is the running counter that shall be included in the payload field in the MAC payload of the MAC frame. This counter is incremented each time a secure frame is transmitted, as specified in the ‘outgoing frame operations’ section in security suites that use frame counters. This counter will not roll over (see 7.6.1.3). This value helps to ensure that the CCM nonce is unique and allows the recipient to use the counter to ensure freshness.

The key sequence counter is a counter that is fixed by the higher layer and that shall be included in the payload field in the MAC payload of the MAC frame. The key sequence counter can be used, for instance, if the frame counter is exhausted. This value helps to ensures that the CCM nonce is unique and allows the recipient to use the counter to ensure freshness. If freshness is used, the higher layer should not decrease this counter as the freshness operation on the recipient side would fail.

The optional external frame counter and optional external key sequence counter are fields that may be stored in the ACL entry that represent the values of the last received frame counter and key sequence counter, respectively, in secure frames corresponding to this ACL entry. If the optional external frame counter and optional external key sequence counter fields are included in the ACL entry, the MAC will use them to verify the sequential freshness of received secure frames as described in the ‘incoming frame operations’ section in security suites that use frame counters.

7.6.2 AES-CTR security suite

The AES-CTR security suite is used when a device is operating in secured mode. The cryptographic operations in this security suite consist of performing AES-CTR encryption (or decryption) on the payload field within the MAC payload using shared data, the frame counter, and the key sequence counter.

The AES-CTR security suite provides the following security services:

- Access control, defined in 5.4.6.1.1
- Data encryption, defined in 5.4.6.1.2
- Sequential freshness, defined in 5.4.6.1.4 (optional)

7.6.2.1 Data formats

The data formats used in the AES-CTR security suite are defined in 7.6.2.1.1 through 7.6.2.1.3.

7.6.2.1.1 MAC PIB formats

For the AES-CTR security suite, the security material stored in *macDefaultSecurityMaterial* or the ACL-SecurityMaterial field in the ACL consists of a symmetric key, a frame counter, and a key sequence counter as well as an optional frame counter and sequence counter that may be used for incoming frames. If these optional fields are included in the security material, the optional operations specified in 7.6.2.3.2 shall be performed. When these optional operations are performed, the security suite provides the sequential freshness security service on received frames. Figure 66 specifies the order and length of the AES-CTR security material components.

Octets: 16	**4**	**1**	**(4)**	**(1)**
Symmetric key	Frame counter	Key sequence counter	Optional external frame counter	Optional external key sequence counter

Figure 66—AES-CTR security material

7.6.2.1.2 Protected payload field formats

In the AES-CTR security suite, the payload field in the MAC payload of a protected frame consists of the frame counter, the key sequence counter, and the encrypted payload. Figure 67 specifies the order and length of the subfields of the payload field of an AES-CTR secured frame. The length of the encrypted payload field is equal to the length of the payload field before it was encrypted.

Octets: 4	**1**	**variable**
Frame counter	Key sequence counter	Encrypted payload

Figure 67—AES-CTR payload field

7.6.2.1.3 CTR input blocks

In the AES-CTR security suite, the input blocks to the CTR encryption function for generating the key stream consist of a flag octet, the address of the device sending the frame, the frame and key sequence counter values, and the block counter value. Figure 68 specifies the order and length of the subfields of the

CTR input blocks, the use of which is described in 7.6.2.3.1 and 7.6.2.3.2. These input blocks correspond to the counters T_1, T_2, ... , T_n as specified in Annex Annex B.

Octets: 1	8	4	1	2
Flags	Source address	Frame counter	Key sequence counter	Block counter

Figure 68—AES-CTR input block

The flags octet used in AES-CTR mode, which is used as padding for the input block, is formatted as specified in Figure 69. The bit values in this flags octet are chosen simply to distinguish the AES-CTR flags from the AES-CCM flags (see Annex Annex B for information on the AES-CCM flags).

Bits: 7	6-2	1	0
1	0	1	0

Figure 69—AES-CTR flags field

7.6.2.2 Security parameters

The CTR encryption operation in AES-CTR security suite, as defined in 7.6.1.4, shall be parameterized by the following:

- The underlying block cipher shall be the AES encryption algorithm as specified in 7.6.1.7.
- The counter input blocks shall be formatted as specified in 7.6.2.1.3 where each input block is the same except that the block counter is set to 0 for the first block and is incremented as specified in 7.6.1.3 for each successive input block. In other words, the i^{th} input block T_i shall have the block counter value set to $i - 1$ (see Annex Annex B for further explanation).

7.6.2.3 Security operations

The operations performed on outgoing and incoming frames are specified in 7.6.2.3.1 and 7.6.2.3.2.

7.6.2.3.1 Outgoing frame operations

When the AES-CTR security suite is invoked to protect an outgoing frame, the MAC sublayer shall perform the following operations:

a) Obtain its own 64 bit extended address, *aExtendedAddress*, along with the frame counter and the sequence counter from the MAC PIB, and construct the counter input blocks as specified in 7.6.2.2.
b) Encrypt the payload field in the MAC payload of the frame using CTR encryption, as specified in 7.6.1.4, with the parameters specified in 7.6.2.2, and using the counter input blocks from step a).
c) Combine the frame counter, sequence counter, and output from step b), as specified in 7.6.2.1.2, to obtain the new payload field.
d) Increment the frame counter as specified in 7.6.1.3 and, if the incrementing succeeds, insert the new counter value into the MAC PIB. If the incrementing operation fails because the counter value rolled over, the device shall abort the operation and issue the MLME-COMM-STATUS.indication primitive to the higher layer with a status of FAILED_SECURITY_CHECK.

7.6.2.3.2 Incoming frame operations

When the AES-CTR security suite is invoked to protect an incoming frame, the MAC sublayer shall perform the following operations in order:

a) If the optional external frame and external key sequence counters are included in the corresponding *macDefaultSecurityMaterial* or ACLSecurityMaterial field, ensure sequential freshness by verifying that the received key sequence counter is greater than or equal to the external key sequence counter from that device. If the key sequence counter is greater than or equal to the external key sequence counter, verify that the received frame counter is greater than or equal to the external frame counter from that device. If either of these checks fails, the device shall reject the frame and issue the MLME-COMM-STATUS.indication primitive to the higher layer with a status of FAILED_SECURITY_CHECK.
b) Obtain the 64 bit extended address of the source either from the frame or from the ACL, extract the frame counter and sequence counter from the payload field in the MAC payload and construct the counter input blocks as specified in 7.6.2.2. If the nonce cannot be constructed because the data are unavailable, the device shall issue the MLME-COMM-STATUS.indication primitive to the higher layers with a status of FAILED_SECURITY_CHECK.
c) Decrypt the encrypted payload field using CTR decryption, as specified in 7.6.1.4, with the parameters specified in 7.6.2.2 and using the counter input blocks from step b).
d) Replace the existing payload field in the MAC payload with the decrypted data from step c). If the optional operation in step a) was performed and the checks succeeded, the last known sequence counter and last known frame counter shall be set to the received values.

7.6.3 AES-CCM security suite

The AES-CCM security suite is used when a device is operating in secured mode. The cryptographic operations in this security suite consist of performing AES-CCM authentication (or verification) on the MHR concatenated with the MAC payload and encryption (or decryption) on the payload field in the MHR using shared data, the frame counter, and the key sequence counter. The AES-CCM security suite shall be implemented using 32 bit, 64 bit, or 128 bit integrity codes.

The AES-CCM security suite provides the following security services:

— Access control, defined in 5.4.6.1.1
— Data encryption, defined in 5.4.6.1.2
— Frame integrity, defined in 5.4.6.1.3
— Sequential freshness, defined in 5.4.6.1.4 (optional)

7.6.3.1 Data formats

The data formats used in the AES-CCM security suite are defined in 7.6.3.1.1 through 7.6.3.1.3.

7.6.3.1.1 MAC PIB formats

For the AES-CCM security suite, the security material stored in *macDefaultSecurityMaterial* or the ACLSecurityMaterial field in the ACL consists of a symmetric key, a frame counter, and a key sequence counter as well as an optional frame counter and sequence counter that may be used for incoming frames. If these optional fields are included in the security material, the optional operations specified in 7.6.3.3.2 shall be performed. When these optional operations are performed, the security suite provides the sequential freshness security service on received frames. Figure 70 specifies the order and length of the AES-CCM security material components.

Octets: 16	4	1	(4)	(1)
Symmetric key	Frame counter	Key sequence counter	Optional external frame counter	Optional external key sequence counter

Figure 70—AES-CCM security material

7.6.3.1.2 Protected payload field formats

In the AES-CCM security suite, the payload field in the MAC payload of a protected frame consists of the frame counter, the key sequence counter, the encrypted payload, and the encrypted integrity code. Figure 71 specifies the order and length of the subfields of the payload field in the MAC payload of an AES-CCM secured frame. The length of the encrypted payload field is equal to the length of the payload field before it was encrypted. The length of the encrypted integrity code subfield corresponds to the length of integrity code required, i.e., 4 octets for 32 bit, 8 octets for 64 bit, or 16 octets for 128 bit.

Octets: 4	1	variable	4, 8, or 16
Frame counter	Key sequence counter	Encrypted payload	Encrypted integrity code

Figure 71—AES-CCM MAC payload field

7.6.3.1.3 CCM nonce

In the AES-CCM security suite, the nonce input used for the CCM authentication and encryption function consists of data explicitly included in the frame and data that both devices can independently obtain. Figure 72 specifies the order and length of the subfields of the CCM nonce.

Octets: 8	4	1
Source address	Frame counter	Key sequence counter

Figure 72—AES-CCM nonce

7.6.3.2 Security parameters

In the AES-CCM security suite, the CCM operations as defined in 7.6.1.6 shall be parameterized by the following:

- — The underlying block cipher shall be the AES encryption algorithm as specified in 7.6.1.7.
- — The length in octets of the length field L shall be 2 octets.
- — The length of the authentication field M shall be 4 octets, 8 octets, or 16 octets as required.
- — The nonce shall be formatted as specified in 7.6.3.1.3.

7.6.3.3 Security operations

The operations performed on outgoing and incoming frames are specified in 7.6.3.3.1 and 7.6.3.3.2.

7.6.3.3.1 Outgoing frame operations

When the AES-CCM security suite is invoked to protect an outgoing frame, the MAC sublayer shall perform the following operations:

a) Obtain its own 64 bit extended address, *aExtendedAddress*, along with the frame counter and the sequence counter from the MAC PIB, and construct the nonce as specified in 7.6.3.1.3.
b) Encrypt and authenticate the MHR and MAC payload in the frame using CCM authentication and encryption, as specified in 7.6.1.6, with the parameters specified in 7.6.3.2. Use the MHR and nonpayload fields in the MAC payload as the authentication data, *a*; the payload field in the MAC payload as the message, *m*; and the nonce computed in step a).
c) Combine the frame counter, sequence counter, and output from step b) (including the encrypted payload and encrypted integrity code), as specified in 7.6.3.1.2, to obtain the new payload field.
d) Increment the frame counter as specified in 7.6.1.3 and, if the incrementing succeeds, insert the new counter value into the MAC PIB. If the incrementing operation fails because the counter value rolled over, the device shall abort the operation and issue the MLME-COMM-STATUS.indication primitive to the higher layer with a status of FAILED_SECURITY_CHECK.

7.6.3.3.2 Incoming frame operations

When the AES-CCM security suite is invoked to protect an incoming frame, the MAC sublayer shall perform the following operations in order:

a) If the optional external frame and external key sequence counters are included in the corresponding *macDefaultSecurityMaterial* or ACLSecurityMaterial field, ensure sequential freshness by verifying that the received key sequence counter is greater than or equal to the external key sequence counter from that device. If the key sequence counter is equal to the external key sequence counter, verify that the received frame counter is greater than the external frame counter from that device. If either of these checks fails, the device shall reject the frame and issue the MLME-COMM-STATUS.indication primitive to the higher layer with a status of FAILED_SECURITY_CHECK.
b) Obtain the 64 bit extended address of the source either from the frame or from the ACL, remove the frame counter and sequence counter from the payload field in the MAC payload, and construct the nonce as specified in 7.6.3.1.3. If the nonce cannot be constructed because the data are unavailable, the device shall reject the frame and issue the MLME-COMM-STATUS.indication primitive to the higher layers with a status of FAILED_SECURITY_CHECK.
c) Decrypt the encrypted payload field and verify the integrity code using CCM decryption and authentication, as specified in 7.6.1.6, with the parameters specified in 7.6.3.2. Use the MHR and nonpayload fields in the MAC payload as the authentication data, *a*; the encrypted payload field as the message, *m*; and the nonce computed in step b). If the integrity code fails, the device shall discard the frame and issue the MLME-COMM-STATUS.indication primitive to the higher layers with a status of FAILED_SECURITY_CHECK.
d) Replace the existing payload field in the MAC payload with the decrypted data from step c). If the optional operation in step a) was performed and the checks succeeded, the last known sequence counter and last known frame counter shall be set to the received values.

7.6.4 AES-CBC-MAC security suite

The AES-CBC-MAC security suite is used when a device is operating in secured mode. The cryptographic operations in this security suite consist of performing AES-CBC-MAC authentication on the MHR and

MAC payload. The AES-CBC-MAC security suite shall be implemented using 32 bit, 64 bit or 128 bit integrity codes.

The AES-CBC-MAC security suite provides the following security services:

- Access control, defined in 5.4.6.1.1
- Frame integrity, defined in 5.4.6.1.3

7.6.4.1 Data formats

The data formats used in the AES-CBC-MAC security suite are defined in 7.6.4.1.1 through 7.6.4.1.3.

7.6.4.1.1 MAC PIB formats

For the AES-CBC-MAC security suite, the security material stored in *macDefaultSecurityMaterial* or the ACLSecurityMaterial field in the ACL consists of a symmetric key. No state information is required between frames. Figure 73 specifies the order and length of the AES-CBC-MAC security material components.

Octets: 16
Symmetric key

Figure 73—AES-CBC-MAC security material

7.6.4.1.2 Protected payload field formats

In the AES-CBC-MAC security suite, the payload field in the MAC payload of a protected frame consists of the existing payload followed by the integrity code. Figure 74 specifies the order and length of the subfields of the payload field in the MAC payload of an AES-CBC-MAC secured frame. The length of the integrity code subfield corresponds to the length of integrity code required, i.e., 4 octets for 32 bit, 8 octets for 64 bit or 16 octets for 128 bit.

Octets: variable	**4, 8 or 16**
Payload	Integrity code

Figure 74—AES-CBC-MAC payload field

7.6.4.1.3 CBC-MAC input blocks

In the AES-CBC-MAC security suite, the input to the CBC-MAC authentication function for generating the integrity code consists of the length of the data to authenticate (not including the length field itself) followed by the MHR followed by the MAC payload. The input is broken up into 16 octet blocks starting from the left and proceeding to the right until the last block, which may be smaller than 16 octets depending on the length of the total input. Figure 75 specifies the order and length of the subfields of the CBC-MAC input.

Octets: 1	**variable = n**	**variable = m**
Length = *n* + *m*	MHR	MAC payload

Figure 75—AES-CBC-MAC input

7.6.4.2 Security parameters

In the AES-CBC-MAC security suite, the CBC-MAC operations as defined in 7.6.1.5 shall be parameterized by the following:

— The underlying block cipher shall be the AES encryption algorithm as specified in 7.6.1.7.
— The input to the CBC-MAC function shall be formatted as specified in 7.6.4.1.3.
— The length of the integrity code M shall be 32 bits, 64 bits, or 128 bits, as required.

7.6.4.3 Security operations

The operations performed on outgoing and incoming frames are specified in 7.6.4.3.1 and 7.6.4.3.2.

7.6.4.3.1 Outgoing frame operations

When the AES-CBC-MAC security suite is invoked to protect an outgoing frame, the MAC sublayer shall perform the following operations:

a) Determine the length in octets of the MHR concatenated with the MAC payload (before the security operations are performed) and encode that length as a 1 octet integer, as specified in 7.6.1.3.
b) Compute the integrity code on the MHR and MAC payload in the frame using CBC-MAC authentication, as specified in 7.6.1.5, with the parameters specified in 7.6.4.2.
c) Combine the existing payload field in the MAC payload and output from step b), as specified in 7.6.4.1.2, to obtain the new payload field.

7.6.4.3.2 Incoming frame operations

When the AES-CBC-MAC security suite is invoked to protect an incoming frame, the MAC sublayer shall perform the following operations:

a) Determine the length in octets of the MHR concatenated with the MAC payload, before the security operations were applied, and encode that length as a 1 octet integer as specified in 7.6.1.3.
b) Parse the payload field of the MAC payload into the payload and integrity code subfields, as specified in 7.6.4.1.2, and verify the integrity code using CBC-MAC authentication, as specified in 7.6.1.5, with the parameters specified in 7.6.4.2. Use the length determined in step a), the MHR from the received frame, and the MAC payload (without the integrity code) as the input data. If the integrity code fails, the device shall discard the frame and issue the MLME-COMM-STATUS.indication primitive to the higher layers with a status of FAILED_SECURITY_CHECK.
c) Remove the integrity code from the payload field in the MAC payload.

7.7 Message sequence charts illustrating MAC-PHY interaction

This subclause illustrates the main tasks specified in IEEE Std 802.15.4-2003. Each task is described by use of a message sequence chart to illustrate the chronological order, rather than the exact timing, of the primitives required for each task.

The primitives necessary for a PAN coordinator to start a new PAN are shown in Figure 76. The first action the next higher layer takes after resetting the MAC sublayer is to initiate a scan to search for other PANs in the area. The scan in the message sequence chart refers to Figure 77, which shows the steps for performing an ED scan. An active scan can also be used to search the area for neighboring PANs, but is not shown here.

Once a new PAN is established, the PAN coordinator is ready to accept requests from other devices to join the PAN. Figure 78 shows the primitives issued by the device requesting association, while Figure 79 illustrates the steps taken by the PAN coordinator allowing association. In the process of joining a PAN, the device requesting association will perform either an active or a passive scan to determine which PANs in the area are allowing association; Figure 80 details the primitives necessary to complete a passive scan. The association procedure also applies to devices joining the PAN through a coordinator that is not the central PAN coordinator (see Clause 3 for definitions of both coordinator and PAN coordinator).

The primitives necessary for transmitting and receiving a single data packet are shown next. The actions taken by the originator of the packet are shown in Figure 81, while the actions taken by the recipient are shown in Figure 82.

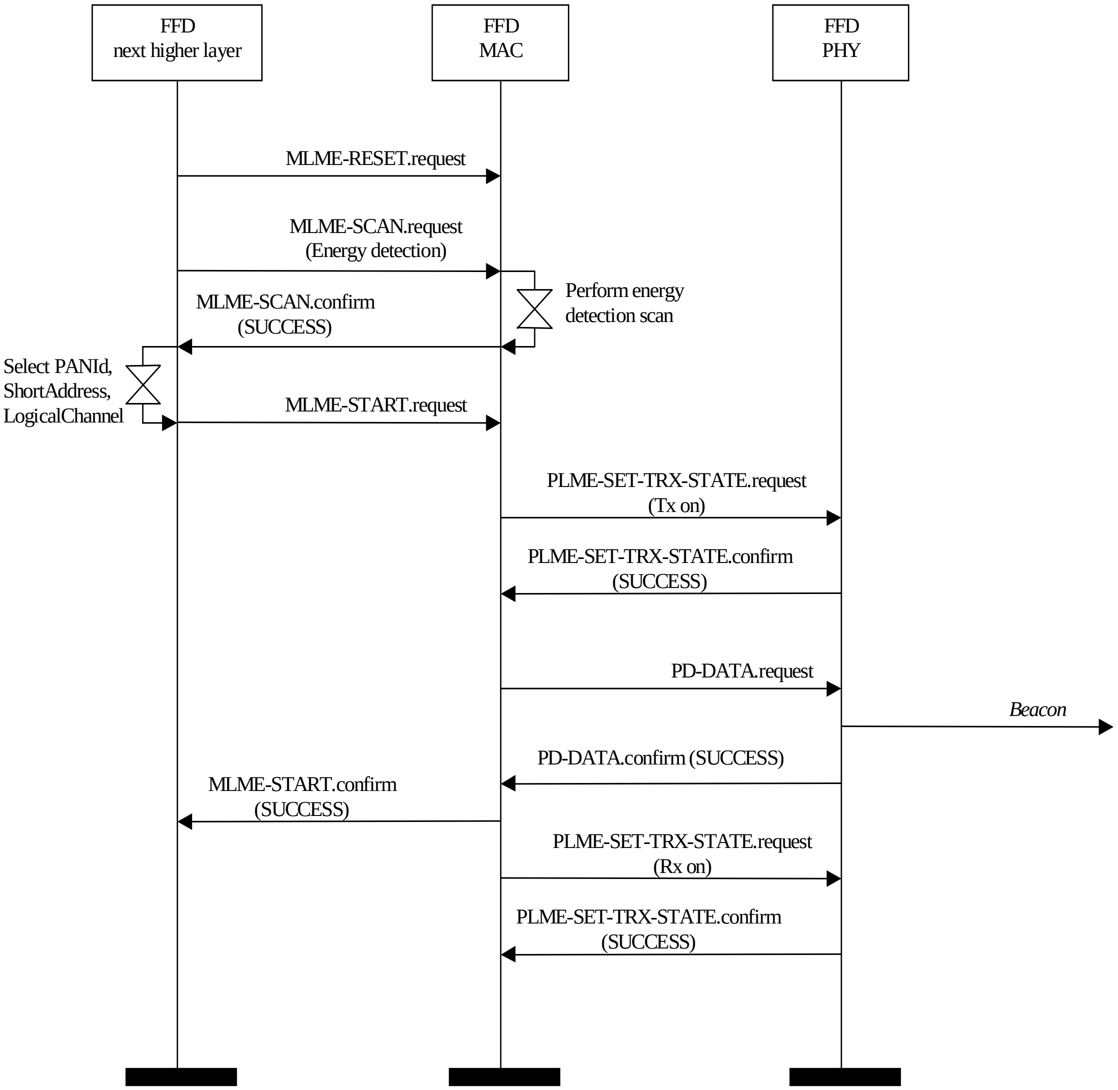


Figure 76—PAN start message sequence chart—PAN coordinator

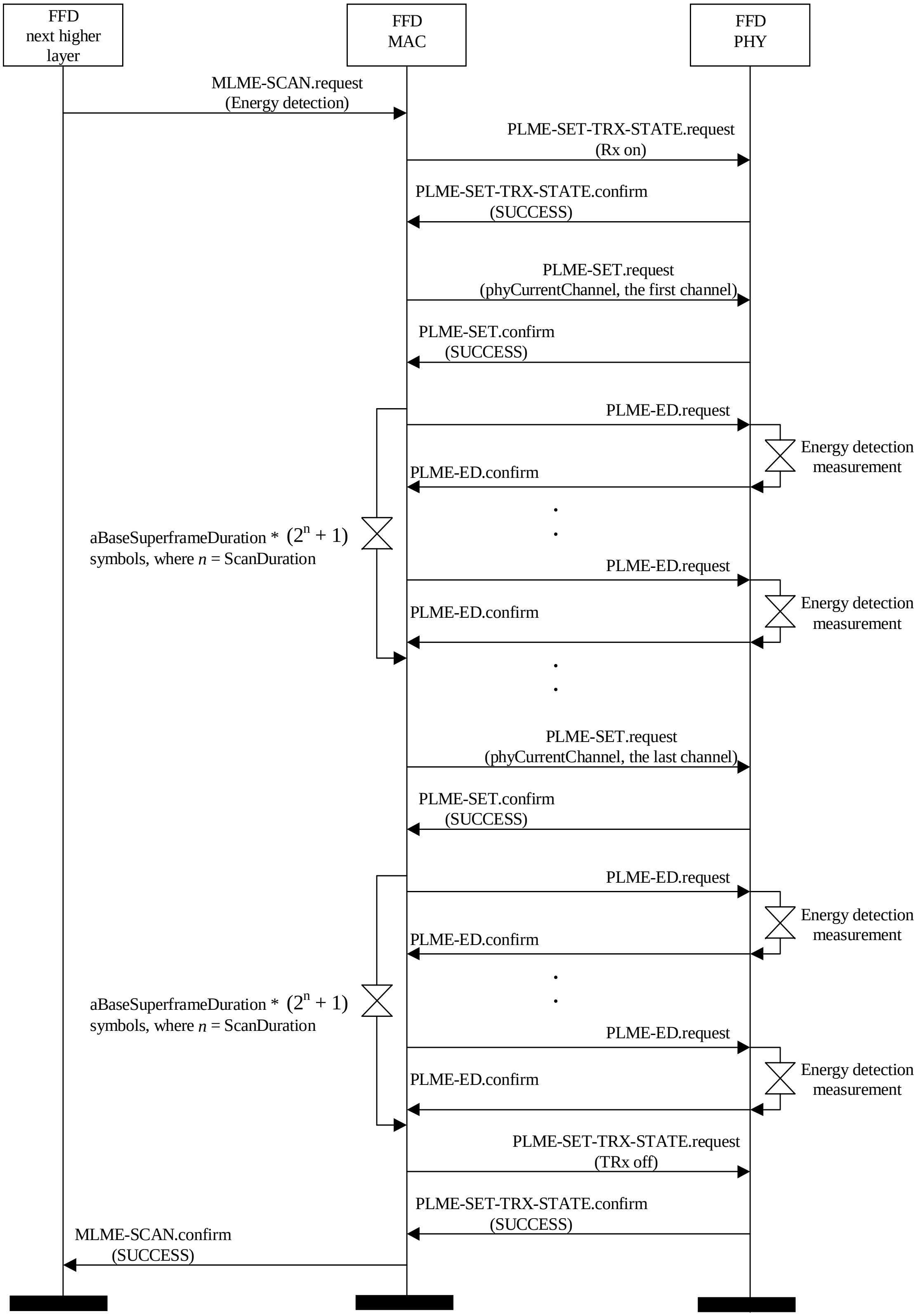


Figure 77—ED scan message sequence chart

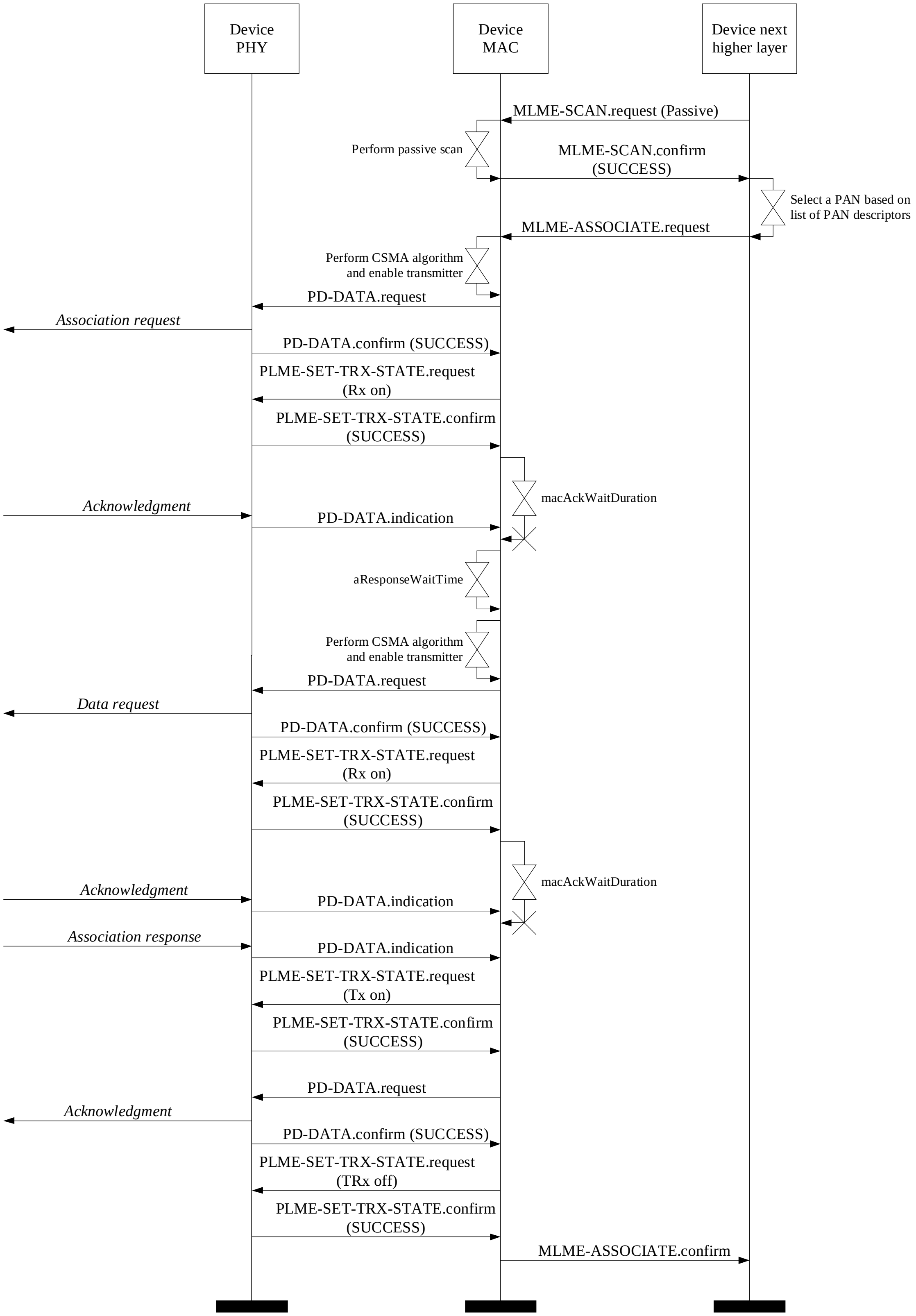

Device PHY
Device MAC
Device next higher layer
MLME-SCAN.request (Passive)
Perform passive scan
MLME-SCAN.confirm (SUCCESS)
Select a PAN based on list of PAN descriptors
MLME-ASSOCIATE.request
Perform CSMA algorithm and enable transmitter
PD-DATA.request
Association request
PD-DATA.confirm (SUCCESS)
PLME-SET-TRX-STATE.request (Rx on)
PLME-SET-TRX-STATE.confirm (SUCCESS)
macAckWaitDuration
Acknowledgment
PD-DATA.indication
aResponseWaitTime
Perform CSMA algorithm and enable transmitter
PD-DATA.request
Data request
PD-DATA.confirm (SUCCESS)
PLME-SET-TRX-STATE.request (Rx on)
PLME-SET-TRX-STATE.confirm (SUCCESS)
macAckWaitDuration
Acknowledgment
PD-DATA.indication
Association response
PD-DATA.indication
PLME-SET-TRX-STATE.request (Tx on)
PLME-SET-TRX-STATE.confirm (SUCCESS)
PD-DATA.request
Acknowledgment
PD-DATA.confirm (SUCCESS)
PLME-SET-TRX-STATE.request (TRx off)
PLME-SET-TRX-STATE.confirm (SUCCESS)
MLME-ASSOCIATE.confirm

Figure 78—Association message sequence chart—device

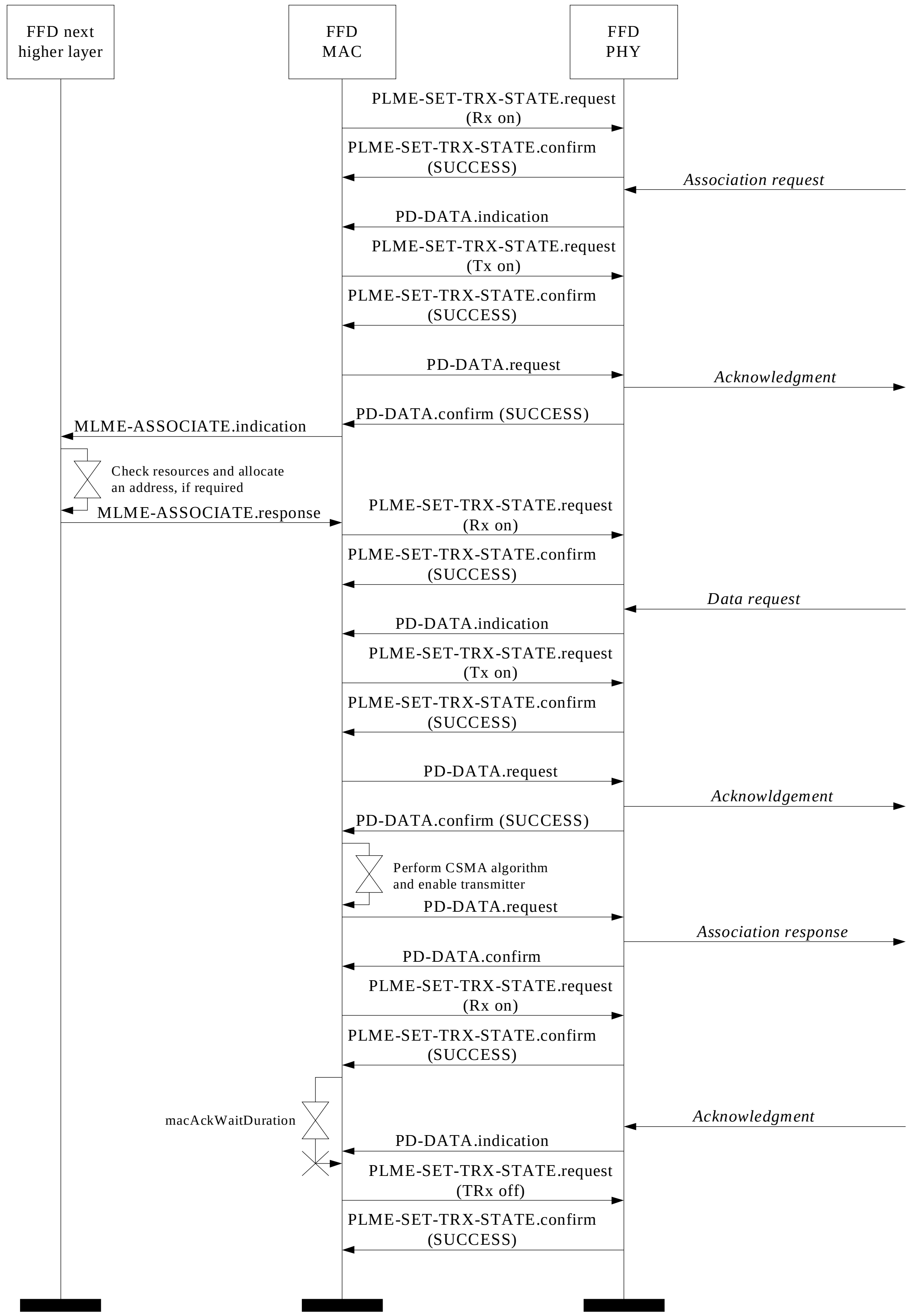

FFD next higher layer
FFD MAC
FFD PHY
PLME-SET-TRX-STATE.request (Rx on)
PLME-SET-TRX-STATE.confirm (SUCCESS)
Association request
PD-DATA.indication
PLME-SET-TRX-STATE.request (Tx on)
PLME-SET-TRX-STATE.confirm (SUCCESS)
PD-DATA.request
Acknowledgment
PD-DATA.confirm (SUCCESS)
MLME-ASSOCIATE.indication
Check resources and allocate an address, if required
MLME-ASSOCIATE.response
PLME-SET-TRX-STATE.request (Rx on)
PLME-SET-TRX-STATE.confirm (SUCCESS)
Data request
PD-DATA.indication
PLME-SET-TRX-STATE.request (Tx on)
PLME-SET-TRX-STATE.confirm (SUCCESS)
PD-DATA.request
Acknowldgement
PD-DATA.confirm (SUCCESS)
Perform CSMA algorithm and enable transmitter
PD-DATA.request
Association response
PD-DATA.confirm
PLME-SET-TRX-STATE.request (Rx on)
PLME-SET-TRX-STATE.confirm (SUCCESS)
macAckWaitDuration
Acknowledgment
PD-DATA.indication
PLME-SET-TRX-STATE.request (TRx off)
PLME-SET-TRX-STATE.confirm (SUCCESS)

Figure 79—Association message sequence chart—coordinator

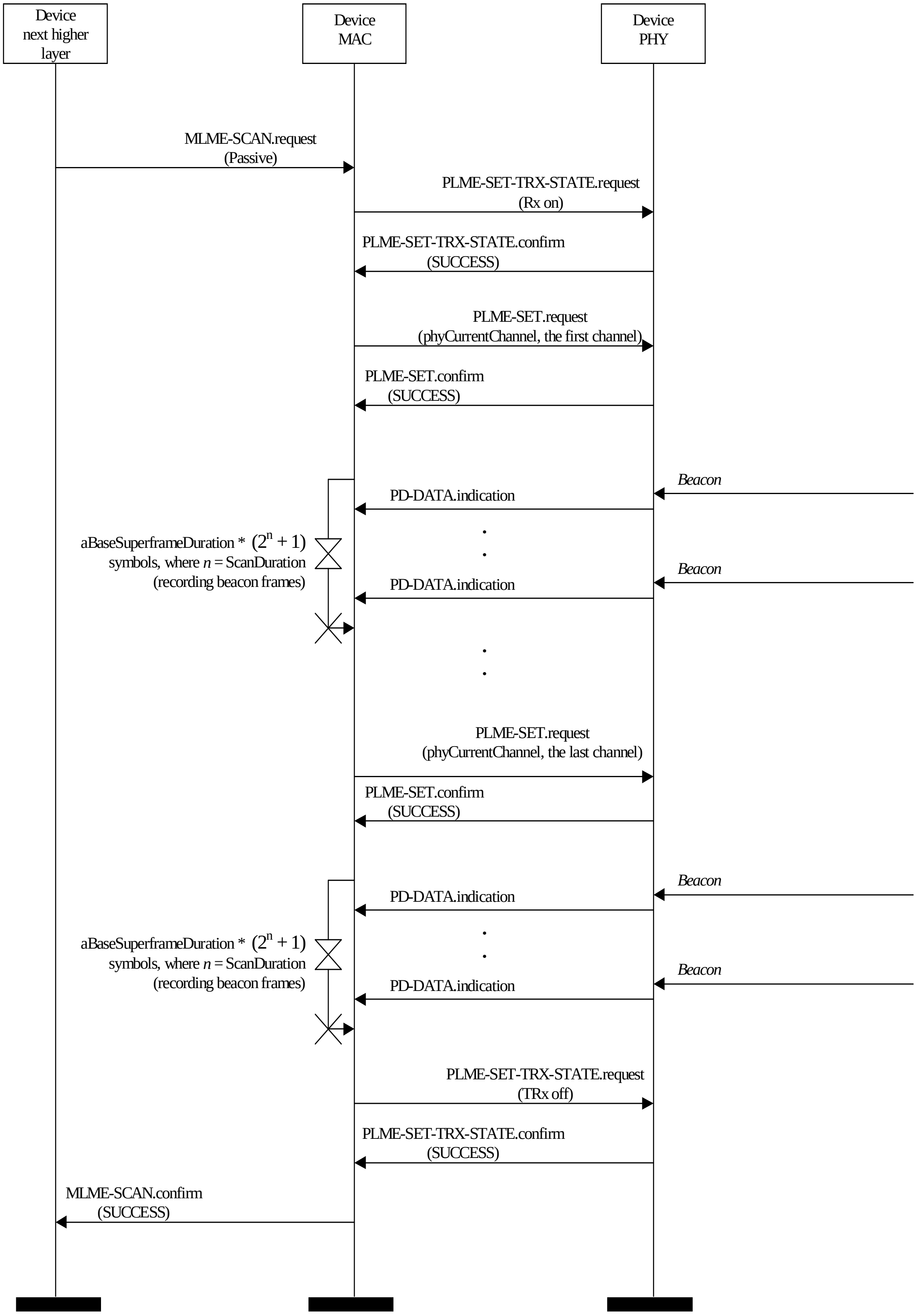

Device
next higher
layer
Device
MAC
Device
PHY
MLME-SCAN.request
(Passive)
PLME-SET-TRX-STATE.request
(Rx on)
PLME-SET-TRX-STATE.confirm
(SUCCESS)
PLME-SET.request
(phyCurrentChannel, the first channel)
PLME-SET.confirm
(SUCCESS)
Beacon
PD-DATA.indication
aBaseSuperframeDuration * $(2^n + 1)$
symbols, where n = ScanDuration
(recording beacon frames)
Beacon
PD-DATA.indication
PLME-SET.request
(phyCurrentChannel, the last channel)
PLME-SET.confirm
(SUCCESS)
Beacon
PD-DATA.indication
aBaseSuperframeDuration * $(2^n + 1)$
symbols, where n = ScanDuration
(recording beacon frames)
Beacon
PD-DATA.indication
PLME-SET-TRX-STATE.request
(TRx off)
PLME-SET-TRX-STATE.confirm
(SUCCESS)
MLME-SCAN.confirm
(SUCCESS)

Figure 80—Passive scan message sequence chart

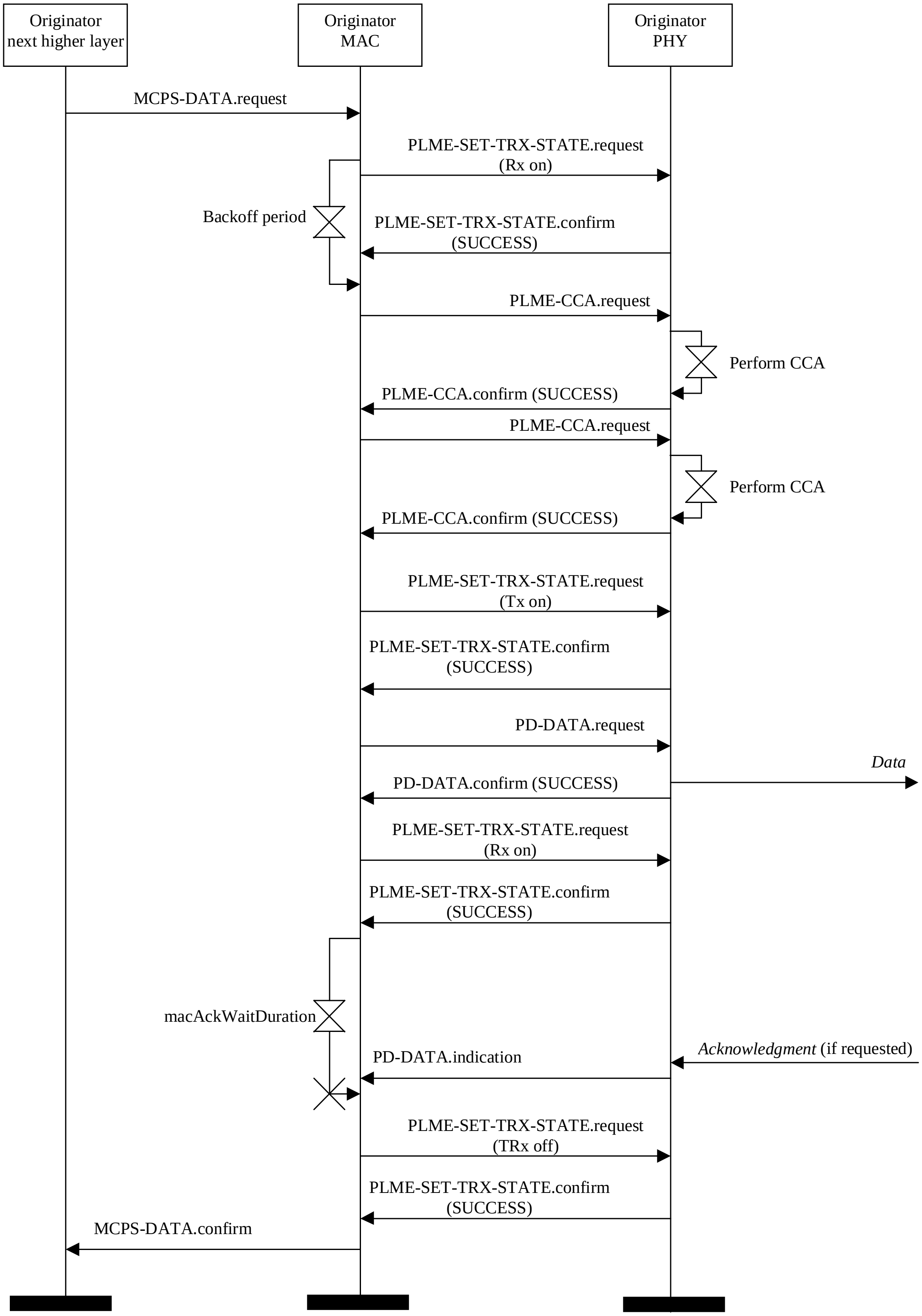

Originator next higher layer
Originator MAC
Originator PHY
MCPS-DATA.request
PLME-SET-TRX-STATE.request (Rx on)
Backoff period
PLME-SET-TRX-STATE.confirm (SUCCESS)
PLME-CCA.request
Perform CCA
PLME-CCA.confirm (SUCCESS)
PLME-CCA.request
Perform CCA
PLME-CCA.confirm (SUCCESS)
PLME-SET-TRX-STATE.request (Tx on)
PLME-SET-TRX-STATE.confirm (SUCCESS)
PD-DATA.request
Data
PD-DATA.confirm (SUCCESS)
PLME-SET-TRX-STATE.request (Rx on)
PLME-SET-TRX-STATE.confirm (SUCCESS)
macAckWaitDuration
Acknowledgment (if requested)
PD-DATA.indication
PLME-SET-TRX-STATE.request (TRx off)
PLME-SET-TRX-STATE.confirm (SUCCESS)
MCPS-DATA.confirm

Figure 81—Data transmission message sequence chart—originator

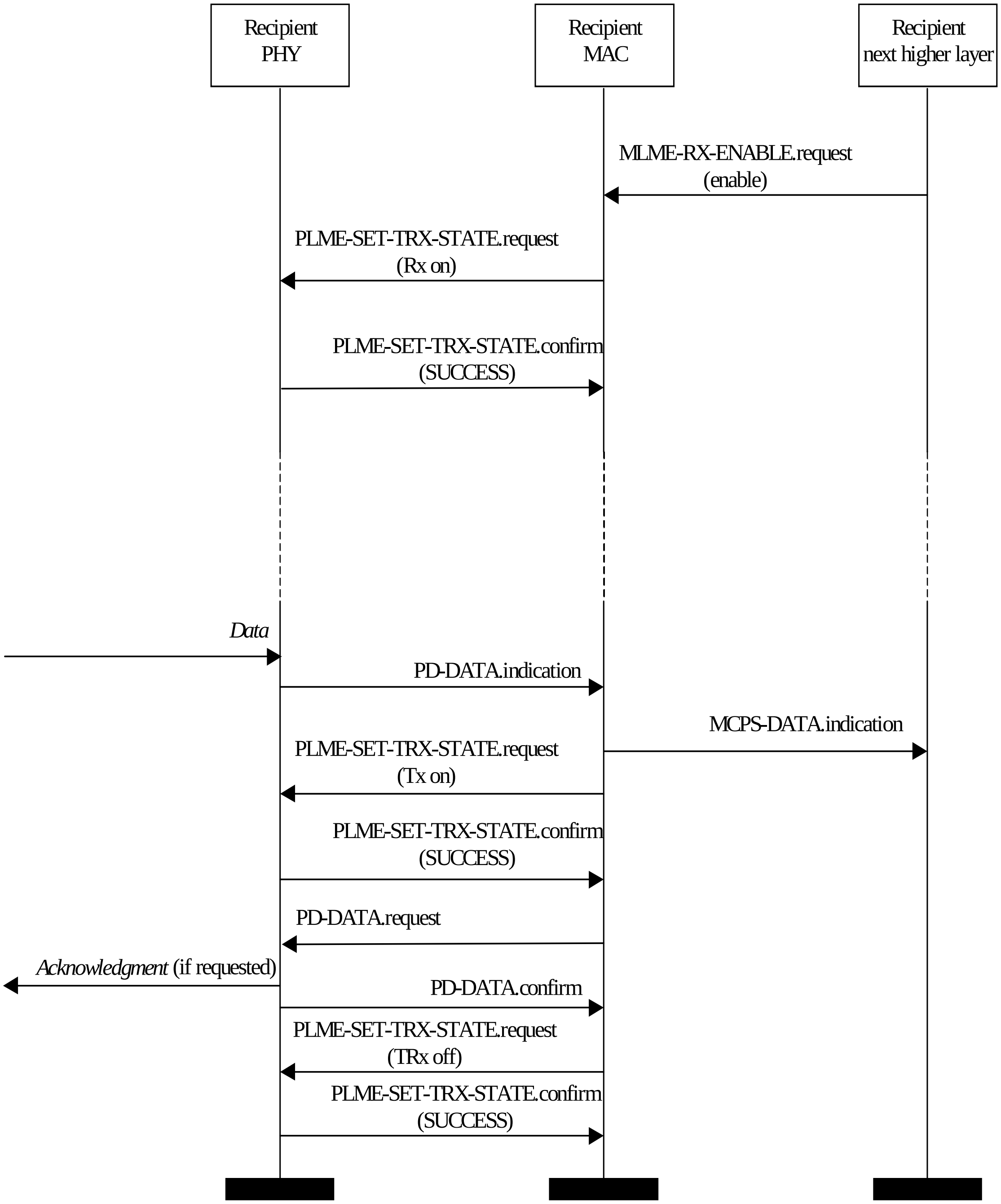

Recipient PHY
Recipient MAC
Recipient next higher layer
MLME-RX-ENABLE.request (enable)
PLME-SET-TRX-STATE.request (Rx on)
PLME-SET-TRX-STATE.confirm (SUCCESS)
Data
PD-DATA.indication
MCPS-DATA.indication
PLME-SET-TRX-STATE.request (Tx on)
PLME-SET-TRX-STATE.confirm (SUCCESS)
PD-DATA.request
Acknowledgment (if requested)
PD-DATA.confirm
PLME-SET-TRX-STATE.request (TRx off)
PLME-SET-TRX-STATE.confirm (SUCCESS)

Figure 82—Data transmission message sequence chart—recipient

Annex A

(normative)

SSCS

The SSCS exists conceptually above the MCPS.

A.1 802.2 Convergence sublayer

The 802.2 convergence sublayer exists above the IEEE 802.15.4 MCPS. This sublayer provides an interface between an instance of an IEEE 802.2 LLC sublayer and the IEEE 802.15.4 MCPS.

A.1.1 MA-UNITDATA.request

The MA-UNITDATA.request primitive requests the transfer of a LLC protocol data unit (LPDU) (i.e., MSDU) from a local IEEE 802.2 Type 1 LLC sublayer entity to a single peer IEEE 802.2 Type 1 LLC sublayer entity or multiple peer IEEE 802.2 Type 1 LLC sublayer entities in the case of a group address.

A.1.1.1 Semantics of the service primitive

The semantics of the MA-UNITDATA.request primitive is as follows:

```
MA-UNITDATA.request          (
                             SrcAddr,
                             DstAddr,
                             RoutingInformation,
                             data,
                             priority,
                             ServiceClass
                             )
```

Table A.1 specifies the parameters for the MA-UNITDATA.request primitive.

Table A.1—MA-UNITDATA.request parameters

Name	Type	Valid range	Description
SrcAddr	IEEE address	Any valid IEEE address	The individual IEEE address of the entity from which the MSDU is being transferred.
DstAddr	IEEE address	Any valid IEEE address	The individual IEEE address of the entity to which the MSDU is being transferred.
RoutingInformation	—	null	This parameter is not used by the MAC sublayer and shall be specified as a null value.
data	Set of octets	—	The set of octets forming the MSDU to be transmitted by the MAC sublayer entity.

Table A.1—MA-UNITDATA.request parameters *(continued)*

Name	Type	Valid range	Description
priority	—	null	This parameter is not used by the MAC sublayer and shall be specified as a null value.
ServiceClass	—	null	This parameter is not used by the MAC sublayer and shall be specified as a null value.

A.1.1.2 When generated

The MA-UNITDATA.request primitive is generated by a local IEEE 802.2 Type 1 LLC sublayer entity when an LPDU (MSDU) is to be transferred to a peer IEEE 802.2 Type 1 LLC sublayer entity or entities.

A.1.1.3 Effect on receipt

On receipt of the MA-UNITDATA.request primitive, the MAC sublayer entity shall begin the transmission of the supplied MSDU.

The MAC sublayer first builds an MPDU to transmit from the supplied arguments. The MPDU shall be transmitted using the CSMA-CA algorithm in the contention period of the frame and without requesting a handshake.

If the CSMA-CA algorithm indicates a busy channel, the MAC sublayer shall issue the MA-UNITDATA-STATUS.indication primitive with a status of CHANNEL_ACCESS_FAILURE. If the MPDU was successfully transmitted, the MAC sublayer shall issue the MA-UNITDATA-STATUS.indication primitive with a status of SUCCESS.

A.1.2 MA-UNITDATA.indication

The MA-UNITDATA.indication primitive indicates the transfer of an LPDU (i.e., MSDU) from the MAC sublayer to the local IEEE 802.2 Type 1 LLC sublayer entity.

A.1.2.1 Semantics of the service primitive

The semantics of the MA-UNITDATA.indication primitive is as follows:

```
MA-UNITDATA.indication         (
                               SrcAddr,
                               DstAddr,
                               RoutingInformation,
                               data,
                               ReceptionStatus,
                               priority,
                               ServiceClass
                               )
```

Table A.2 specifies the parameters for the MA-UNITDATA.indication primitive.

Table A.2—MA-UNITDATA.indication parameters

Name	Type	Valid range	Description
SrcAddr	IEEE address	Any valid IEEE address	The individual IEEE address of the entity from which the MSDU has been received
DstAddr	IEEE address	Any valid IEEE address	The individual IEEE address of the entity to which the MSDU is being transferred.
RoutingInformation	—	null	This parameter is not used by the MAC sublayer and shall be specified as a null value.
data	Set of octets	—	The set of octets forming the MSDU received by the MAC sublayer entity.
ReceptionStatus	—	null	This parameter is not used by the MAC sublayer and shall be specified as a null value.
priority	—	null	This parameter is not used by the MAC sublayer and shall be specified as a null value.
ServiceClass	—	null	This parameter is not used by the MAC sublayer and shall be specified as a null value.

A.1.2.2 When generated

On receipt of a data packet at the local MAC sublayer entity, the FCS field is checked. If it is valid, the MAC sublayer shall issue the MA-UNITDATA.indication primitive to the IEEE 802.2 Type 1 LLC sublayer entity, indicating the arrival of a MSDU. If the FCS is not valid, the packet shall be discarded, and the IEEE 802.2 Type 1 LLC sublayer entity shall not be informed.

A.1.2.3 Effect on receipt

The effect on receipt of the MA-UNITDATA.indication primitive is not specified in this standard.

A.1.3 MA-UNITDATA-STATUS.indication

The MA-UNITDATA-STATUS.indication primitive reports the results of a request to transfer a LPDU (MSDU) from a local IEEE 802.2 Type 1 LLC sublayer entity to a single peer IEEE 802.2 Type 1 LLC sublayer entity or to multiple peer IEEE 802.2 Type 1 LLC sublayer entities.

A.1.3.1 Semantics of the service primitive

The semantics of the MA-UNITDATA-STATUS.indication primitive is as follows:

```
MA-UNITDATA-STATUS.indication    (
                                 SrcAddr,
                                 DstAddr,
                                 status,
                                 ProvPriority,
                                 ProvServiceClass
                                 )
```

Table A.3 specifies the parameters for the MA-UNITDATA-STATUS.indication primitive.

Table A.3—MA-UNITDATA-STATUS.indication parameters

Name	Type	Valid Range	Description
SrcAddr	IEEE address	Any valid IEEE address	The individual IEEE address of the entity from which the MSDU has been transferred.
DstAddr	IEEE address	Any valid IEEE address	The individual IEEE address of the entity to which the MSDU has been transferred.
status	Enumeration	SUCCESS, TRANSMISSION_PENDING, NO_BEACON, or CHANNEL_ACCESS_FAILURE	The status of the last MSDU transmission.
ProvPriority	—	null	This parameter is not used by the MAC sublayer and shall be specified as a null value.
ProvServiceClass	—	null	This parameter is not used by the MAC sublayer and shall be specified as a null value.

A.1.3.2 When generated

The MA-UNITDATA-STATUS.indication primitive is generated by the MAC sublayer entity in response to an MA-UNITDATA.request primitive issued by the IEEE 802.2 Type 1 LLC sublayer.

A.1.3.3 Effect on receipt

The receipt of the MA-UNITDATA-STATUS.indication primitive by the IEEE 802.2 Type 1 LLC sublayer entity signals the completion of the current data transmission.

Annex B

(normative)

Security implementation

B.1 Generic CCM mode

CCM is a generic authenticate-and-encrypt block cipher mode. CCM is currently defined for use only with block ciphers with a 128 bit block size, such as AES. The CCM ideas can easily be extended to other block sizes, but the ideas will require further definitions.

For the generic CCM mode, there are two parameter choices to be made. The first choice is M, the size of the authentication field. The choice of the value for M involves a trade-off between message expansion and the probability that an attacker can undetectably modify a message. Valid values are 4, 6, 8, 10, 12, 14, and 16 octets. The second choice is L, the size of the length field. This value requires a trade-off between the maximum message size and the size of the nonce. Different applications require different trade-offs, so L is a parameter. Valid values are 2 to 8 octets (the value $L = 1$ is reserved). Table B.1 shows the CCM mode parameters.

Table B.1—Parameters of CCM mode

Name	Description	Field size	Encoding of field
M	Number of octets in authentication field	3 bits	$(M -)/2$
L	Number of octets in length field	3 bits	$L - 1$

B.1.1 Inputs

To send a message, the sender must provide the following information:

— An encryption key K suitable for the block cipher.

— A nonce N of $15 - L$ octets. Within the scope of any encryption key K, the nonce value shall be unique. In other words, the set of nonce values used with any given key shall not contain any duplicate values. Using the same nonce for two different messages encrypted with the same key destroys the security properties of this mode.

— The message m, consisting of a string of $l(m)$ octets where $0 \leq l(m) < 2^{8L}$. The length restriction ensures that $l(m)$ can be encoded in a field of L octets.

— Additional authenticated data a, consisting of a string of $l(a)$ octets where $0 \leq l(a) < 2^{64}$. These additional data are authenticated, but not encrypted, and are not included in the output of this mode. They can be used to authenticate plaintext headers or contextual information that affects the interpretation of the message. Users who do not wish to authenticate additional data can provide a string of length zero.

This information is summarized in Table B.2.

Table B.2—Inputs for CCM

Name	Description	Field size	Encoding of field
K	Block cipher key	Depends on block cipher	String of octets
N	Nonce	15 – *L* octets	Not specified
m	Message to be encrypted and sent	*l(m)* octets	String of octets
a	Additional authenticated data	*l(a)* octets	String of octets

B.1.2 Authentication

The first step is to compute the authentication field *T*. This is done using CBC-MAC. First define a sequence of blocks B_0, B_1, ... B_n and then apply CBC-MAC to these blocks.

The first block B_0 is formatted as indicated in Table B.3.

Table B.3—First authentication block B_0

Octet no.	0	1 ... 15 – *L*	16 – *L* ... 15
Contents	Flags	Nonce *N*	*l*(*m*)

The value *l(m)* is encoded in most-significant-octet-first order.

The flags field is formatted as indicated in Table B.4.

Table B.4—Authentication flags octet

Bit no.	7	6	5	4	3	2	1	0
Contents	Reserved	Adata	M			L		

The reserved bit is reserved for future expansions and should always be set to 0. The Adata bit is set to 0 if *l*(*a*) = 0 and set to 1 if *l*(*a*) > 0. The *M* field is assigned the value of 4*(bit 5) + 2*(bit 4) + (bit 3) and encodes the value of *M* as (*M* – 2)/2. As *M* can take on the even values from 4 to 16, the 3 bit field can take on the values from 1 to 7. The *L* field is assigned the value of 4*(bit 2) + 2*(bit 1) + (bit 0) and encodes the size of the length field used to store *l(m)*. The parameter *L* can take on the values from 2 to 8 (the value *L* = 1 is reserved). This value is encoded in the 3 bit field using the values from 1 to 7 by choosing the field value as *L* – 1 (the zero value is reserved).

If *l*(*a*) > 0 (as indicated by the Adata field), then one or more blocks of authentication data are added. These blocks contain *l*(*a*) and *a* encoded in a reversible manner. First construct a string that encodes *l*(*a*).

If $0 < l(a) < 2^{16}\text{-}2^8$ then the length field is encoded as 2 octets, which contain the value *l*(*a*) in most-significant-octet-first order.

If $2^{16} - 2^8 \le l(a) < 2^{32}$, then the length field is encoded as 6 octets consisting of the octets 0 x ff, 0 x fe, and 4 octets encoding $l(a)$ in most-significant-octet-first order.

If $2^{32} \le l(a) < 2^{64}$, then the length field is encoded as 10 octets consisting of the octets 0 x ff, 0 x ff, and 8 octets encoding $l(a)$ in most-significant-octet-first order.

This information is summarized in Table B.5. Note that all fields are interpreted in most-significant-octet-first order.

Table B.5—Length encoding for additional authentication data

First two octets	Followed by	Comment
0 x 0000		Reserved
0 x 0001 ... 0 x FEFF		For $0 < l(a) < 2^{16} - 2^8$
0 x FF00 ... 0 x FFFD		Reserved
0 x FFFE	4 octets *l(a)*	For $2^{16} - 2^8 \le l(a) < 2^{32}$
0 x FFFF	8 octets *l(a)*	For $2^{32} \le l(a) < 2^{64}$

The blocks encoding *a* are formed by concatenating the string that encodes $l(a)$ with *a* itself and splitting the result into 16 octet blocks, padding the last block with zeroes if necessary. These blocks are appended to the first block B_0.

After the (optional) additional authentication blocks have been added, add the message blocks. The message blocks are formed by splitting the message *m* into 16 octet blocks, padding the last block with zeroes if necessary. If the message *m* consists of the empty string, then no blocks are added in this step.

The result is a sequence of blocks $B_0, B_1, ..., B_n$. The CBC-MAC is now computed by

$$X_1 := \mathrm{E}(K, B_0)$$

$$X_{i+1} := \mathrm{E}(K, X_i \oplus B_i) \quad \text{for } i = 1, ... , n$$

$$T := \text{first-}M\text{-octets}(X_{n+1})$$

where

E() is the block cipher encryption function,
T is the MAC value.

Note that the last block B_n is XORed with X_n and encrypted with the block cipher to give *T.*

B.1.3 Encryption

To encrypt the message data, use CTR mode. First define the key stream blocks by

$$S_i := \mathrm{E}(K, A_i) \quad \text{for } i = 0, 1, 2, ...$$

The values A_i are formatted as shown in Table B.6.

Table B.6—Encryption blocks A_i

Octet no.	0	1 ... 15 – *L*	16 – *L* ... 15
Contents	Flags	Nonce *N*	Counter *i*

where *i* is encoded in most-significant-octet-first order.

The flags field is formatted as shown in Table B.7.

Table B.7—Encryption flags octet

Bit no.	7	6	5	4	3	2	1	0
Contents	Reserved	Reserved	0			L		

The reserved bits are reserved for future expansions and shall be set to 0. Bit 6 corresponds to the Adata bit in the B_0 block, but as this bit is not used here, it is reserved. Bit 3, bit 4, and bit 5 are set to 0. This ensures that all the *A* blocks are distinct from B_0, which has the nonzero encoding of *M* in this position. Bit 0, bit 1, and bit 2 contain *L*, using the same encoding as in B_0.

The message is encrypted by XORing the octets of message *m* with the first $l(m)$ octets of the concatenation of S_1, S_2, S_3, ... Note that S_0 is not used to encrypt the message.

The authentication value *U* is computed by encrypting *T* with the key stream block S_0 and truncating it to the desired length.

$$U := T \oplus \text{first-}M\text{-octets}(S_0\ )$$

B.1.4 Output

The final result *c* consists of the encrypted message *m*, followed by the encrypted authentication value *U*.

B.1.5 Decryption

To decrypt a message, the following information is required:

- The encryption key *K*.
- The nonce *N*.
- The additional authenticated data *a*.
- The encrypted and authenticated message *c*.

Decryption starts by recomputing the key stream to recover the message *m* and the MAC value *T*. The message and additional authentication data is then used to recompute the CBC-MAC value and check *T*.

If the *T* value is not correct, the receiver shall not reveal any information except for the fact that *T* is incorrect. In particular, the receiver shall not reveal the decrypted message, the value *T*, or any other information.

B.1.6 Restrictions

All implementations shall limit the total amount of data that are encrypted with a single key. The sender shall ensure that the total number of block cipher encryption operations in the CBC-MAC and encryption together shall not exceed 2^{61}. (This allows close to 2^{64} octets to be encrypted and authenticated using CCM, which should be more than enough for most applications.) Receivers that do not expect to decrypt the same message twice may also implement this limit.

The receiver shall verify the CBC-MAC before releasing any information such as the plaintext. If the CBC-MAC verification fails, the receiver shall destroy all information, except for the fact that the CBC-MAC verification failed

B.1.7 List of symbols

Table B.8 provides a list of the symbols used for the specification of CCM.

Table B.8—List of symbols

Name	Description	Size	Comment
a	Additional authenticated data	$l(a)$ octets	Use empty string if not desired.
A_i	Counter block to generate key stream	16 octets	Contains block counter, nonce, and flags.
B_i	Input block for CBC-MAC	16 octets	Encode *N*, *L*, *M*, *m*, and *a* uniquely.
c	Ciphertext	$l(m) + M$ octets	Includes the encrypted MAC.
K	Block cipher key	N/A	At least 128 bits, preferably 256 bits.
L	Number of octets in length field	3 bits	Values 1 ... 8, encoded in 3 bits as $L – 1$.
m	Message to be encrypted and sent	$l(m)$ octets	Subject to $0 \le l(m) < 2^{8L}$
M	Number of octets in authentication field	3 bits	Values 4, 6, 8, ..., 16. Encoded value is $(M – 2)/2$
N	Nonce	$15 – L$ octets	Nonce should never be repeated for same key.
S_i	Block of the encryption key stream	16 octets	Use S_0, S_1, S_2, ... to encrypt *m* and *T*.
T	Unencrypted authentication tag	*M* octets	
U	Encrypted authentication tag	*M* octets	Appended to the message after encryption
X_i	Intermediate value of CBC-MAC	16 octets	

B.2 CTR Encryption

The CTR is a confidentiality mode that features the application of the forward cipher to a set of input blocks, called counters, to produce a sequence of output blocks that are XORed with the plaintext to produce the ciphertext, and vice versa. The sequence of counters must have the property that each block in the sequence is different from every other block. This condition is not restricted to a single message: across all of the messages that are encrypted under the given key, all of the counters must be distinct. In this standard, the counters for a given message are denoted T_1, T_2, … , T_n. Given a sequence of counters, T_1, T_2, … , T_n, the CTR mode is defined as follows:

CTR encryption:$O_j = CIPH_K(T_j)$for j = 1, 2 … n;
$C_j = P_j \oplus O_j$for j = 1, 2 … n – 1;
$C^*_n = P^*_n \oplus MSB_u(O_n)$.

CTR decryption:$O_j = CIPH_K(T_j)$for j = 1, 2 … n;
$P_j = C_j \oplus O_j$for j = 1, 2 … n – 1;
$P^*_n = C^*_n \oplus MSB_u(O_n)$.

In CTR encryption, the forward cipher function is invoked on each counter block, and the resulting output blocks are XORed with the corresponding plaintext blocks to produce the ciphertext blocks. For the last block, which may be a partial block of u bits, the most significant u bits of the last output block are used for the XOR operation. The remaining $b – u$ bits of the last output block are discarded, where b is the length in bits of the block cipher.

In CTR decryption, the forward cipher function is invoked on each counter block, and the resulting output blocks are XORed with the corresponding ciphertext blocks to recover the plaintext blocks. For the last block, which may be a partial block of u bits, the most significant u bits of the last output block are used for the XOR operation. The remaining $b – u$ bits of the last output block are discarded.

In both CTR encryption and CTR decryption, the forward cipher functions can be performed in parallel. Similarly, the plaintext block that corresponds to any particular ciphertext block can be recovered independently from the other plaintext blocks if the corresponding counter block can be determined. Moreover, the forward cipher functions can be applied to the counters prior to the availability of the plaintext or ciphertext data.

The CTR mode is illustrated in Figure B.1.

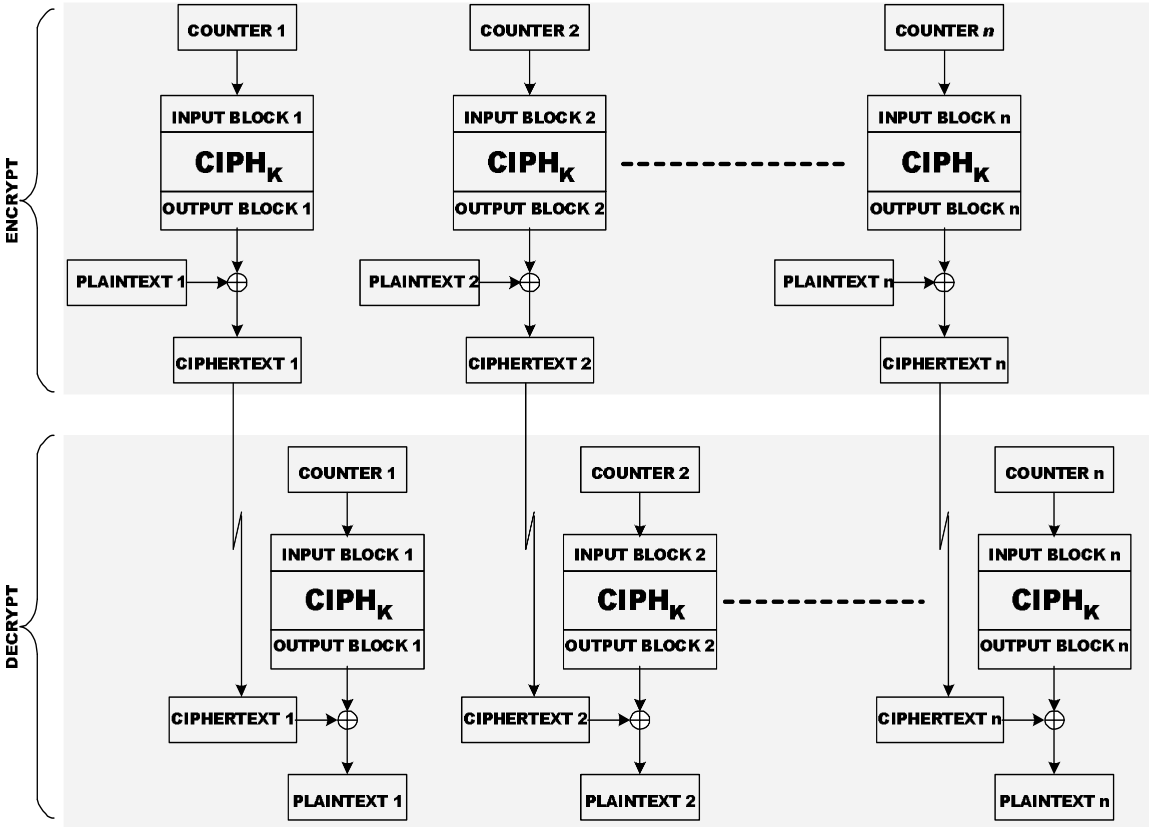


Figure B.1—CTR mode

B.3 CBC-MAC

The CBC-MAC algorithm makes use of an underlying block cipher to provide data integrity on input data. The block cipher transforms (or encrypts) input vectors of the block size to output vectors of the block size using a cryptographic key. Let *D* be any input vector and assume a key has been selected. The vector of length equal to the block size, *O*, which is the output of the block cipher when applied to *D*, using the enciphering operation, is represented as follows:

$$O = \mathrm{e}(D)$$

The data (e.g., record, file, message, program) to be authenticated is grouped into contiguous blocks, D_1, D_2,.... D_n, each with length equal to the block size. If the number of data bits is not a multiple of the block size, then the final input block will be a partial block of data, left justified, with zeroes appended to form a full block. The calculation of the MIC is given by the following equations where ⊕ represents the XOR of two vectors.

$$O_1 = \mathrm{e}(D_1)$$

$$O_2 = \mathrm{e}(D_2 \oplus O_1)$$

$$O_3 = \mathrm{e}(D_3 \oplus O_2)$$

$$. . .$$

$$O_n = \mathrm{e}(D_n \oplus O_{n-1})$$

The MIC is selected from O_n. Devices that implement CBC-MAC shall be capable of selecting the leftmost M bits of O_n as the MIC, where $32 < M < 128$ and M is a multiple of 8. A block diagram of the MIC generation is given in Figure B.2.

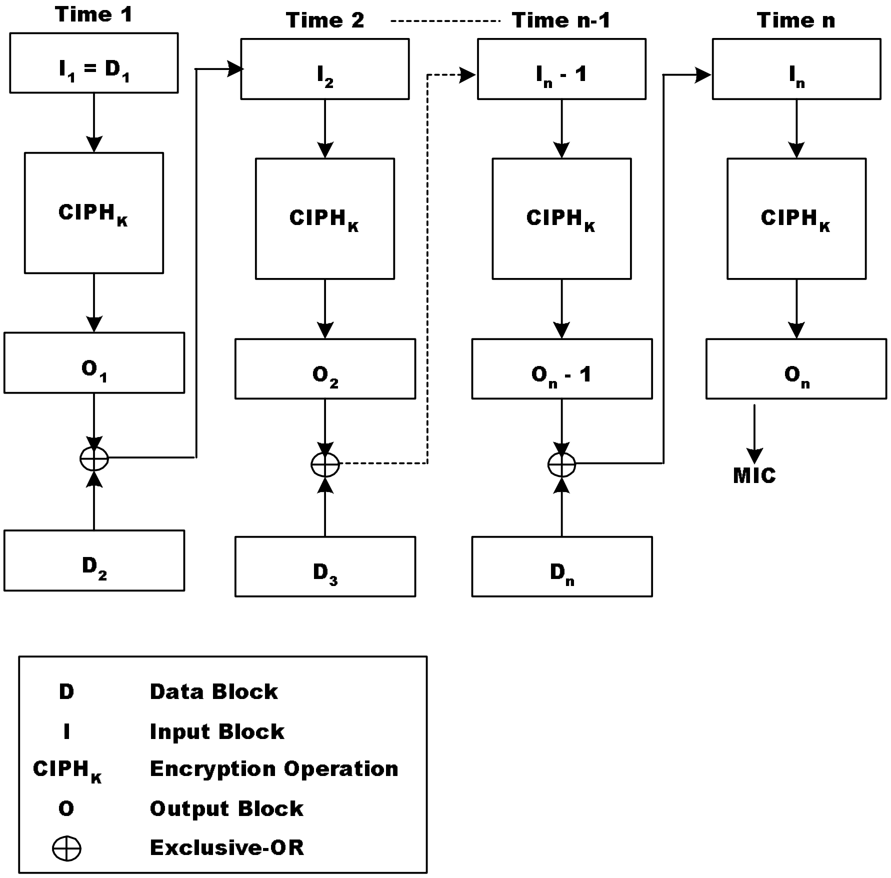


Figure B.2— CBC-MAC

Annex C

(normative)

Protocol implementation conformance statement (PICS) proforma[7]

C.1 Introduction

To evaluate conformance of a particular implementation, it is necessary to have a statement of which capabilities and options have been implemented for a given standard. Such a statement is called a protocol implementation conformance statement (PICS).

C.1.1 Scope

This annex provides the PICS proforma for IEEE Std 802.15.4-2003 in compliance with the relevant requirements, and in accordance with the relevant guidance, given in ISO/IEC 9646-7:1995.

C.1.2 Purpose

The supplier of a protocol implementation claiming to conform to IEEE Std 802.15.4-2003 shall complete the following PICS proforma and accompany it with the information necessary to identify fully both the supplier and the implementation.

The PICS of a protocol implementation is a statement of which capabilities and options of the protocol have been implemented. The statement is in the form of answers to a set of questions in the PICS proforma. The questions in a proforma consist of a systematic list of protocol capabilities and options as well as their implementation requirements. The implementation requirement indicates whether implementation of a capability is mandatory, optional, or conditional depending on options selected. When a protocol implementor answers questions in a PICS proforma, the implementor indicates whether an item is implemented and provides explanations if an item is not implemented.

C.2 Abbreviations and special symbols

Notations for requirement status:

M	Mandatory
O	Optional
O.n	Optional, but support of at least one of the group of options labeled O.n is required.
N/A	Not applicable
X	Prohibited
“item”:	Conditional, status dependent upon the support marked for the “item”

For example, FD1: O.1 indicates that the status is optional but at least one of the features described in FD1 and FD2 is required to be implemented, if this implementation is to follow the standard to which this PICS proforma is part.

[7] Copyright release for PICS proformas: Users of this standard may freely reproduce the PICS proforma in this annex so that it can be used for its intended purpose and may further publish the completed PICS.

C.3 Instructions for completing the PICS proforma

If it is claimed to conform to this standard, the actual PICS proforma to be filled in by a supplier shall be technically equivalent to the text of the PICS proforma in this annex and shall preserve the numbering, naming, and ordering of the PICS proforma.

A PICS that conforms to this annex shall be a conforming PICS proforma completed in accordance with the instructions for completion given in this annex.

The main part of the PICS is a fixed-format questionnaire, divided into five tables. Answers to the questionnaire are to be provided in the rightmost column, either by simply marking an answer to indicate a restricted choice (such as Yes or No) or by entering a value or a set or range of values.

C.4 Identification of the implementation

Implementation under test (IUT) identification

IUT name: ____

IUT version: ____

____

System under test (SUT) identification

SUT name: ____

Hardware configuration: ____

____

____

Operating system: ____

Product supplier

Name: ____

Address: ____

____

Telephone number: ____

Facsimile number: ____

Email address: ____

Additional information: ____

Client

Name: ____

Address: ____

____

Telephone number: ____

Facsimile number: ____

Email address: ____

Additional information: ____

PICS contact person

Name: ____

Address: ____

____

Telephone number: ____

Facsimile number: ____

Email address: ____

Additional information: ____

PICS/System conformance statement

Provide the relationship of the PICS with the system conformance statement for the system: ____

____

____

C.5 Identification of the protocol

This PICS proforma applies to IEEE Std 802.15.4™-2003.

C.6 Global statement of conformance

The implementation described in this PICS proforma meets all of the mandatory requirements of the referenced standard.

[ ] Yes

[ ] No

Note—Answering ‘No’ indicates nonconformance to the specified protocol standard. Nonsupported mandatory capabilities are to be identified in the following tables, with an explanation by the implementor explaining why the implementation is nonconforming.

The supplier will have fully complied with the requirements for a statement of conformance by completing the statement contained in this subclause. However, the supplier may find it helpful to continue to complete the detailed tabulations in the subclauses that follow.

C.7 PICS proforma tables

The following tables are composed of the detailed questions to be answered, which make up the PICS proforma. There are three major subclauses. The first subclause contains the major roles for an IEEE 802.15.4 device. The second subclause contains the major capabilities for the physical layer (PHY) and radio frequencies (RFs). The third subclause contains the major capability for the MAC sublayer. Further subclauses within these subclauses may exist.

C.7.1 Major roles for IEEE 802.15.4 devices

Table C.1—Functional device types

Item number	Item description	Reference	Status	Support		
				N/A	Yes	No
FD1	Is this a full function device (FFD)	Clause 5, 5.1	O.1			
FD2	Is this a reduced function device (RFD)	Clause 5, 5.1	O.1			
FD3	Support of 64 bit IEEE address	Clause 5, 5.2	M			
FD4	Assignment of short network address (16 bit)	Clause 5, 5.2	FD1 : M			
FD5	Support of short network address (16 bit)	Clause 5, 5.2	M			
O.1 At least one of these features shall be supported.						

C.7.2 Major capabilities for the PHY

C.7.2.1 PHY functions

Table C.2—PHY functions

Item number	Item description	Reference	Status	Support		
				N/A	Yes	No
PLF1	Transmission of packets	5.3.1, Clause 6, 6.2.1.1	M			
PLF2	Reception of packets	5.3.1, Clause 6, 6.2.1.3	M			
PLF3	Activation of radio transceiver	5.3.1, Clause 6, 6.2.2.7, 6.2.2.8	M			
PLF4	Deactivation of radio transceiver	5.3.1, Clause 6, 6.2.2.7, 6.2.2.8	M			
PLF5	Energy detection (ED)	5.3.1, Clause 6, 6.2.2.3, 6.2.2.4, 6.7.7	FD1: M O			
PLF6	Link quality indication (LQI)	5.3.1, Clause 6, 6.2.1.3, 6.7.8	M			
PLF7	Channel selection	5.3.1, Clause 6, 6.2.2.9, 6.2.2.10	M			
PLF8	Clear channel assessment (CCA)	5.3.1, Clause 6, 6.2.2.1 6.7.9	M			
PLF8.1	Mode 1	6.7.9	O.2			
PLF8.2	Mode 2	6.7.9	O.2			
PLF8.3	Mode 3	6.7.9	O.2			
O.2 At least one of these features shall be supported.						

C.7.2.2 PHY packet

Table C.3—PHY packet

Item number	Item description	Reference	Status	Support		
				N/A	Yes	No
PLP1	PHY protocol data unit (PPDU) packet	6.3	M			

C.7.2.3 RF

Table C.4—RF

Item number	Item description	Reference	Status	Support		
				N/A	Yes	No
RF1	868/915 MHz PHY	5.3.1, Clause 6, Table 1, 6.6.1, 6.6	O.3			
RF1.1	868-868.6 MHz	5.3.1, Clause 6, Table 1, 6.6.1	M			
RF1.2	902-928 MHz	5.3.1, Clause 6, Table 1, 6.6.1	M			
RF2	2450 MHz PHY	5.3.1, Clause 6, Table 1, 6.6.1, 6.5	O.3			
O.3 At least one of these features shall be supported.						

C.7.3 Major capabilities for the MAC sublayer

C.7.3.1 MAC sublayer functions

Table C.5—MAC sublayer functions

Item number	Item description	Reference	Status	Support		
				N/A	Yes	No
MLF1	Transmission of data	5.3.2, 7.1.1	M			
MLF1.1	Purge data	7.1.1.4, 7.1.1.5	FD1 : M FD2 : O			
MLF2	Reception of data	5.3.2, 7.1.1	M			
MLF2.1	Promiscuous mode	7.5.6.2, 7.4.2	FD1 : M FD2 : O			
MLF2.2	Control of PHY receiver	7.1.10	M			
MLF3	Beacon management	5.3.2, Clause 7	M			
MLF3.1	Transmit beacons	5.3.2, Clause 7, 7.5.2.4	FD1 : M FD2: O			
MLF3.2	Receive beacons	5.3.2, Clause 7, 7.1.5, 7.5.4.1	M			
MLF4	Channel access mechanism	5.3.2, Clause 7, 7.5.1	M			

Table C.5—MAC sublayer functions *(continued)*

Item number	Item description	Reference	Status	Support		
				N/A	Yes	No
MLF5	Guaranteed time slot (GTS) management	5.3.2, Clause 7, 7.1.7, 7.3.3, 7.5.7	FD1:M FD2: O			
MLF5.1	GTS management (allocation)	5.3.2, Clause 7, 7.1.7, 7.3.3, 7.5.7	FD1 : M FD2 : O			
MLF5.2	GTS management (request)	5.3.2, Clause 7, 7.1.7, 7.3.3, 7.5.7	FD1 : M FD2 : O			
MLF6	Frame validation	5.3.2, 5.4.4.3, 7.1.1.3.2, 7.2, 7.5.6.2	M			
MLF7	Acknowledged frame delivery	5.3.2, Clause 7, 7.1.1.3.2, 7.2.1.1.4, 7.5.6.4	M			
MLF8	Association and disassociation	5.3.2, Clause 7, 7.1.3, 7.1.4, 7.3.1, 7.5.3	M			
MLF9	Security	5.3.2, 5.4.6, Clause 7, 7.5.8	M			
MLF9.1	Unsecured mode	5.4.6.2.1, 7.5.8, 7.5.8.2	M			
MLF9.2	ACL mode	5.4.6.2.2, 7.5.8, 7.5.8.3	MLF9.3: M O			
MLF9.3	Secured mode	5.4.6.2.3, 7.5.8, 7.5.8.4	O			
MLF 9.3.1	Access control	5.4.6.1.1	O.4			
MLF 9.3.2	Data encryption	5.4.6.1.2	O.4			
MLF 9.3.3	Frame integrity	5.4.6.1.3	O.4			
MLF 9.3.4	Sequential freshness	5.4.6.1.4	O.4			
MLF10.1	ED	7.5.2.1, 7.5.2.1.1	FD1: M FD2: O			
MLF10.2	Active scanning	7.5.2.1, 7.5.2.1.2	FD1: M FD2: O			
MLF10.3	Passive scanning	7.5.2.1, 7.5.2.1.3	M			
MLF10.4	Orphan scanning	7.5.2.1, 7.5.2.1.4	M			
MLF11	Control/define/determine/declare superframe structure	5.4.1, 7.5.1.1	FD1: M			

Table C.5—MAC sublayer functions *(continued)*

Item number	Item description	Reference	Status	Support		
				N/A	Yes	No
MLF12	Follow/use superframe structure	5.4.1, 7.5.1.1	M			
MLF13	Store one transaction	7.5.5	FD1: M			
O.4 At least one of these features shall be supported.						

C.7.3.2 MAC frames

Table C.6—MAC frames

Item number	Item description	Reference	Transmitter		Receiver	
			Status	Support N/A Yes No	Status	Support N/A Yes No
MF1	Beacon	5.4.3.1, 7.2.2.1	FD1: M		M	
MF2	Data	5.4.3.2, 7.2.2.2	M		M	
MF3	Acknowledgment	5.4.3.3, 7.2.2.3	M		M	
MF4	Command	5.4.3.4, 7.2.2.4	M		M	
MF4.1	Association request	5.4.3.4, 7.2.2.4, 7.3.1.1	M		FD1: M	
MF4.2	Association response	5.4.3.4, 7.2.2.4, 7.3.1.2	FD1: M		M	
MF4.3	Disassociation notification	5.4.3.4, 7.2.2.4, 7.3.1.3	M		M	
MF4.4	Data request	5.4.3.4, 7.2.2.4, 7.3.2.1	M		FD1: M	
MF4.5	PAN identifier conflict notification	5.4.3.4, 7.2.2.4, 7.3.2.2	M		FD1: M	
MF4.6	Orphaned device notification	5.4.3.4, 7.2.2.4, 7.3.2.3	M		FD1: M	
MF4.7	Beacon request	5.4.3.4, 7.2.2.4, 7.3.2.4	FD1: M		FD1: M	

Table C.6—MAC frames *(continued)*

Item number	Item description	Reference	Transmitter		Receiver	
			Status	Support N/A Yes No	Status	Support N/A Yes No
MF4.8	Coordinator realignment	5.4.3.4, 7.2.2.4, 7.3.2.5	FD1: M		M	
MF4.9	GTS request	5.4.3.4, 7.2.2.4, 7.3.3.1	MLF5 : M		MLF5: M	

Annex D

(informative)

Formal description of IEEE 802.15.4 operation

D.1 Specification and description language (SDL)

This annex describes the protocol behavior and abstract data structure of the MAC sublayer and PHY SDL model. That is it explains the behavior of these protocols. It does not specify the exact coding of the frames, packets, or messages or the parameters contained herein.

D.1.1 SDL overview

SDL is an object-oriented, formal language defined by The International Telecommunications Union–Telecommunications (ITU–T) Standardization Sector [formerly Comité Consultatif International Telegraphique et Telephonique (CCITT)] as Recommendation Z.100. The language is intended for the specification of complex, event-driven, real-time, and interactive applications involving many concurrent activities that communicate using discrete signals. SDL is usually used in combination with other languages: message sequence chart, Abstract Syntax Notation Number 1 (ASN.1), and TTCN. The use of traditional SDL state models, message sequence charts, and ASN.1 is a powerful combination that covers most aspects of system engineering. These languages have been studied in the same group within the ITU-T. ITU-T Recommendation Z.100 coupled with ITU-T Recommendation Z.105 and Recommendation Z.107 define the use of SDL with ASN.1. ITU-T Recommendation Z.109 defines a unified modeling language (UML) profile for SDL, and ITU-T Recommendation Z.120 defines message sequence charts. The SDL source is written using SDL-88, with one exception. Therefore, as long as SDL-92 and SDL-2000 are backward/ forward compatible with SDL-88, then so is the SDL source. The exception is the use of the ‘choice’ construct, which is currently specific to Telelogic’s product and not part of the SDL-XX recommendations. However, the construct is similar to the one used in the ASN.1, which is now part of the SDL recommendations.

NOTES:

1—The SDL definitions in this annex should be usable with any SDL tool that supports the SDL-92 or SDL-2000 update of ITU-T Recommendation Z.100. More info: http://www.sdl-forum.org/Tools/Commercial.htm.

2—The SDL code in this annex was generated using the Telelogic Tau SDL Suite (on the Sun OS v5.6) v4.2; from Telelogic AB, Malmö, Sweden (+46 40 17 47 00; http://www.telelogic.se); U.S. office in Irvine, CA (+1 949 830 8022; http://www.telelogic.com). The suite can be integrated with any operating system or kernel for automatic generation of complete real-time applications.

3—The use of Telelogic’s product to prepare this annex does not constitute an endorsement of Telelogic Tau SDL Suite by the IEEE LAN/MAN Standards Committee or by the IEEE.

4—All SDL behavior diagrams are 150mm x 180mm so that both headers and footers can be added to the annex pages that con-tain these diagrams and to satisfy the publishing margin requirements.

D.1.2 SDL system overview

The SDL system is shown in Figure D.1. It shows the PHY and MAC sublayer following the IEEE 802 architecture as SDL blocks. Each of these blocks are described in detail in their respective clauses (i.e., D.2 and D.3). There are five SAPs shown that interface with the radio (RF), PHY/MAC (PD and PLME), or the upper layers (MCPS and MLME). The SAPs are defined as channels and gates in the SDL system. The primitives that are carried by these channels are labelled as signallist. See D.4 for the detailed member signals for each of these signallists.

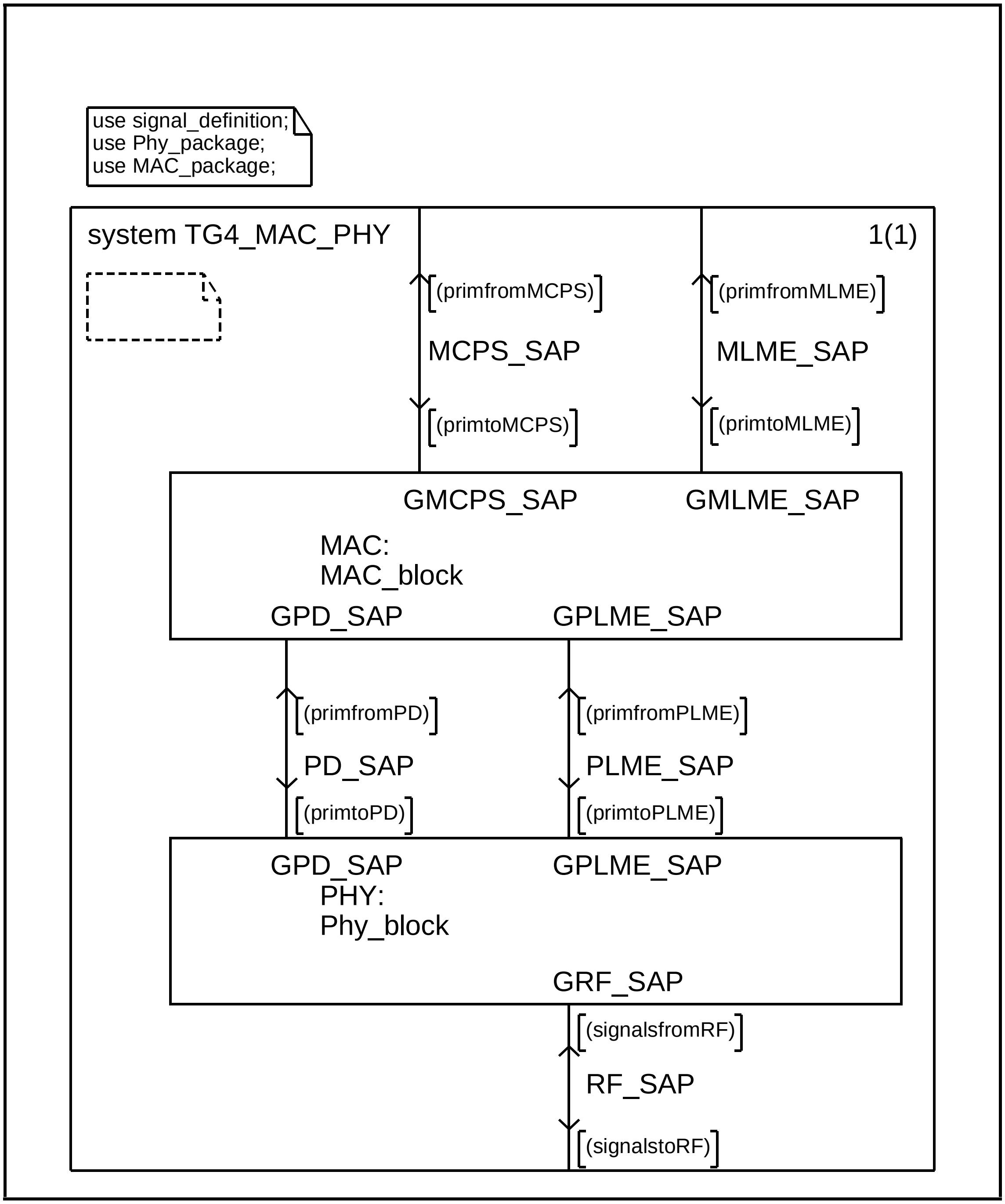


Figure D.1—IEEE 802.15.4 system view

D.1.3 IEEE 802.15.4 SDL model overview

The IEEE 802.15.4 SDL model describes only the protocol behavior and abstract data structure. It is meant to explain the behavior (i.e., the interaction) of the protocols. It does not contain “code.” In other words, the actual bit structure and coding are not supported. Also implementation issues, e.g., database management, are not part of the requirements, but are used as an example, so that the SDL tool extensions, e.g., the simulator and validator, can be used.

The protocols modeled are the PHY and the MAC sublayer.

Annex D.2 contains the Physical block (and package).

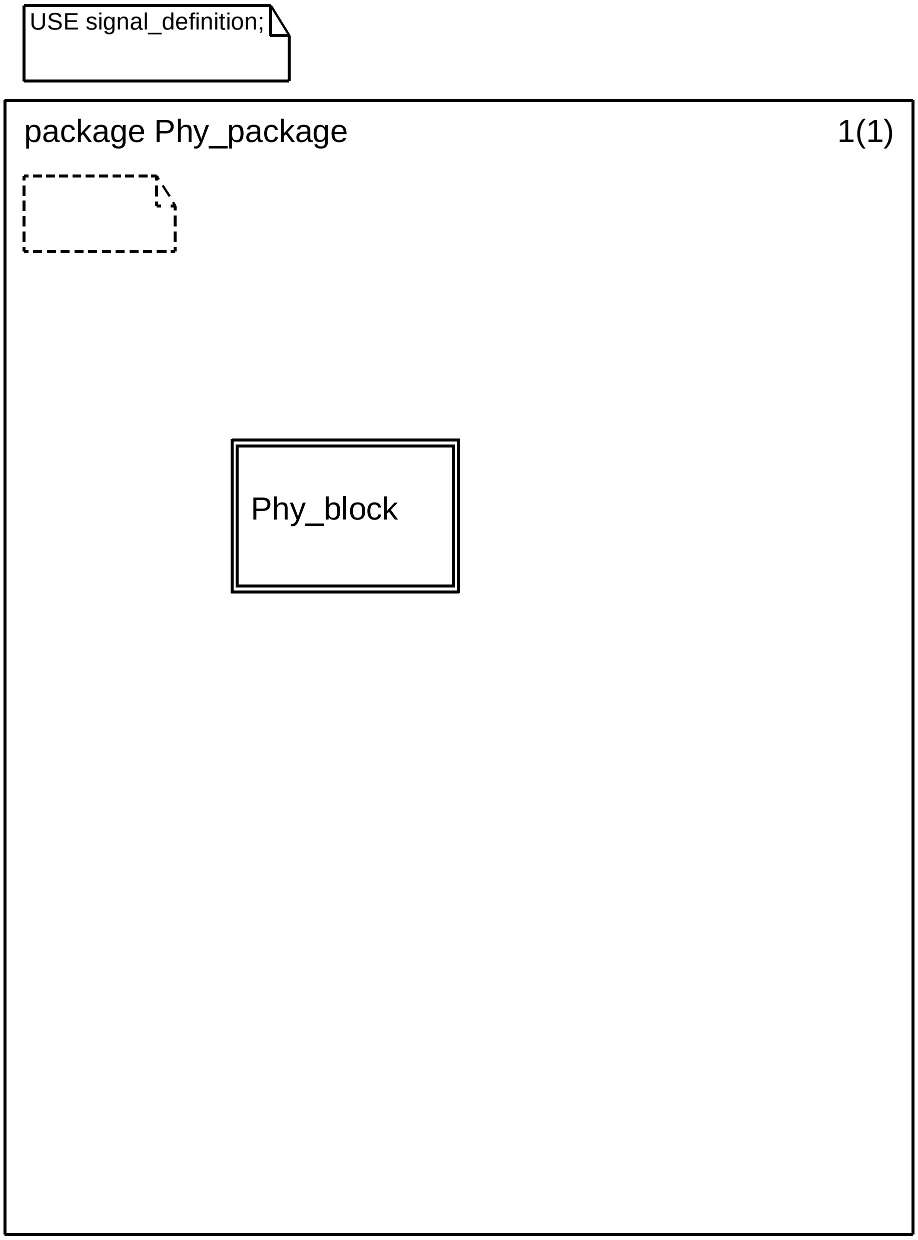
Annex D.3 contains the MAC sublayer (and package).

Annex D.4 contains the SDL package called “Signal_definition” contains all the signals, signallists, newtypes, and synonyms used by the entire system.

The first page of each process contains the signals (i.e., signallists) entering and leaving this process via SDL gates. The next pages contain all the variables, timers, and procedures used within this process. The remainder of the pages cover the protocol described by the process.

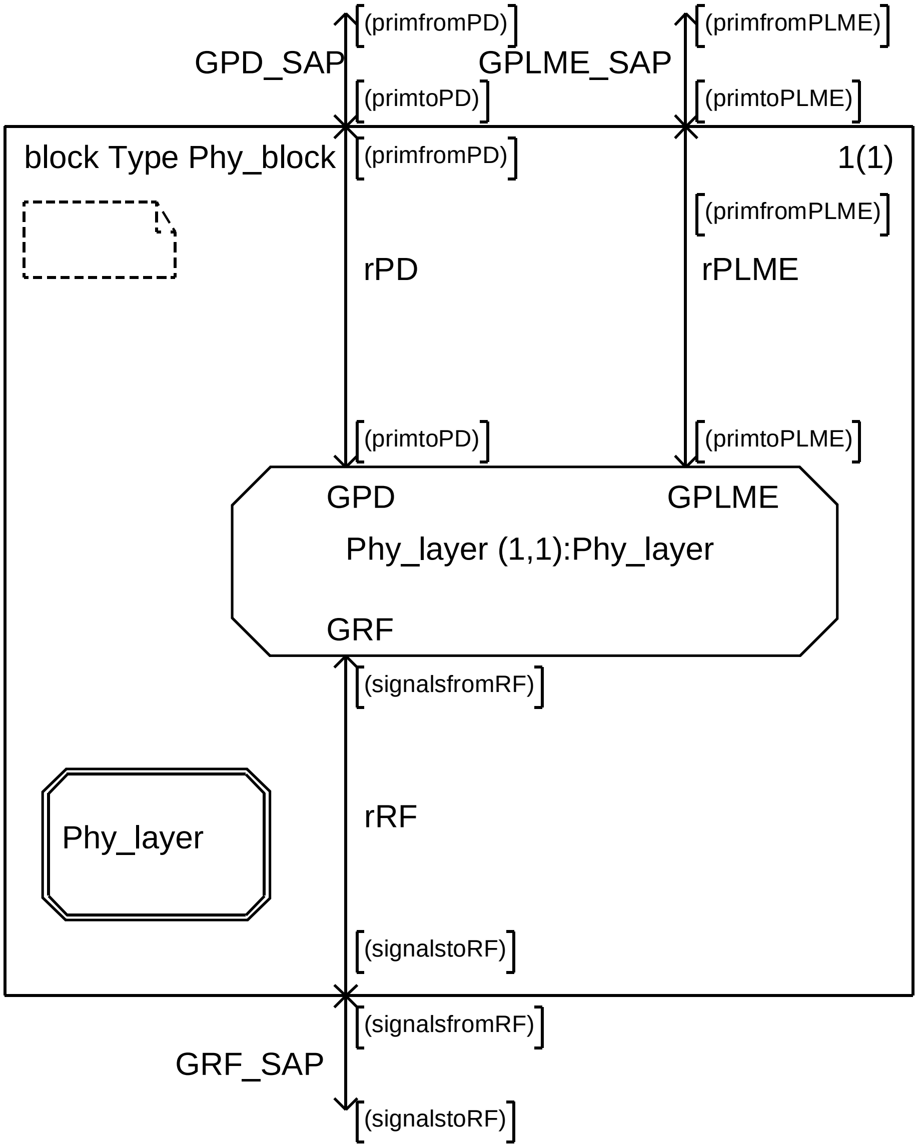
All variables are to be initialized in an “initialize” procedure. Some are set in an “implementation” procedure when the variable is truly an implementation decision. The goal here is to identify which variables are needed and defined by the protocol with default values and which variables are needed, but whose values are an implementation decision.

D.2 IEEE 802.15.4 PHY package



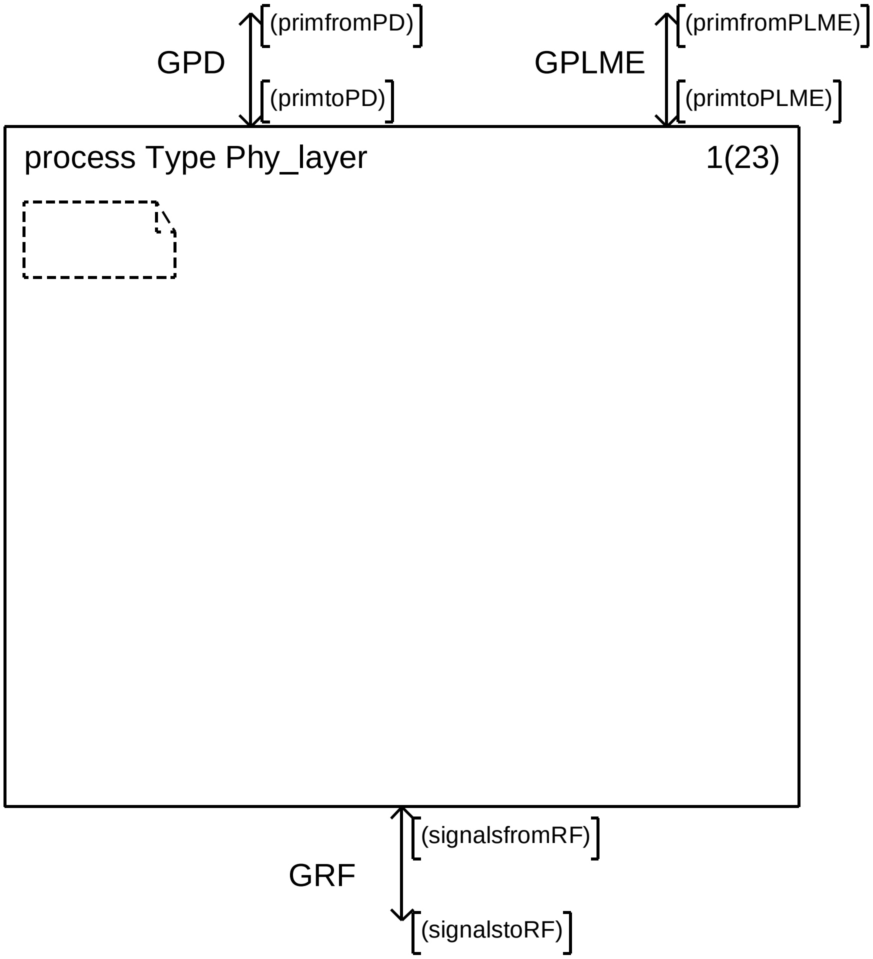
D.2.1 Block type Phy_block

.



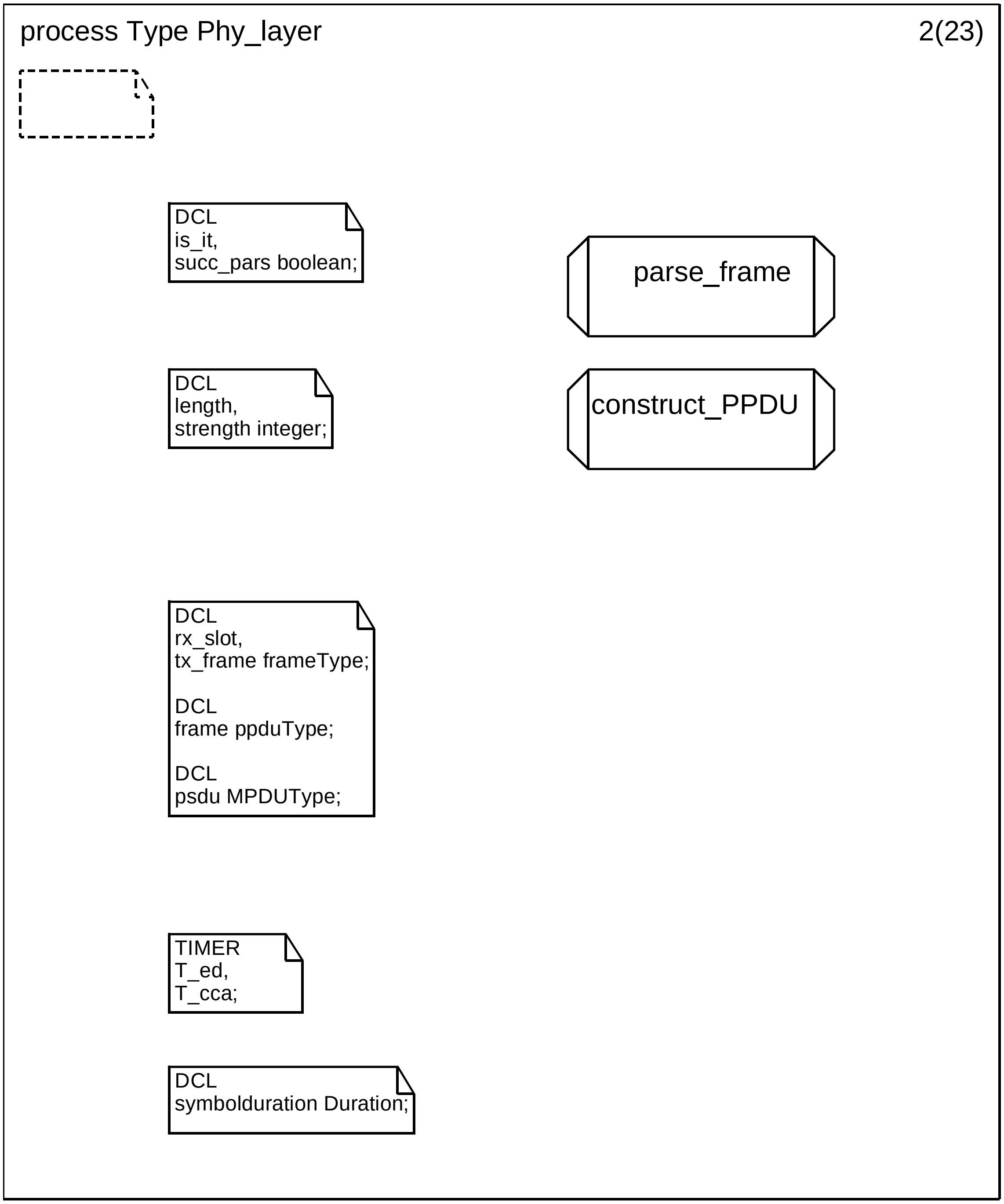
D.2.1.1 Process type Phy_Layer (1)

.



D.2.1.2 Process type Phy_Layer (2)

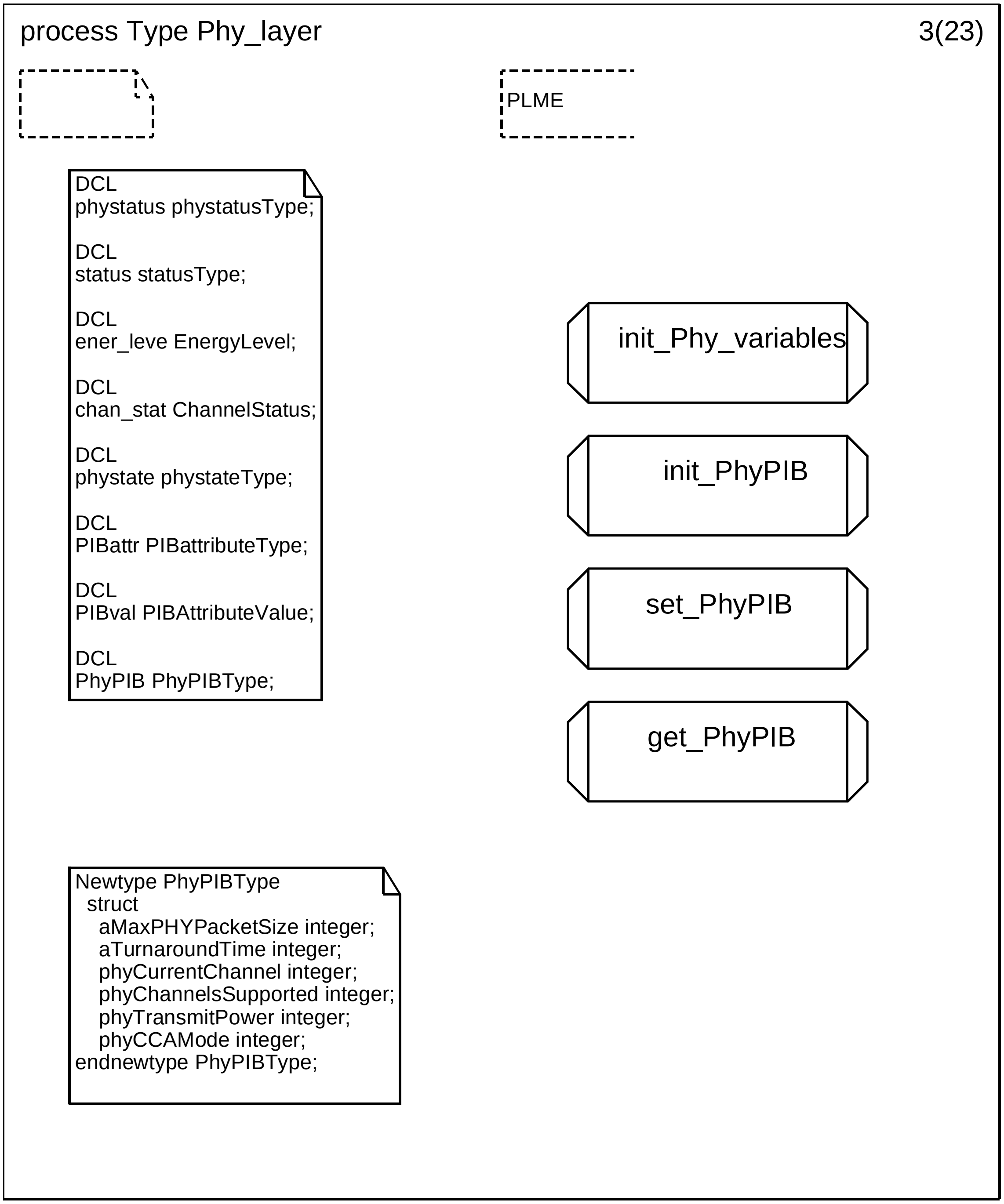
.



```
DCL
is_it,
succ_pars boolean;

DCL
length,
strength integer;

DCL
rx_slot,
tx_frame frameType;

DCL
frame ppduType;

DCL
psdu MPDUType;

TIMER
T_ed,
T_cca;

DCL
symbolduration Duration;
```

D.2.1.3 Process type Phy_Layer (3)

.

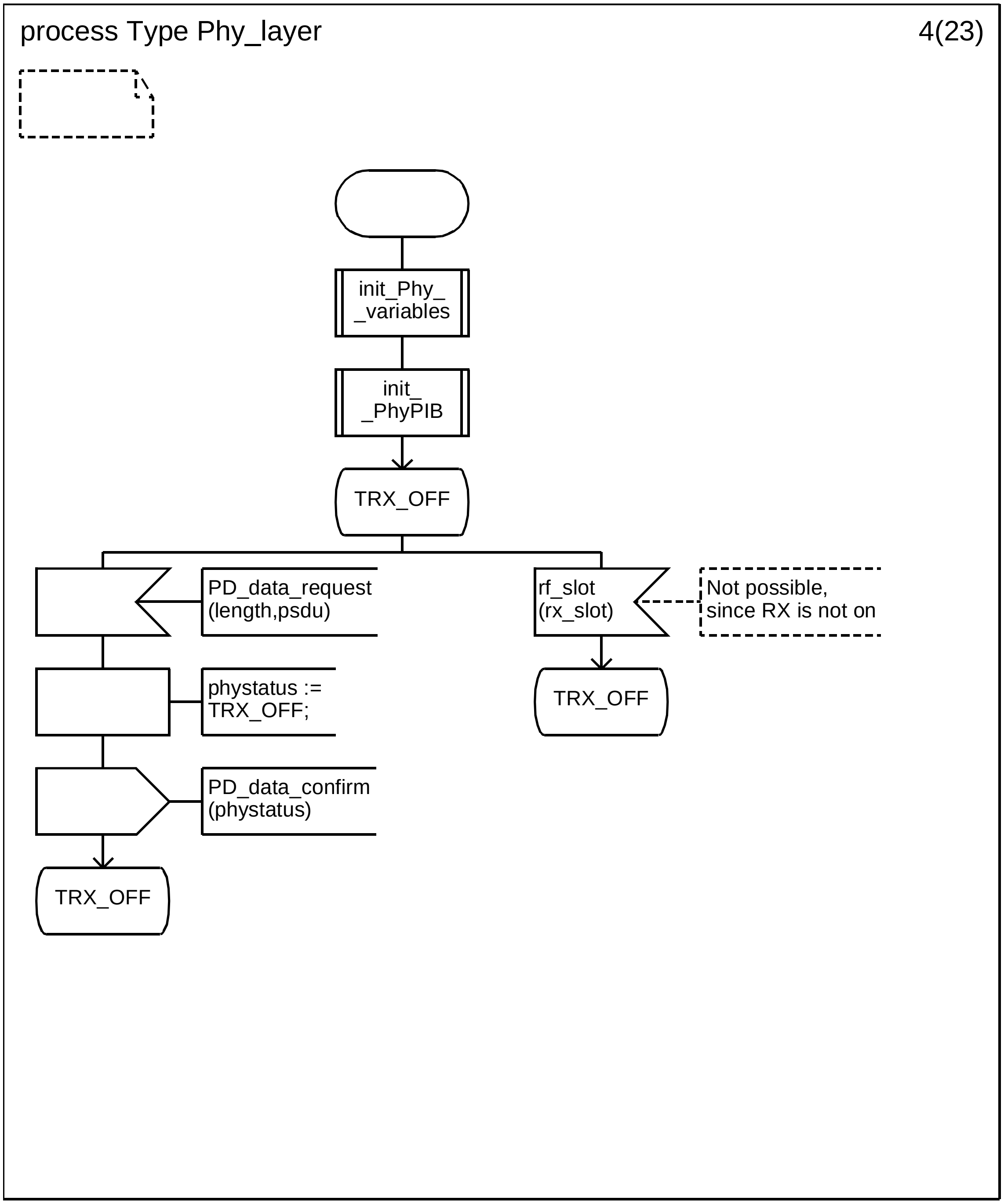


```
DCL
phystatus phystatusType;

DCL
status statusType;

DCL
ener_leve EnergyLevel;

DCL
chan_stat ChannelStatus;

DCL
phystate phystateType;

DCL
PIBattr PIBattributeType;

DCL
PIBval PIBAttributeValue;

DCL
PhyPIB PhyPIBType;
```

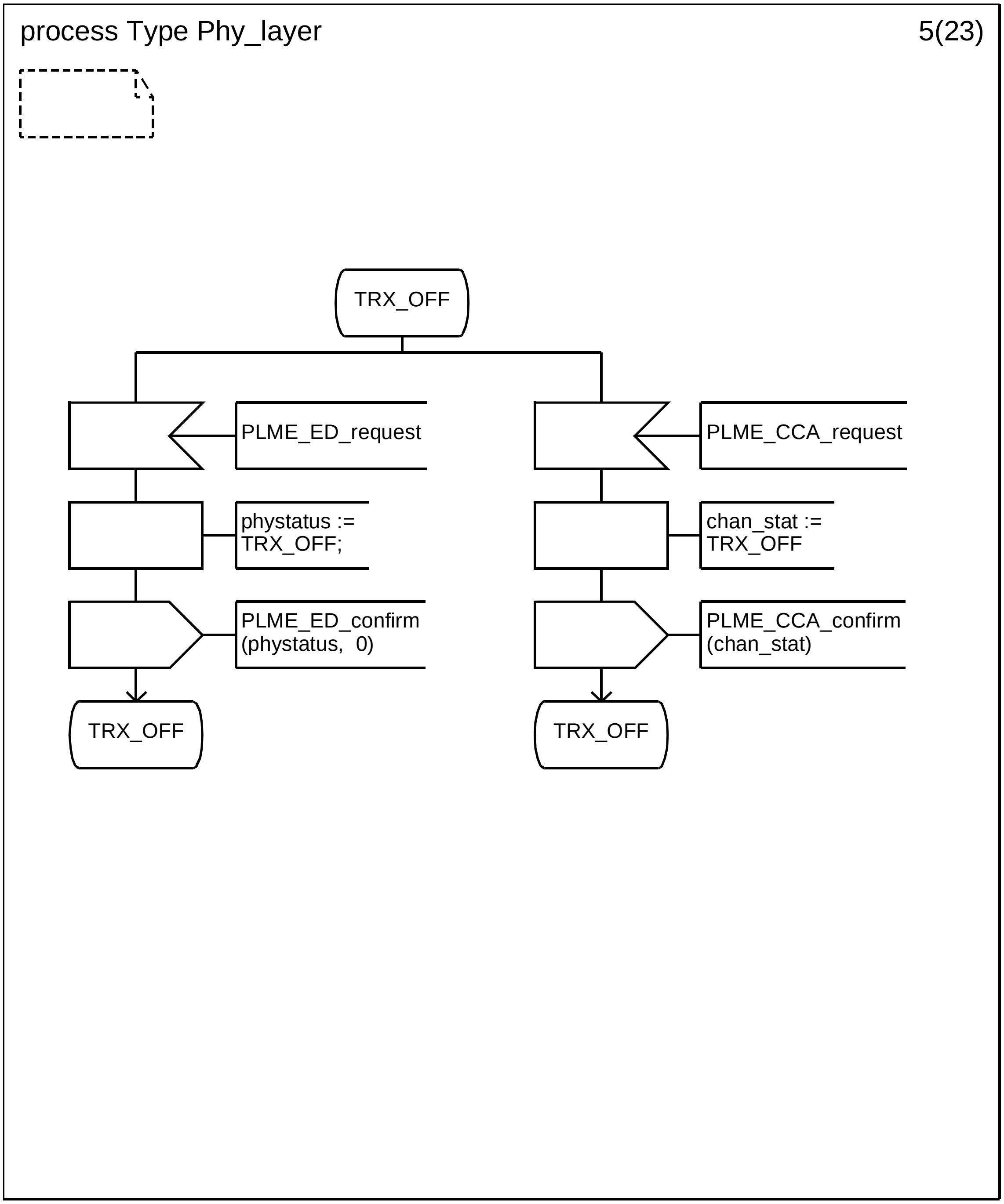
```
Newtype PhyPIBType
  struct
    aMaxPHYPacketSize integer;
    aTurnaroundTime integer;
    phyCurrentChannel integer;
    phyChannelsSupported integer;
    phyTransmitPower integer;
    phyCCAMode integer;
endnewtype PhyPIBType;
```

D.2.1.4 Process type Phy_Layer (4)

.

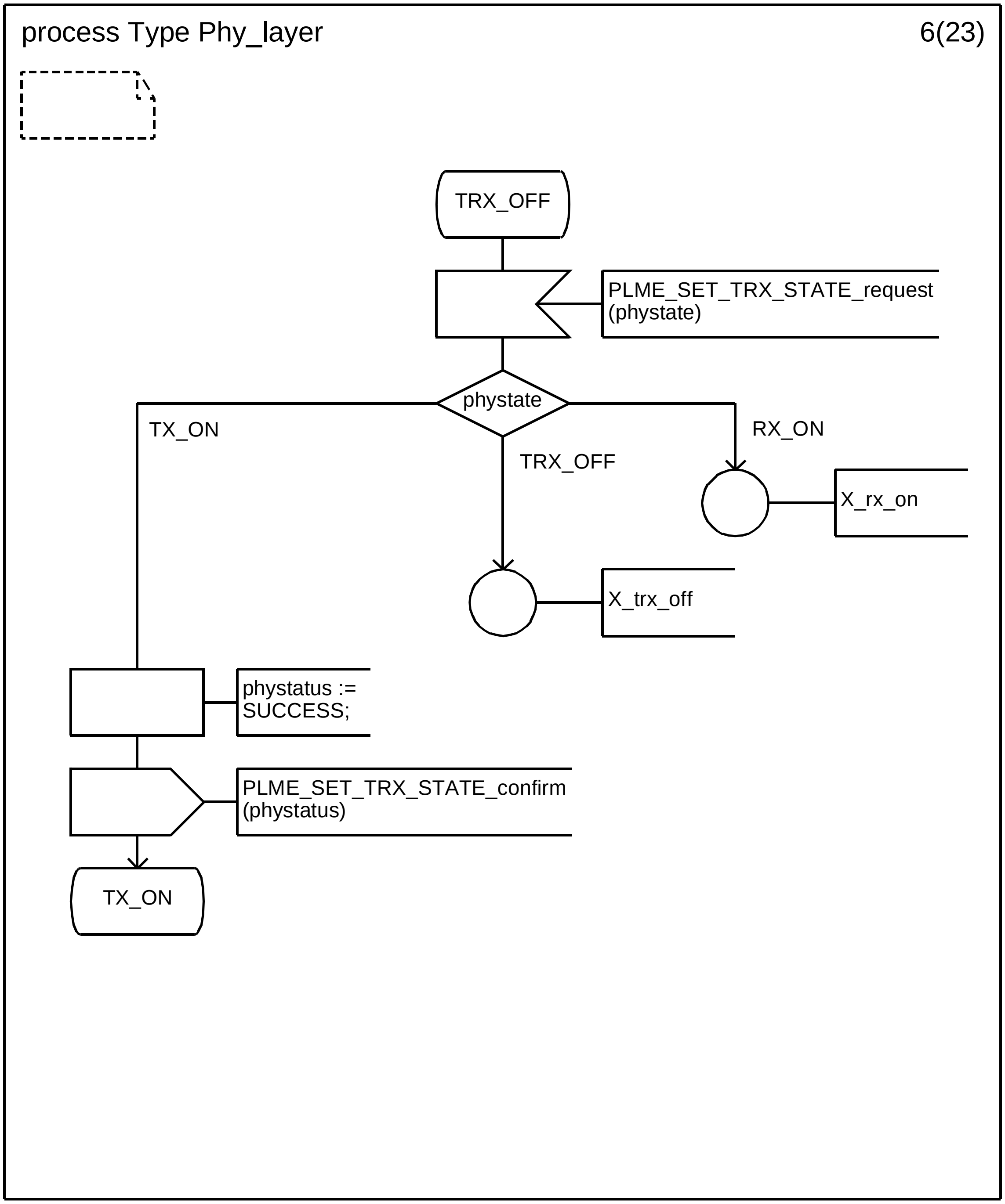


D.2.1.5 Process type Phy_Layer (5)



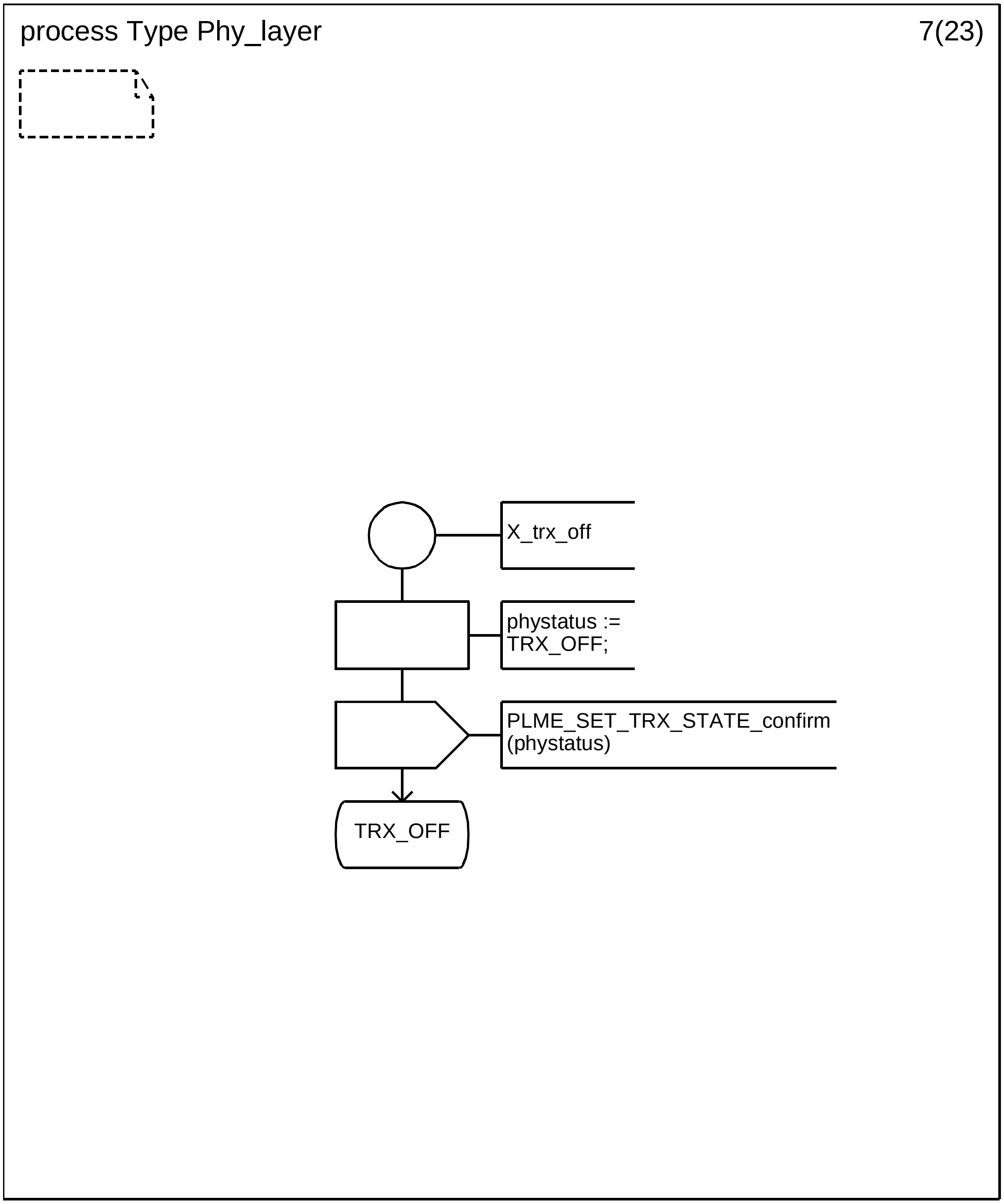
D.2.1.6 Process type Phy_Layer (6)

.



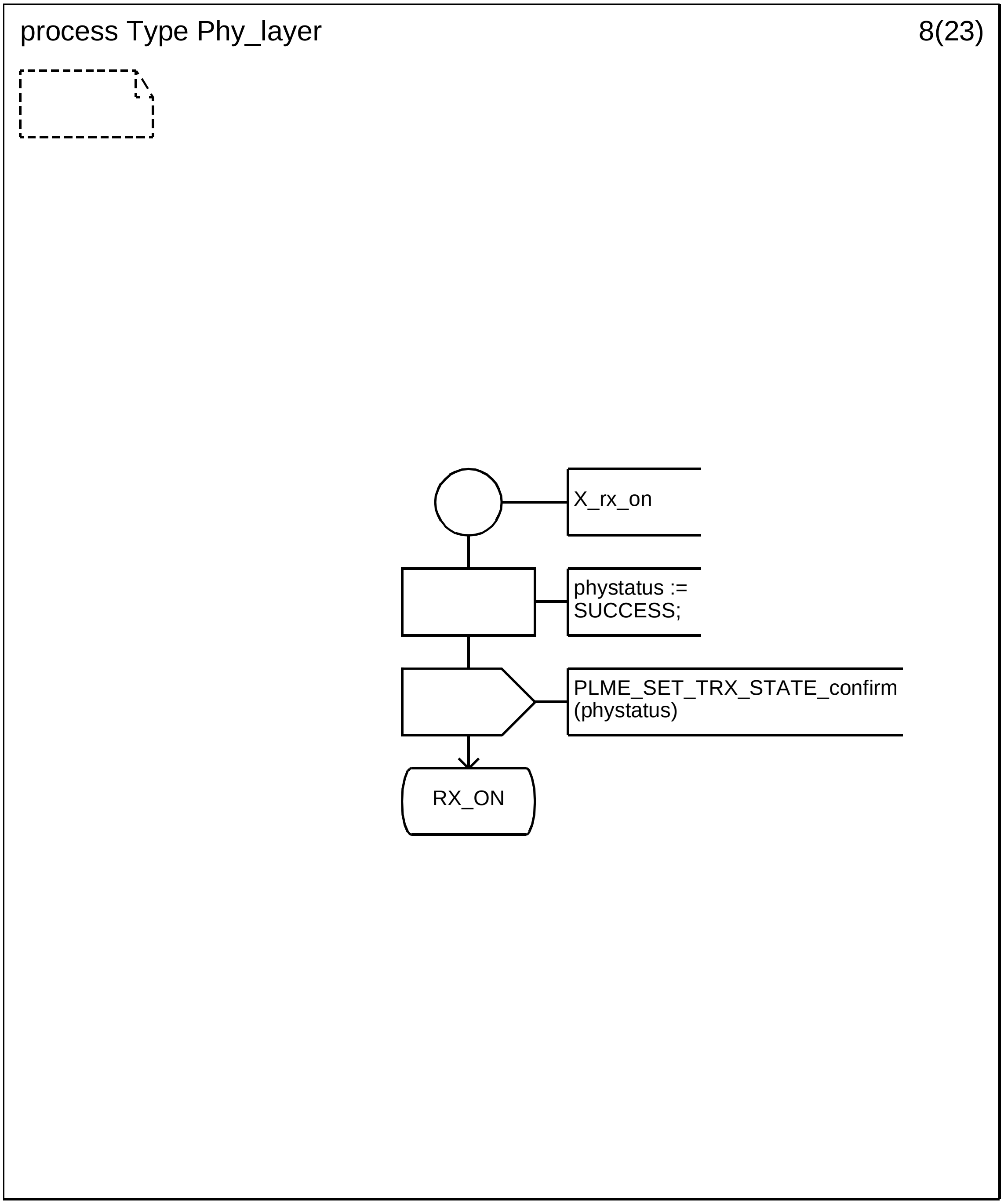
D.2.1.7 Process type Phy_Layer (7)

.

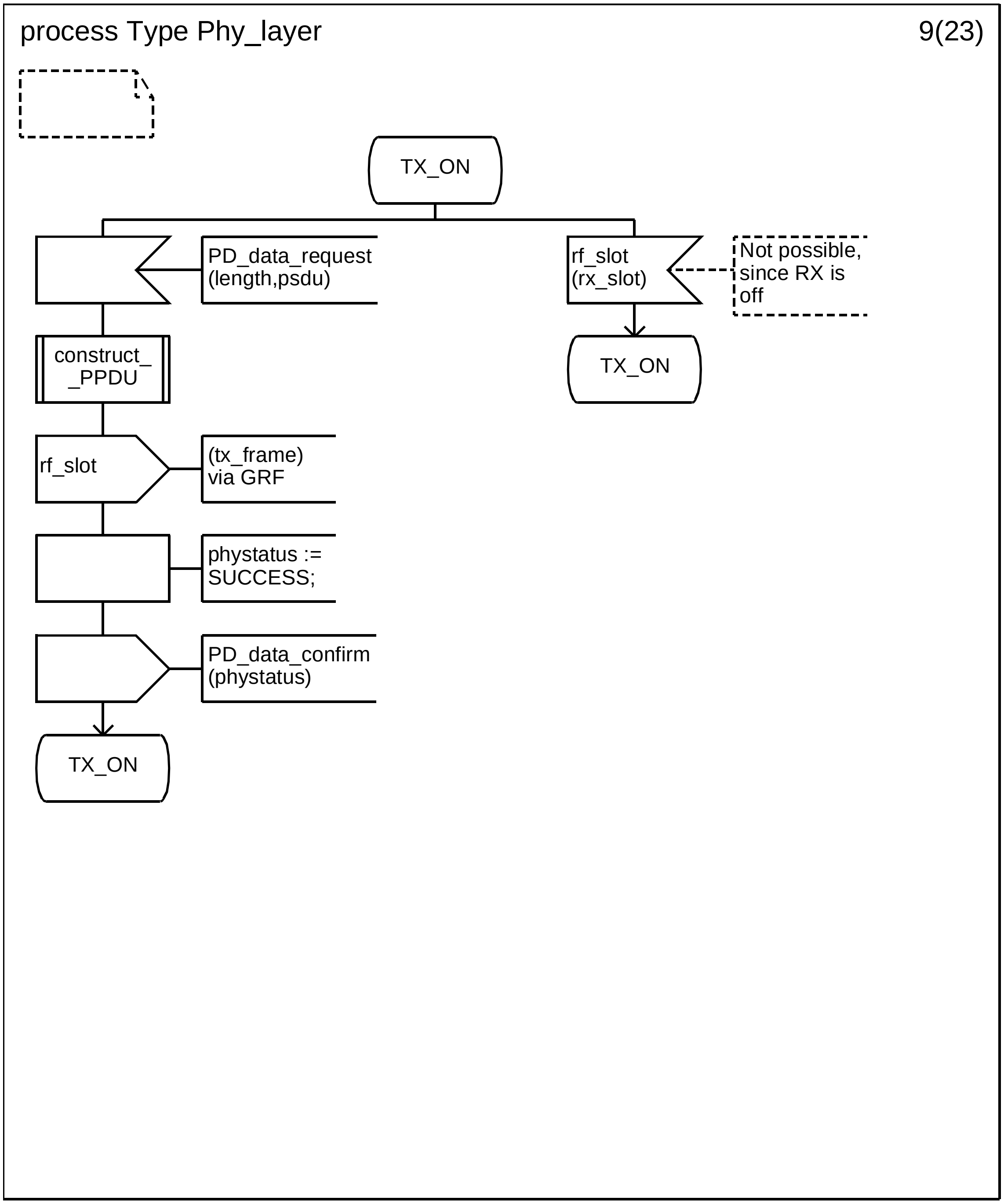


D.2.1.8 Process type Phy_Layer (8)

.

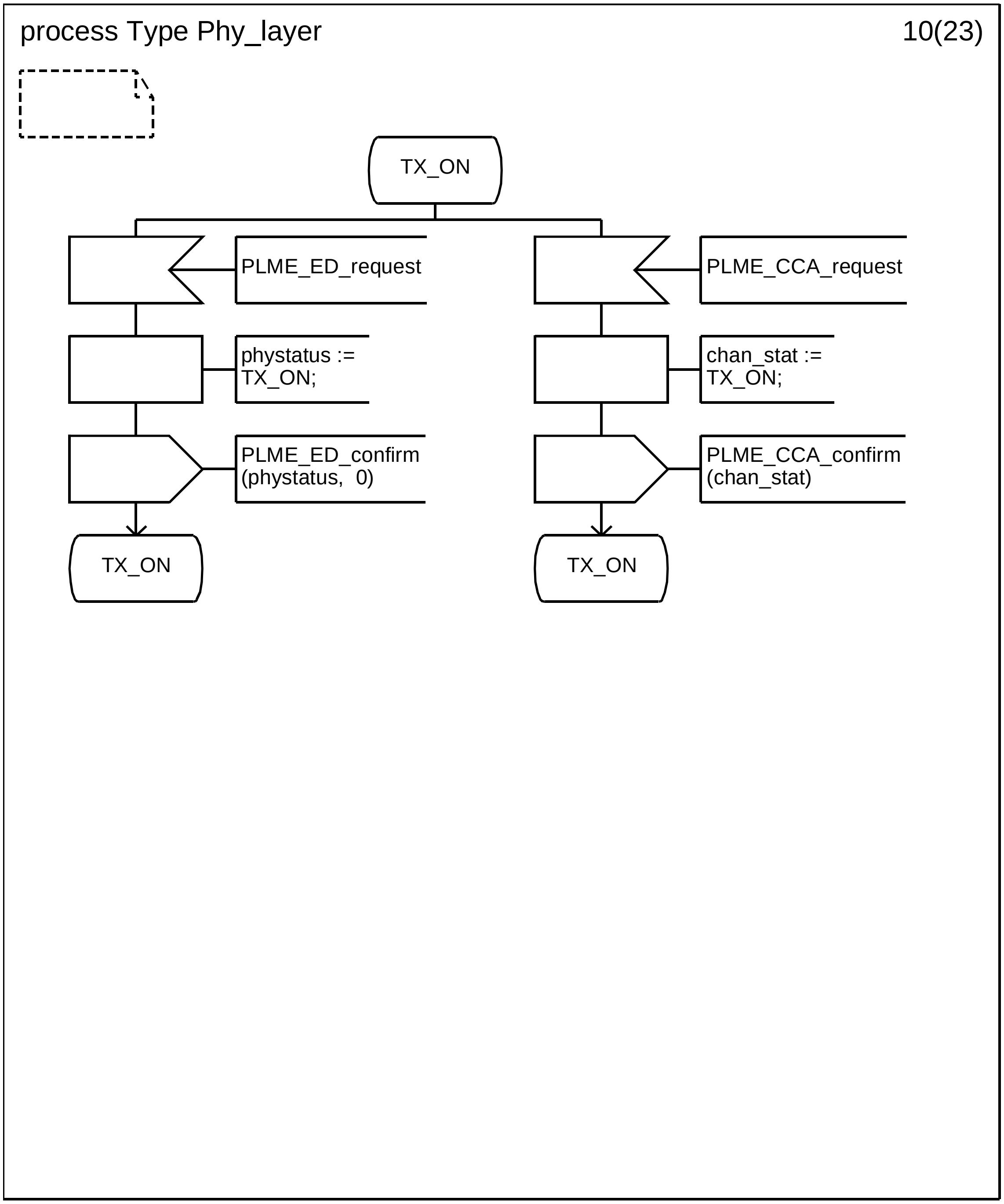


D.2.1.9 Process type Phy_Layer (9)



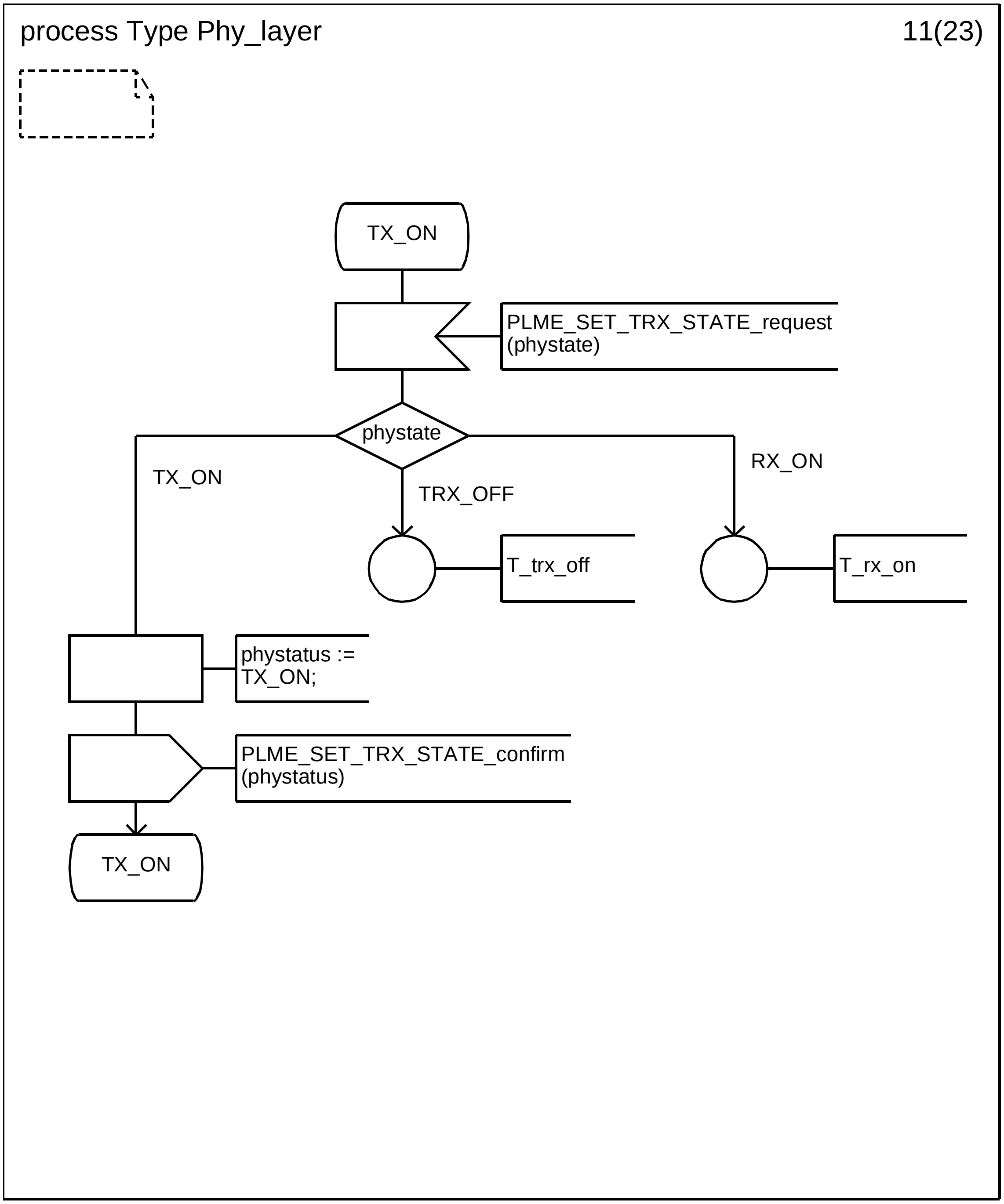
D.2.1.10 Process type Phy_Layer (10)

.



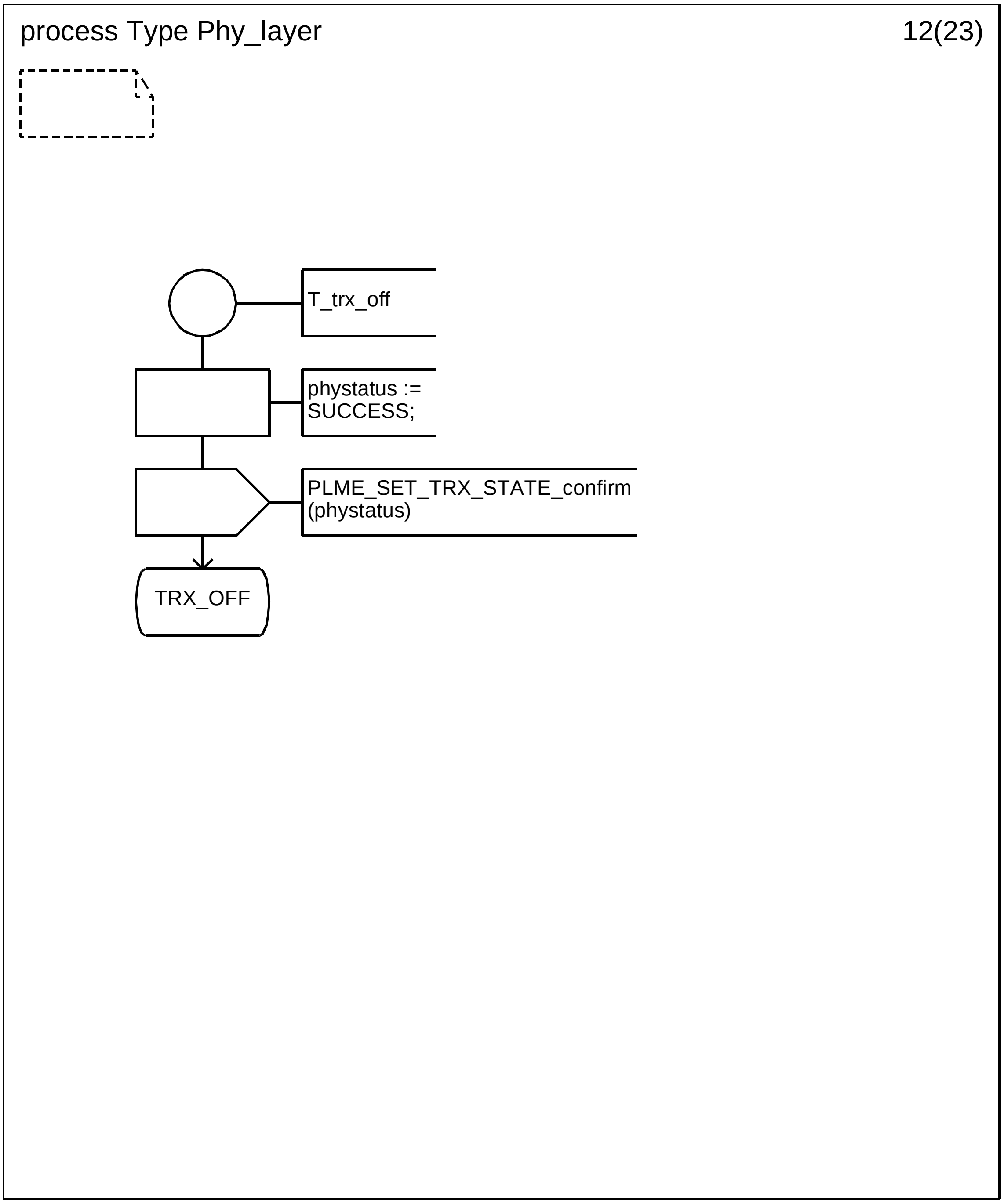
D.2.1.11 Process type Phy_Layer (11)

.



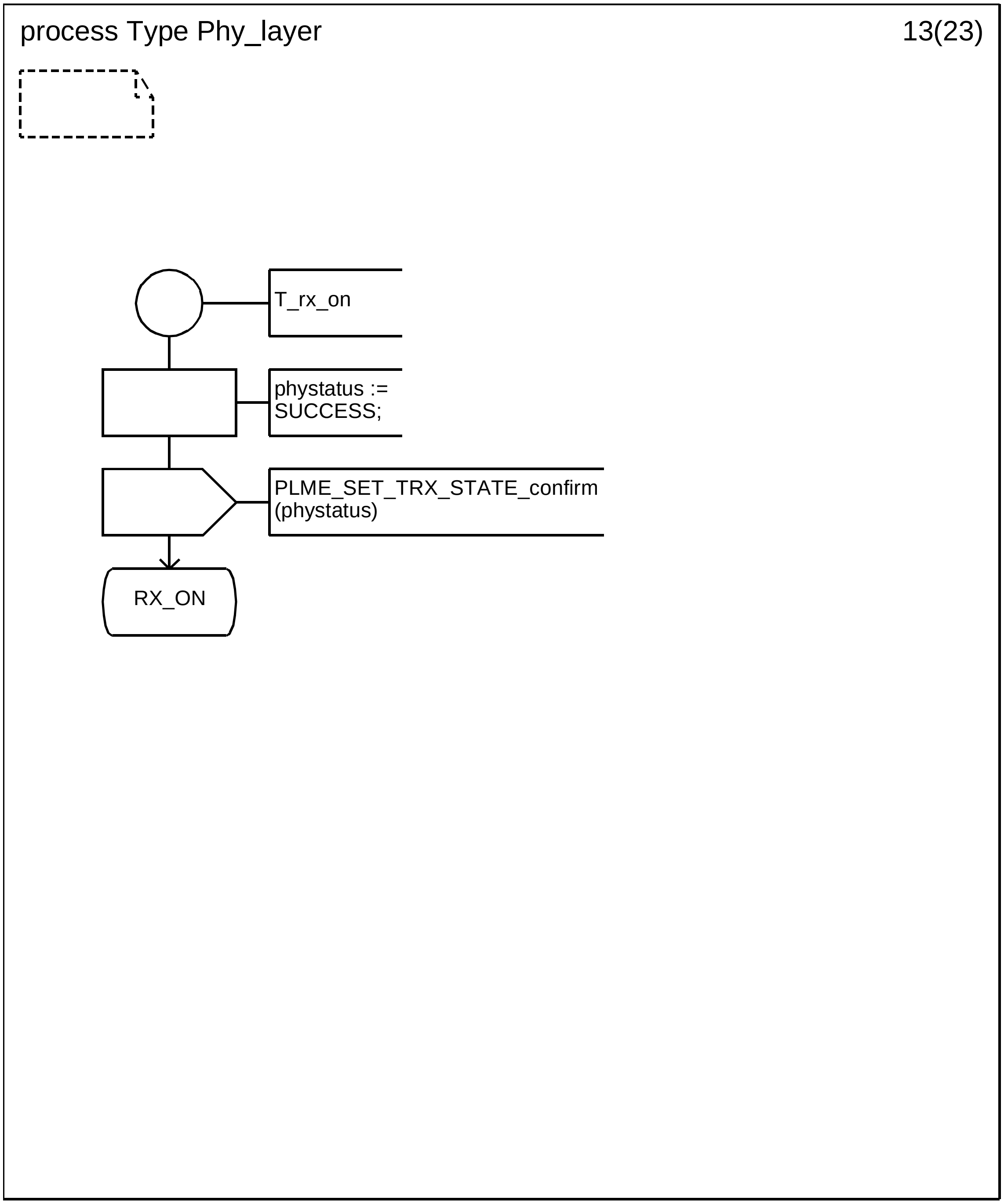
D.2.1.12 Process type Phy_Layer (12)

.



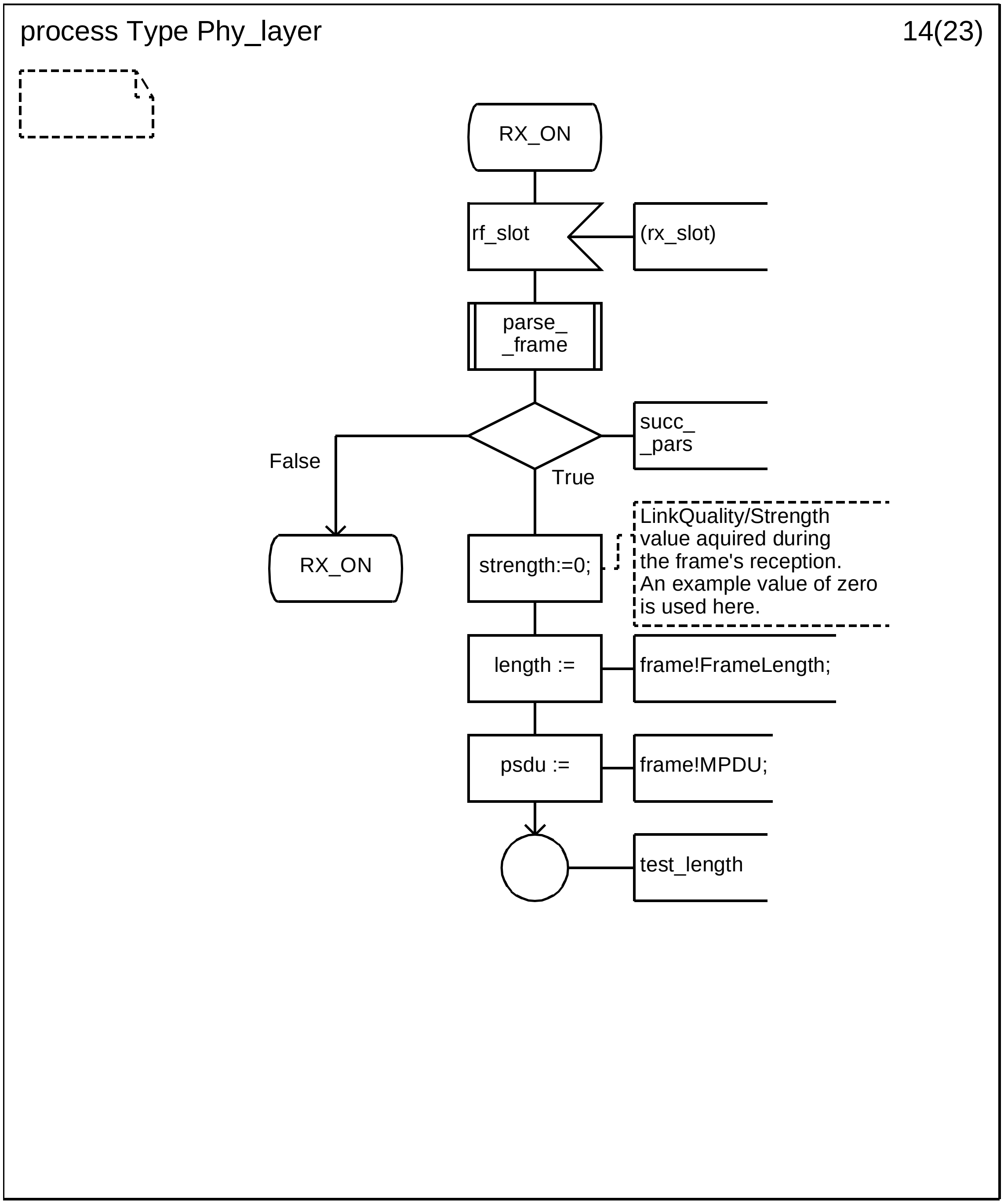
D.2.1.13 Process type Phy_Layer (13)

.



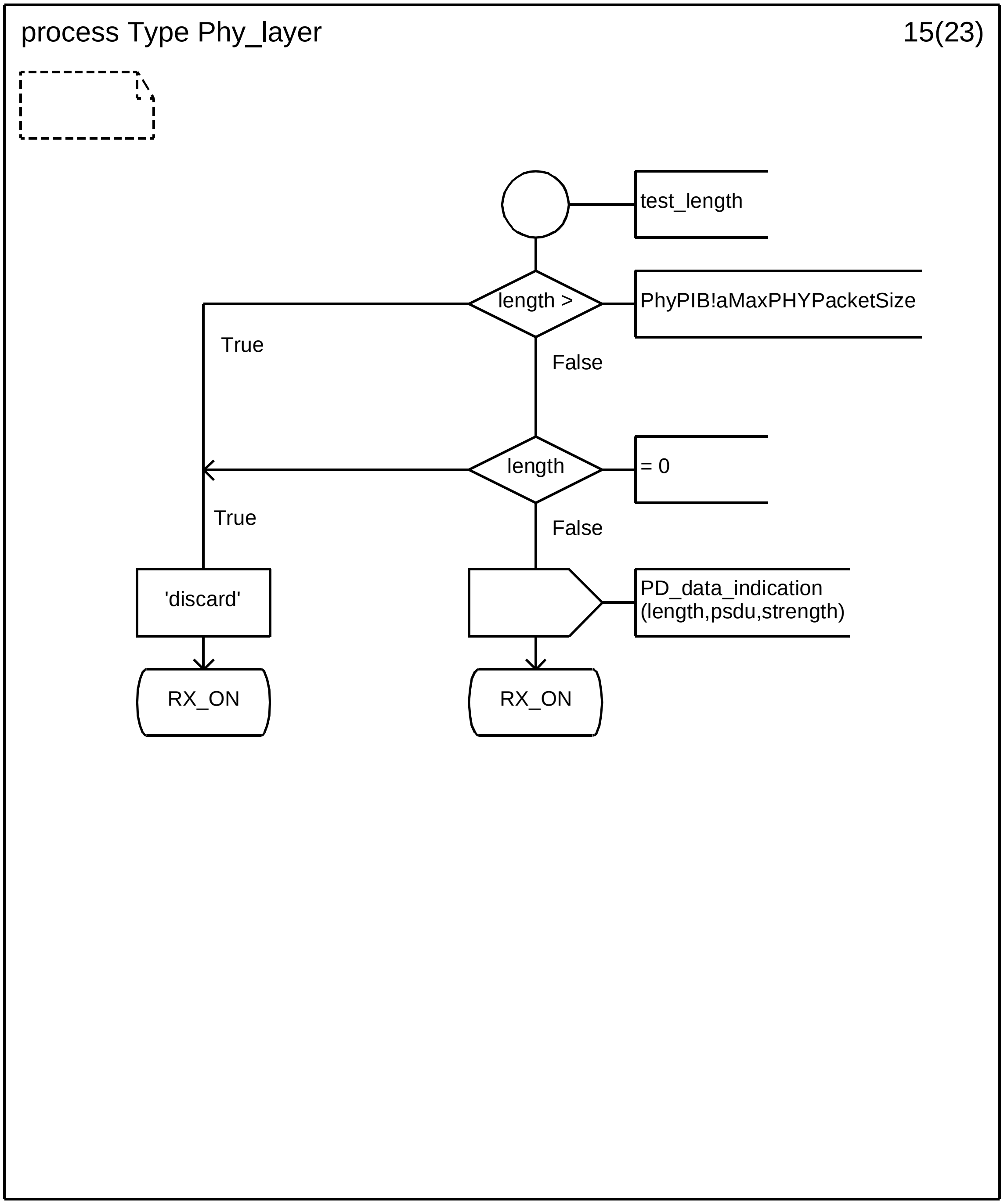
D.2.1.14 Process type Phy_Layer (14)

.



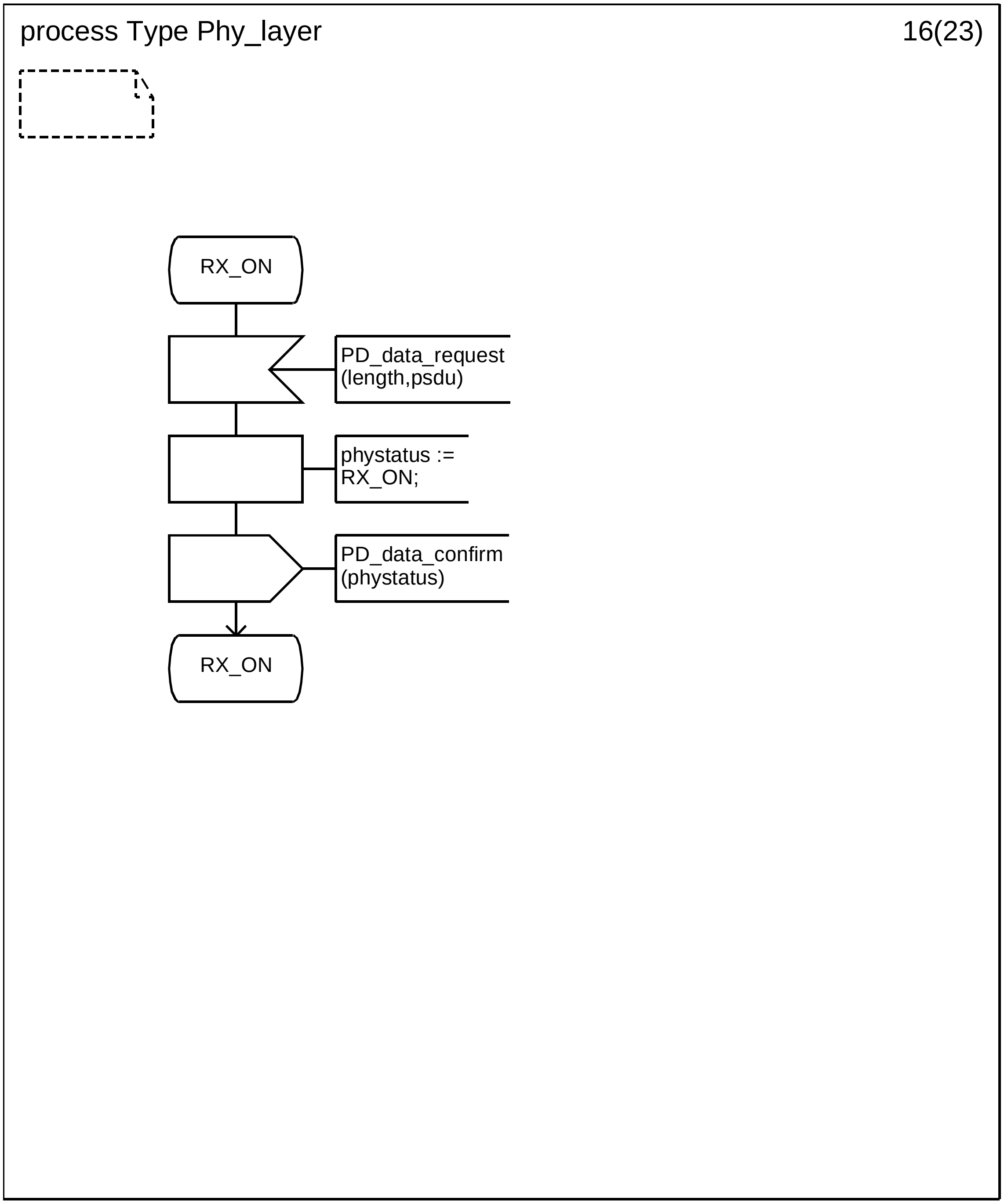
D.2.1.15 Process type Phy_Layer (15)

.



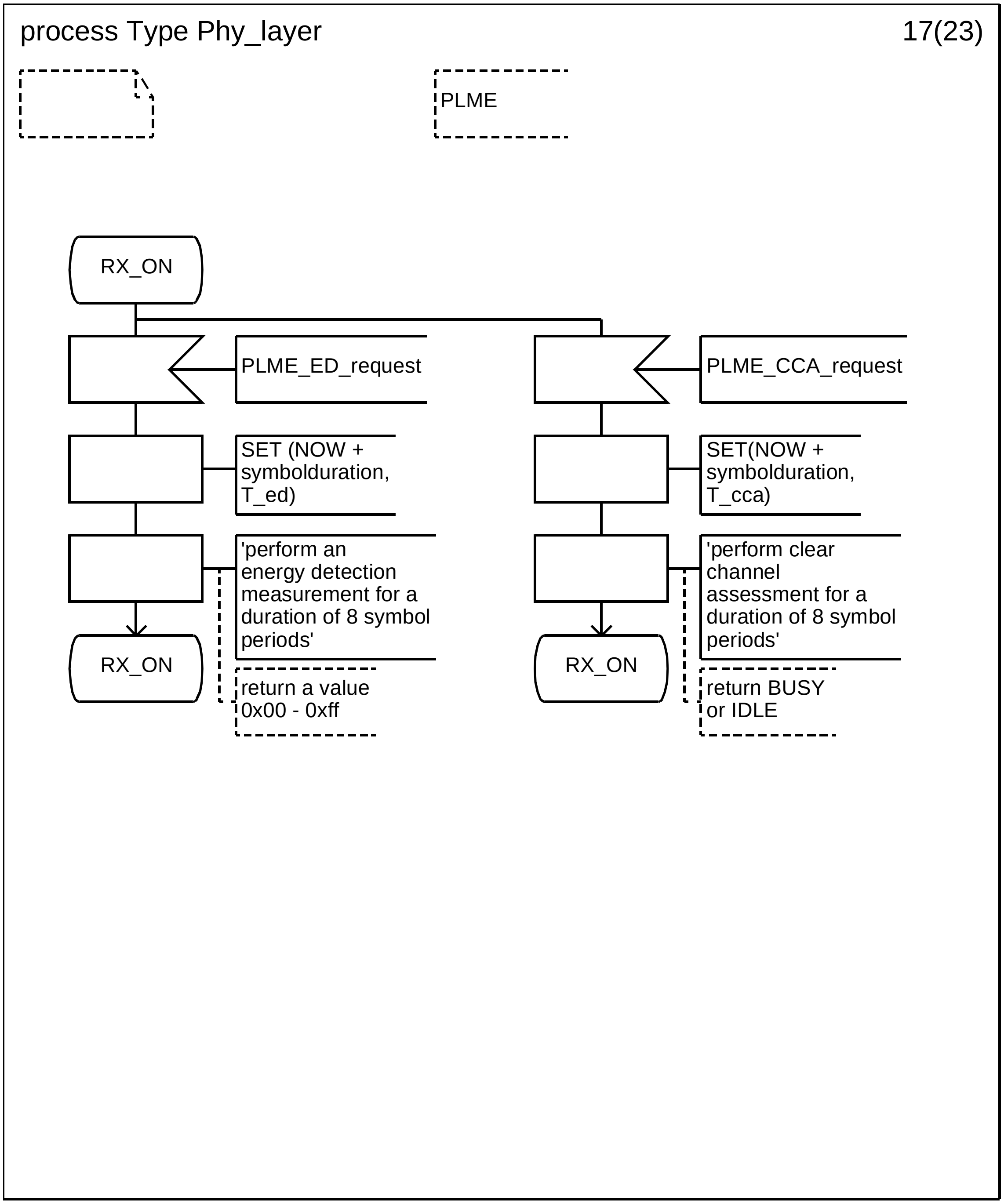
D.2.1.16 Process type Phy_Layer (16)

.



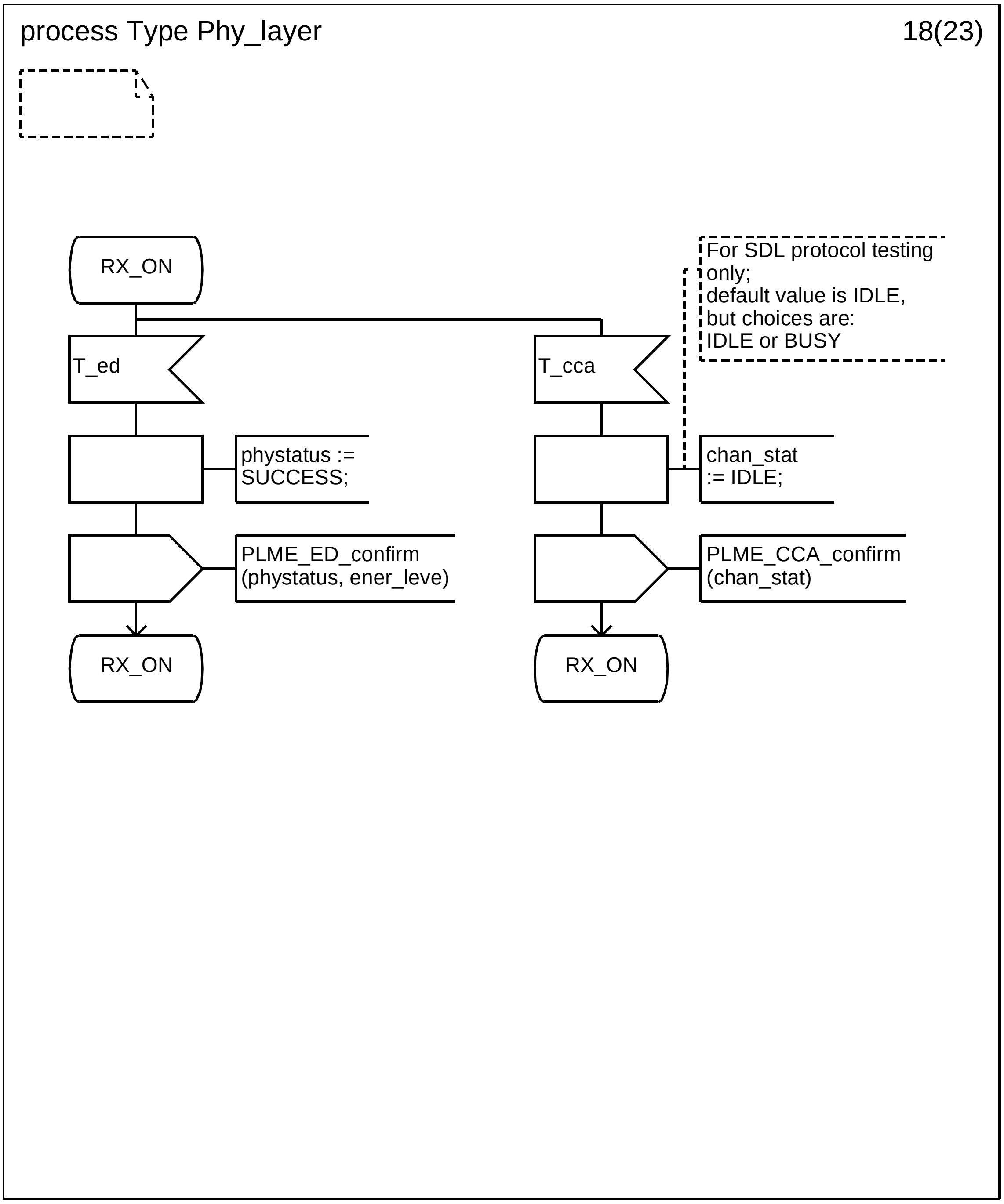
D.2.1.17 Process type Phy_Layer (17)

.



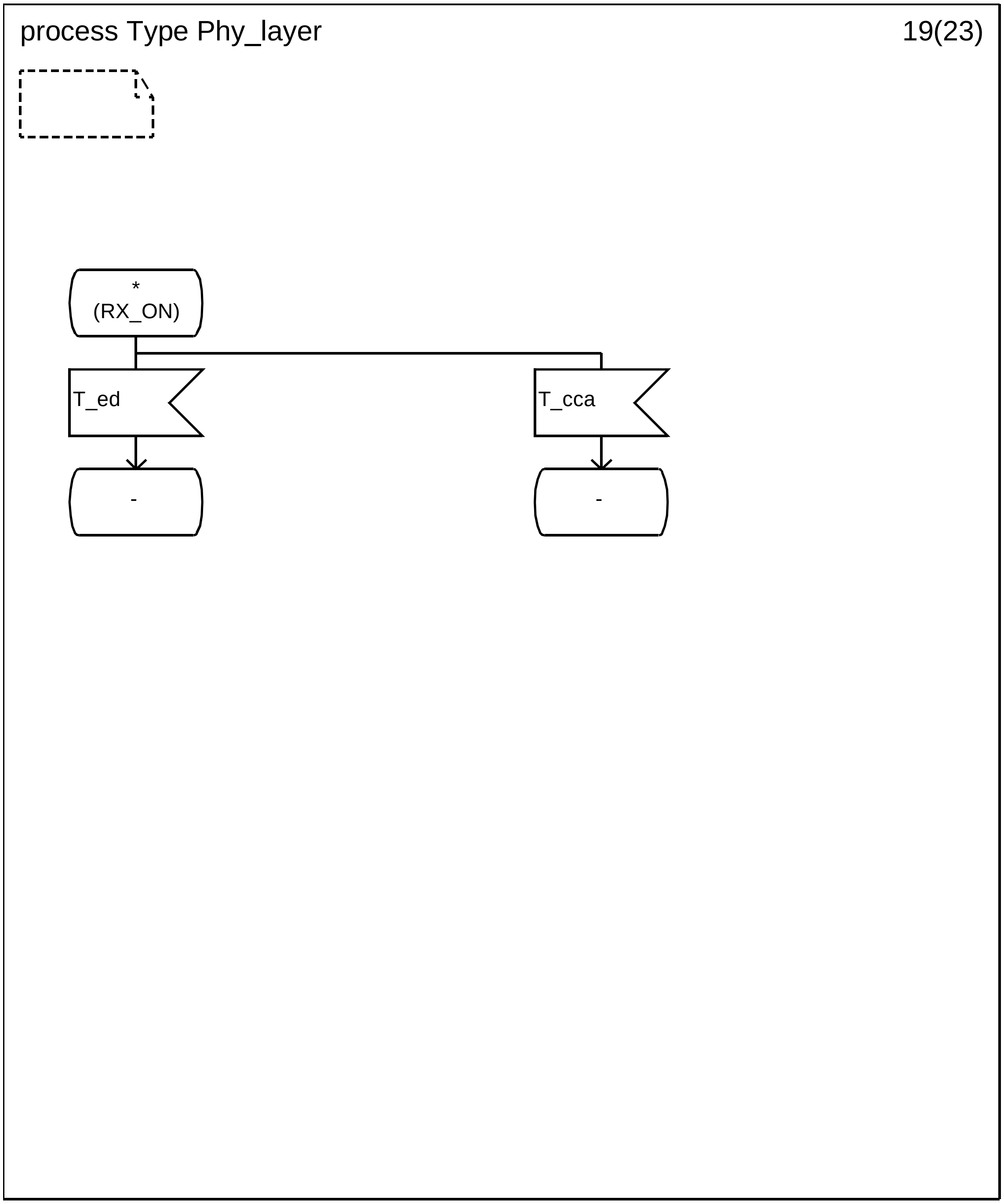
D.2.1.18 Process type Phy_Layer (18)

.



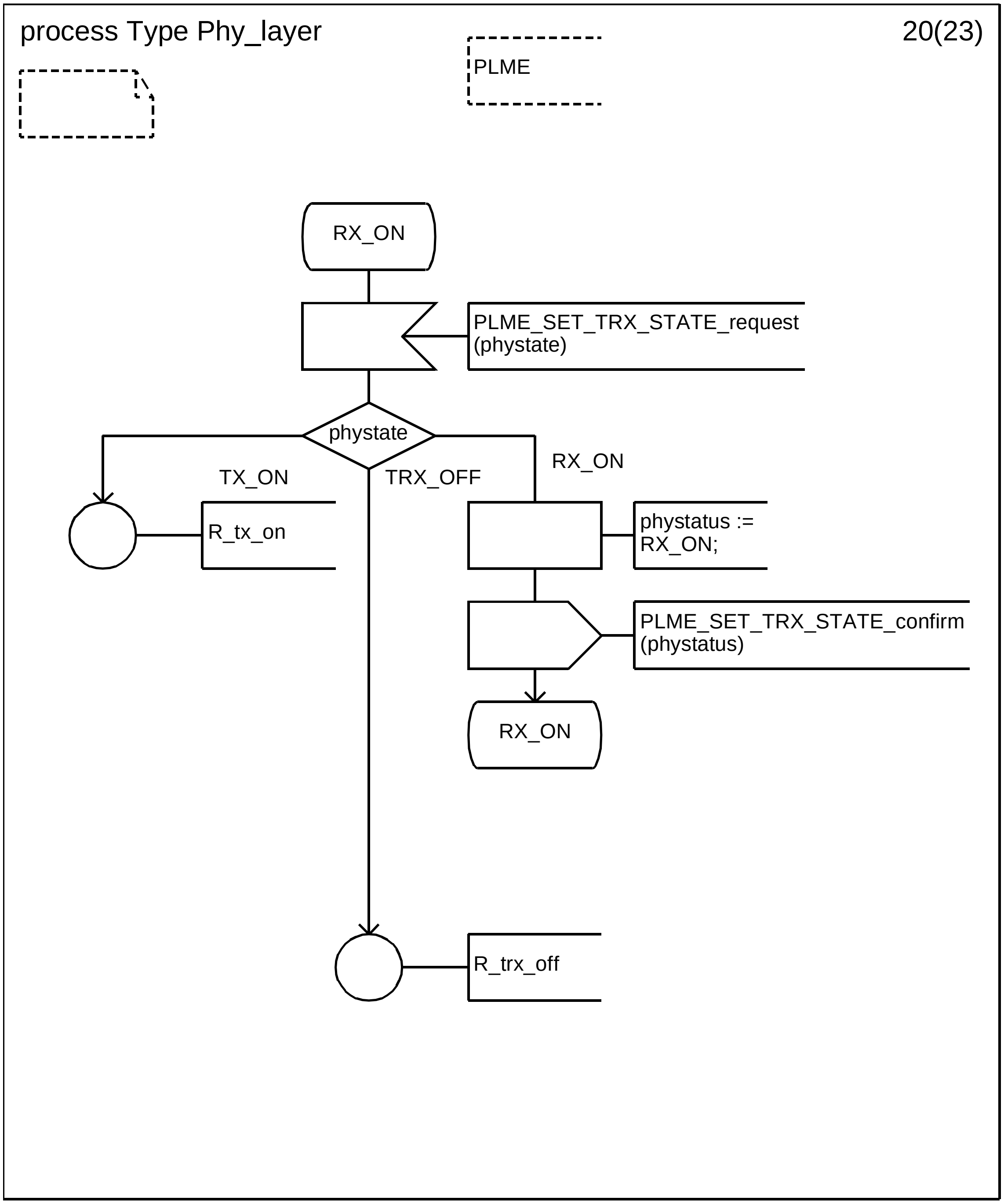
D.2.1.19 Process type Phy_Layer (19)

.



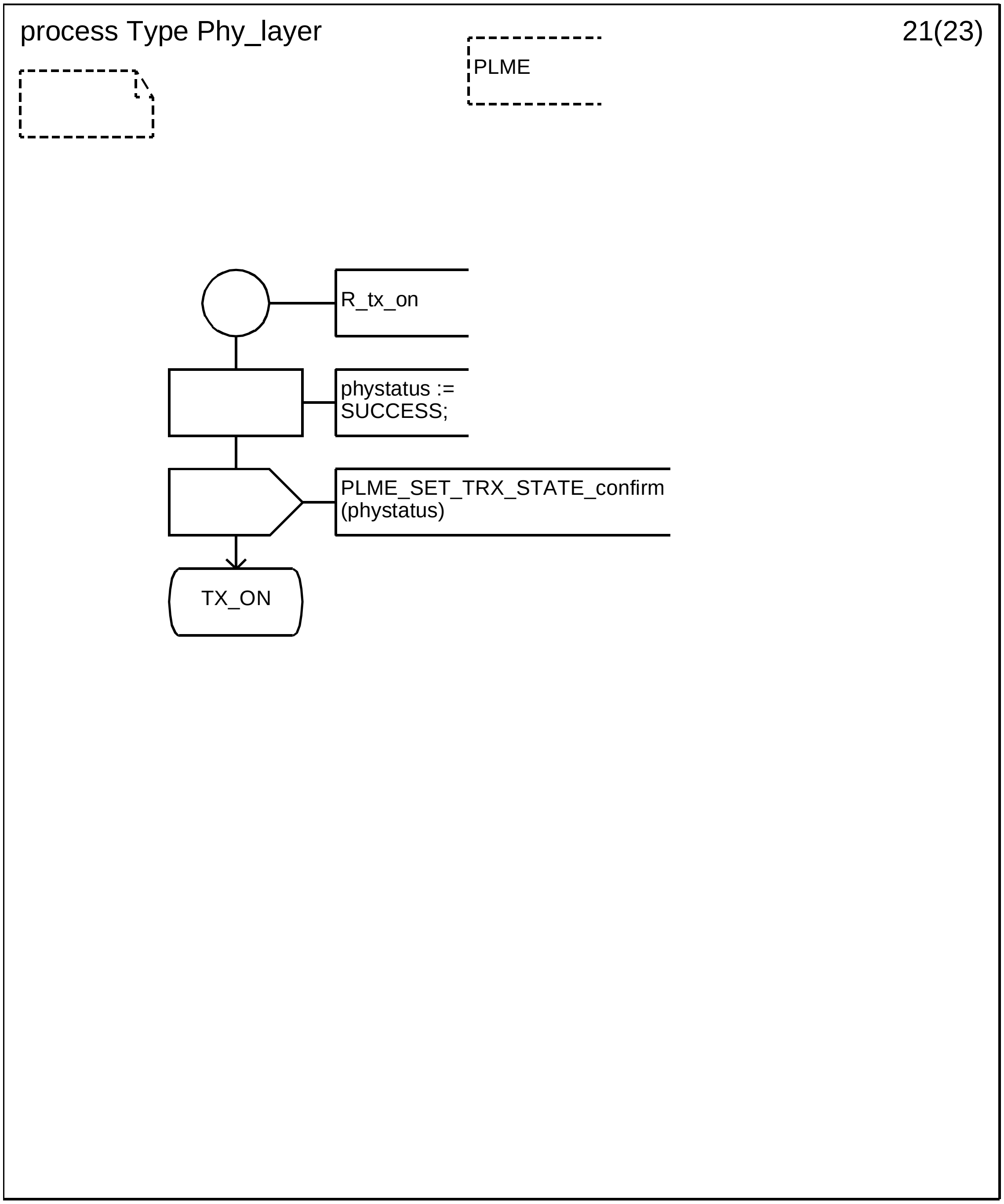
D.2.1.20 Process type Phy_Layer (20)

.



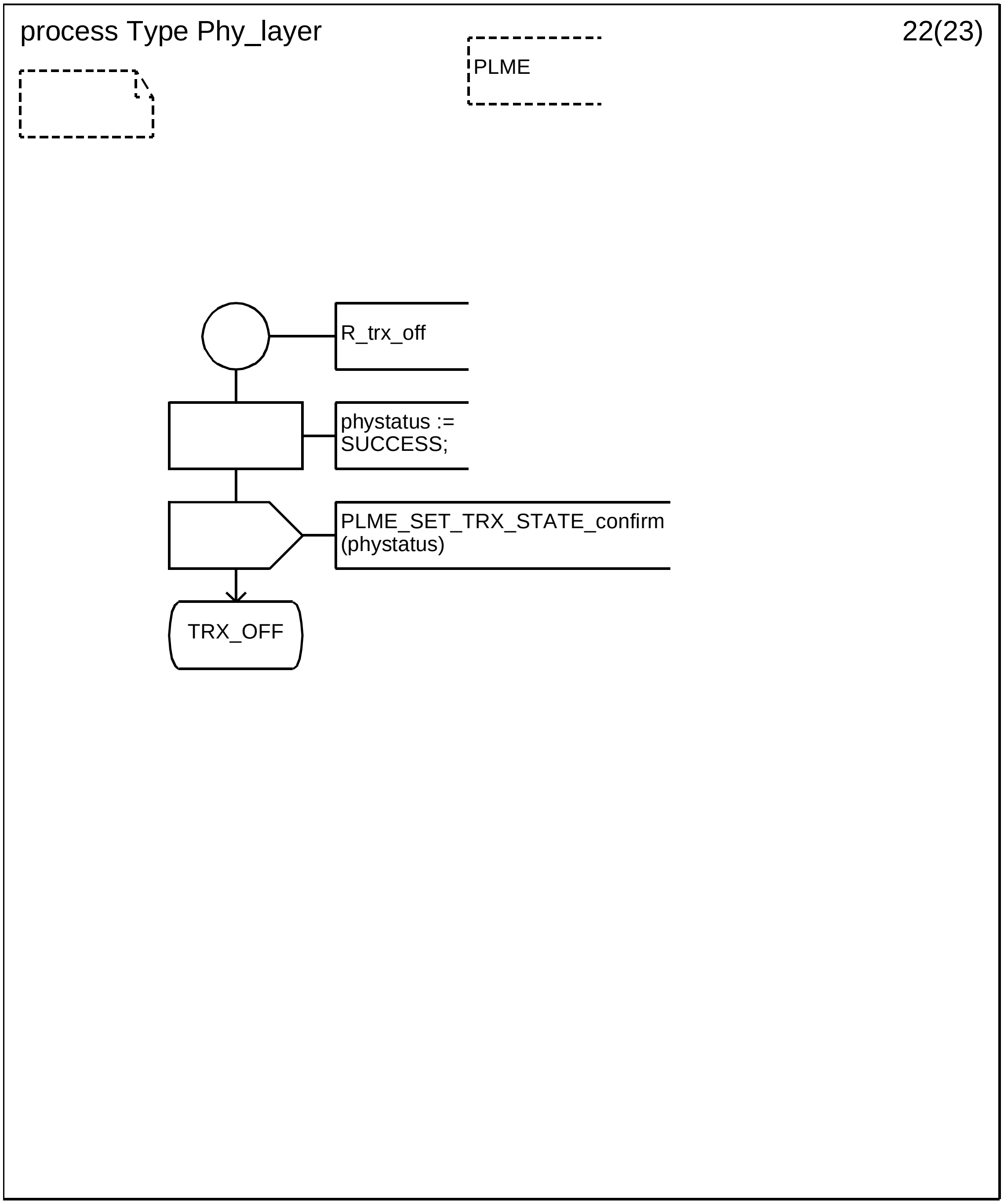
D.2.1.21 Process type Phy_Layer (21)

.



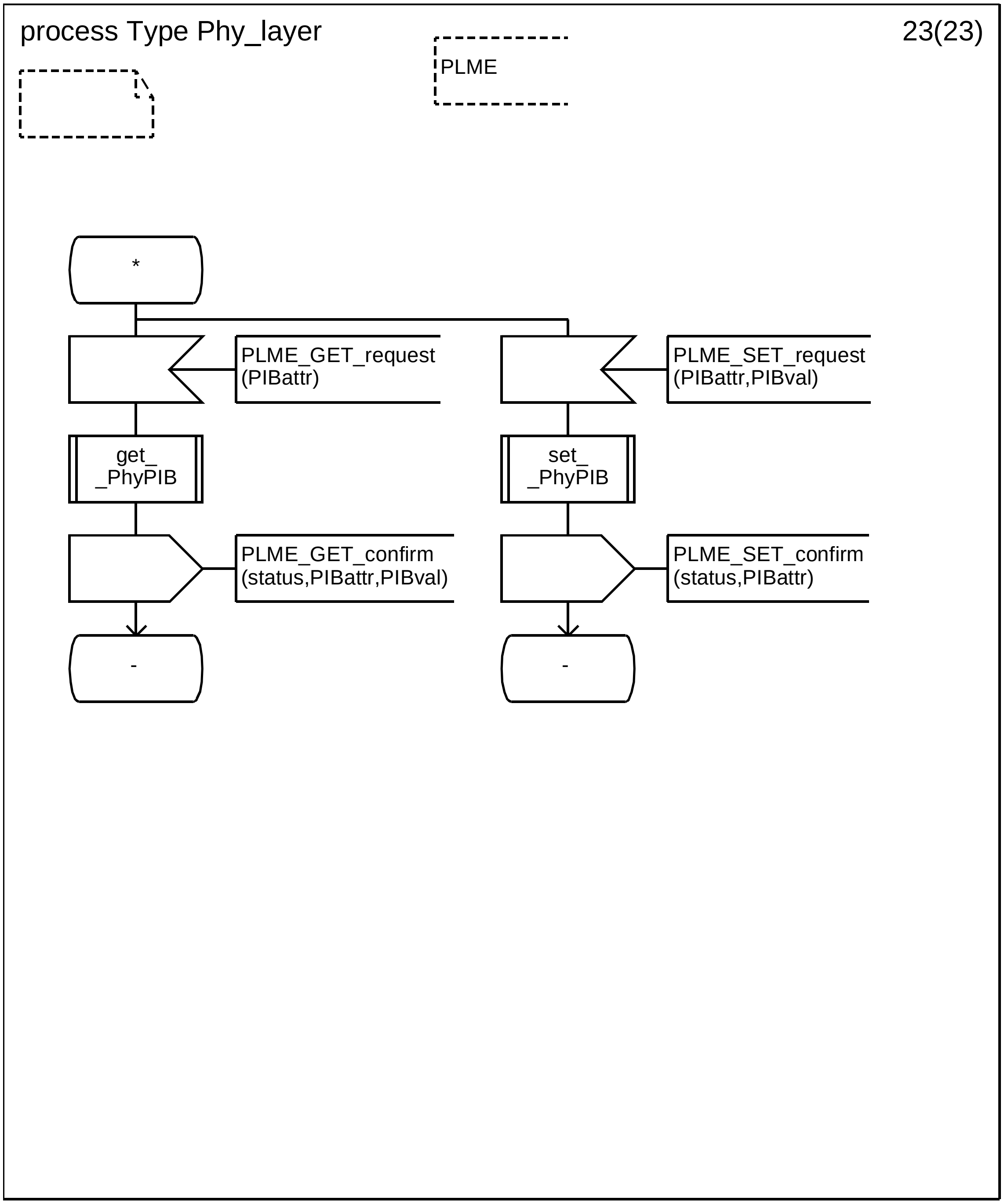
D.2.1.22 Process type Phy_Layer (22)

.



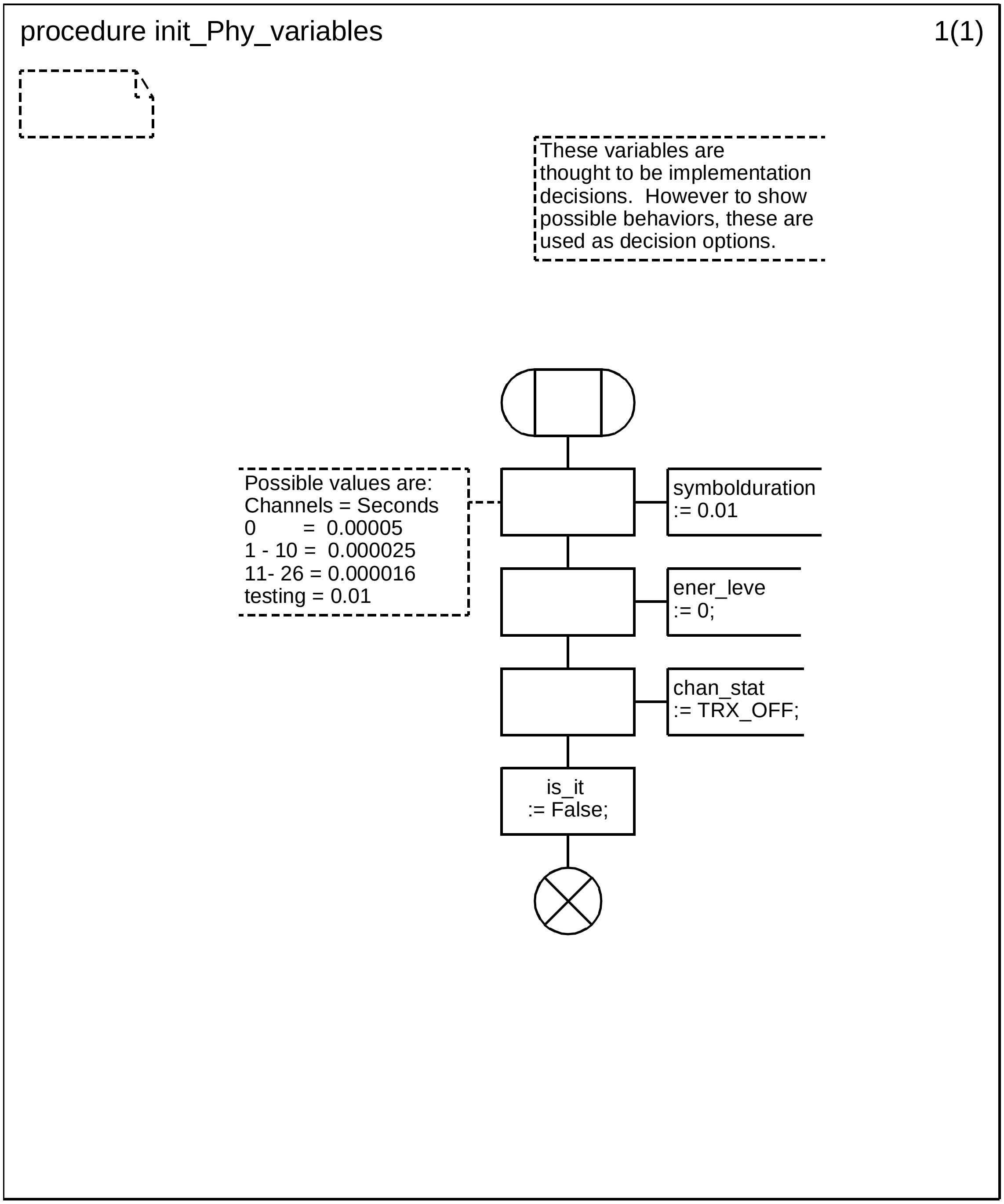
D.2.1.23 Process type Phy_Layer (23)

.



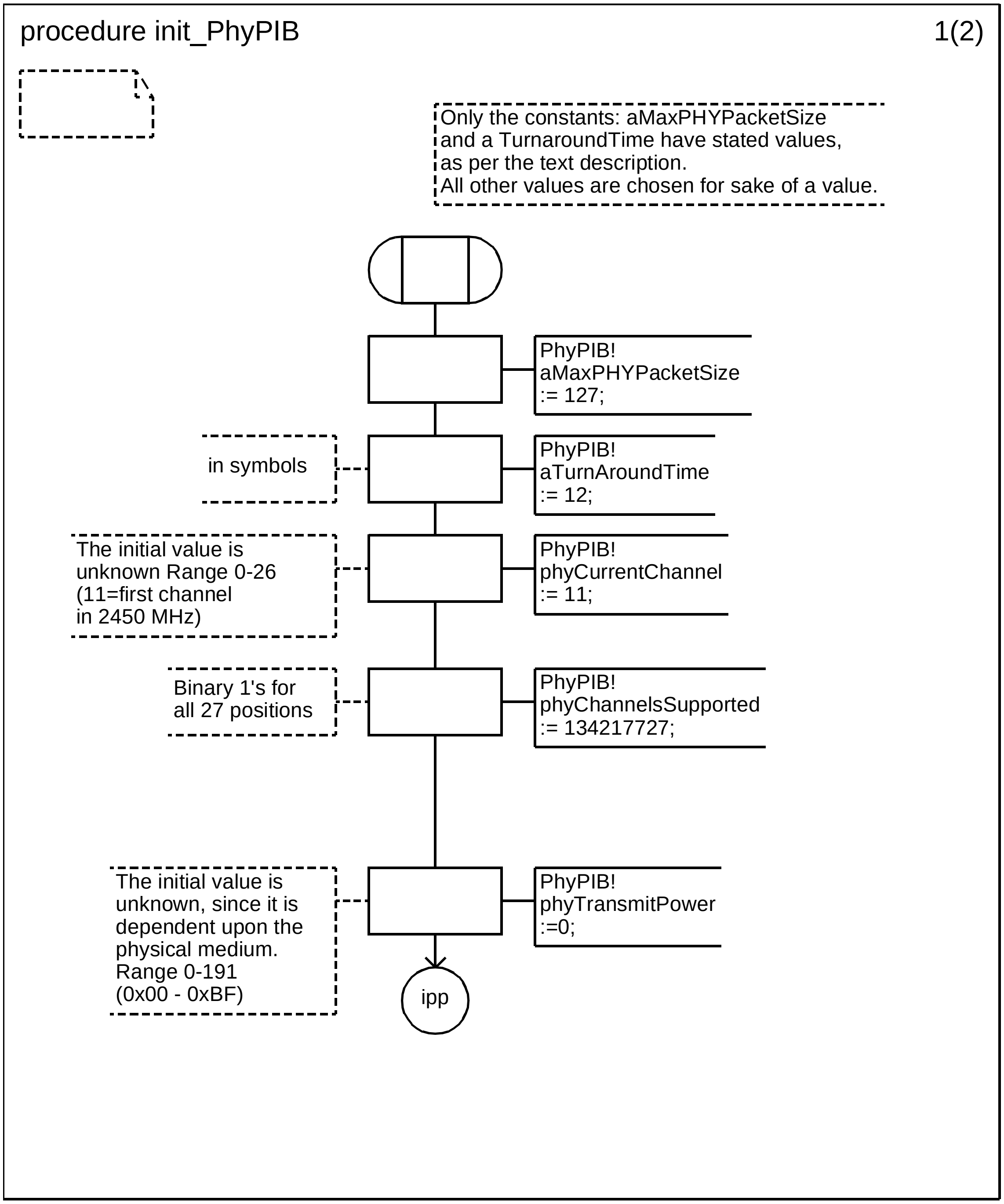
D.2.1.23.1 Procedure init_Phy_variables

.



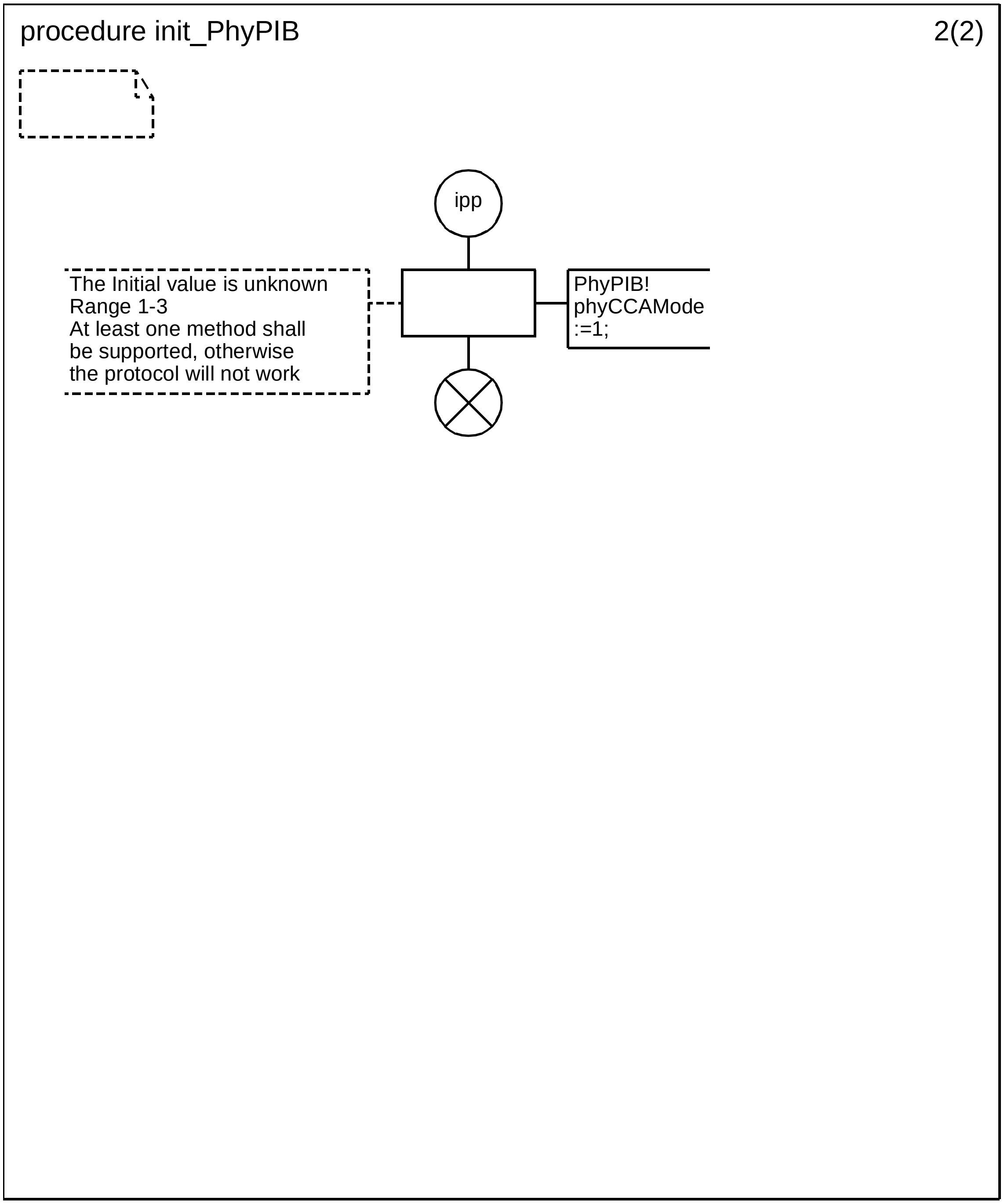
D.2.1.23.2 Procedure init_PhyPIB (1)

.



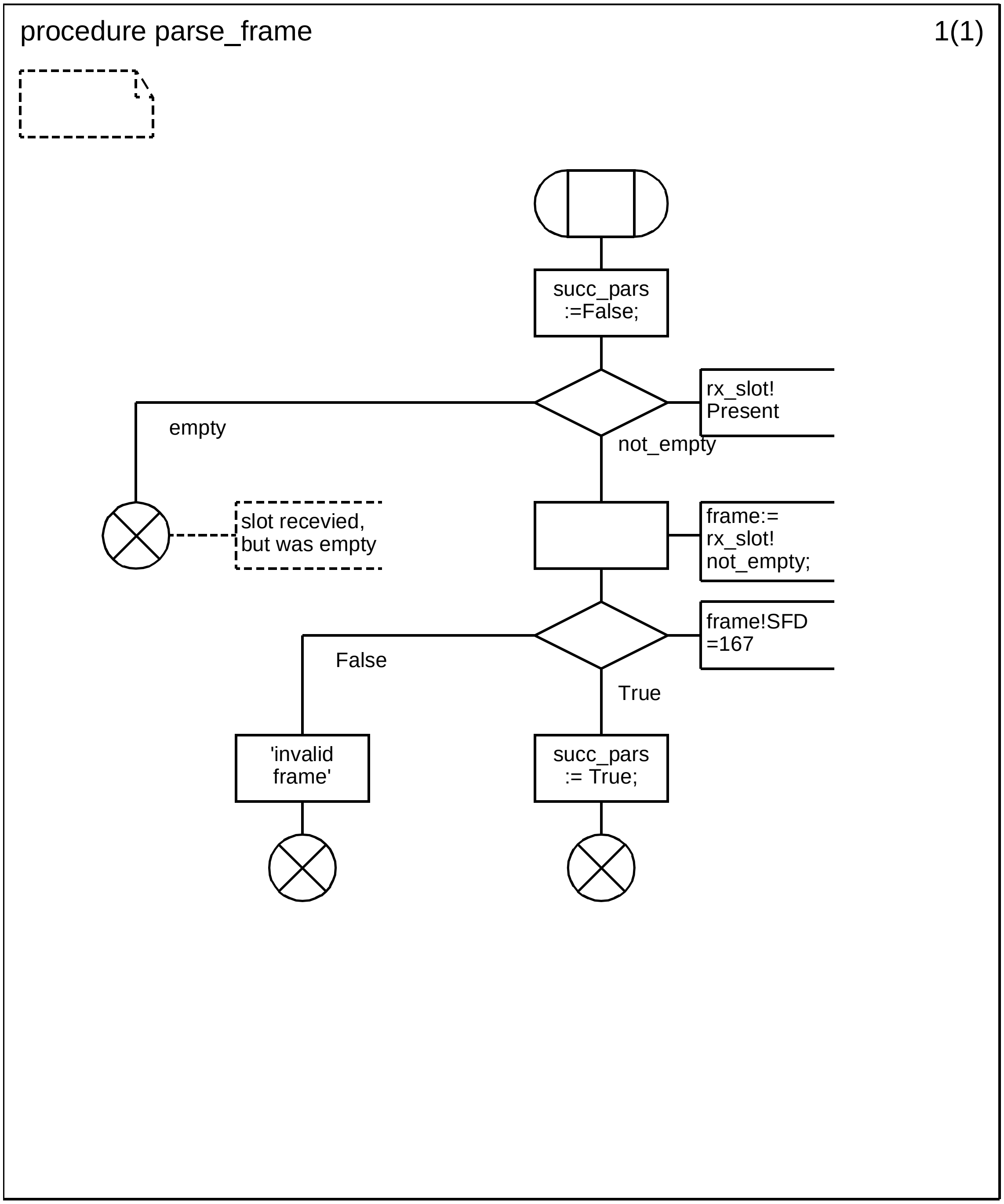
D.2.1.23.3 Procedure init_PhyPIB (2)

.



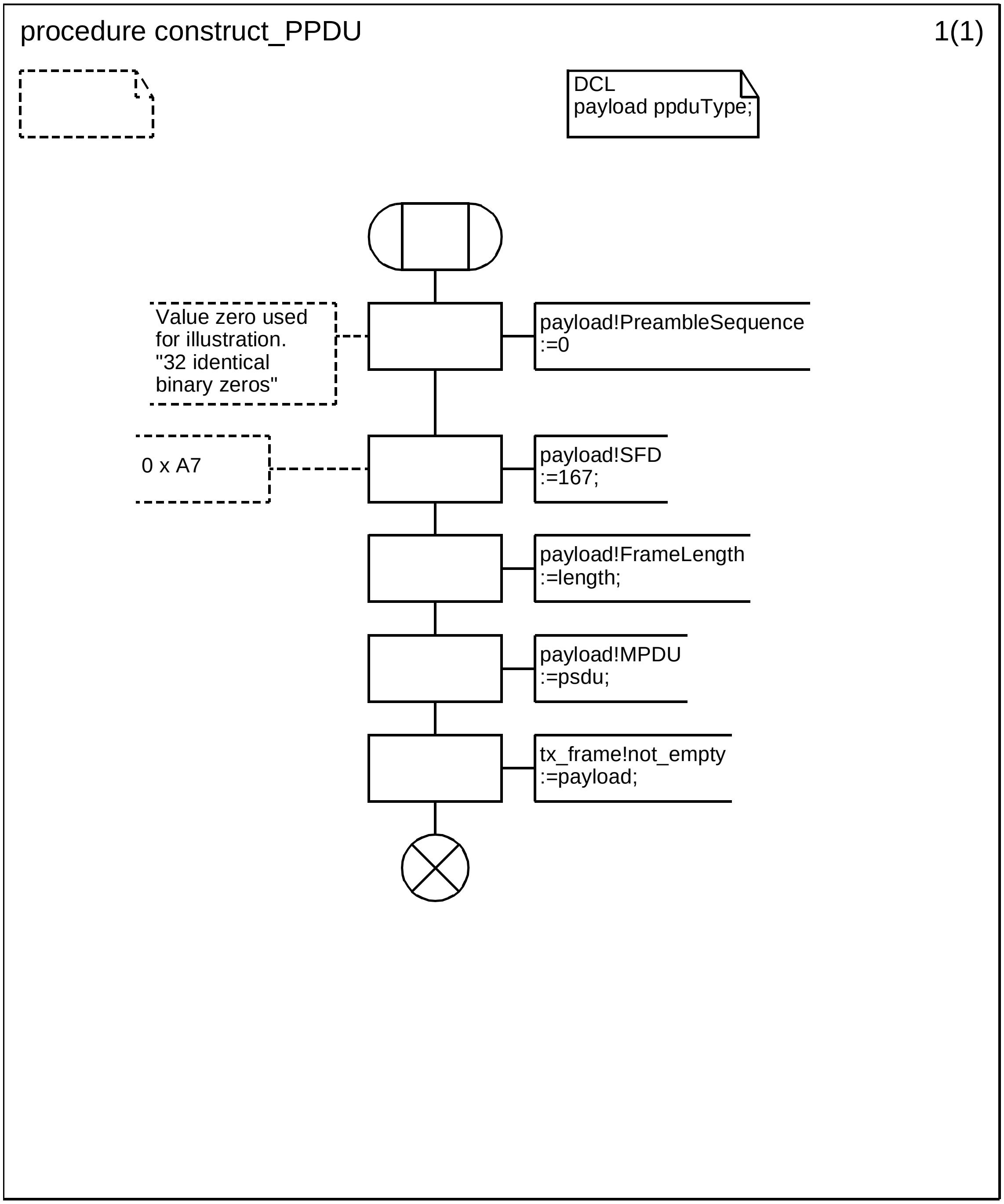
D.2.1.23.4 Procedure parse_frame

.



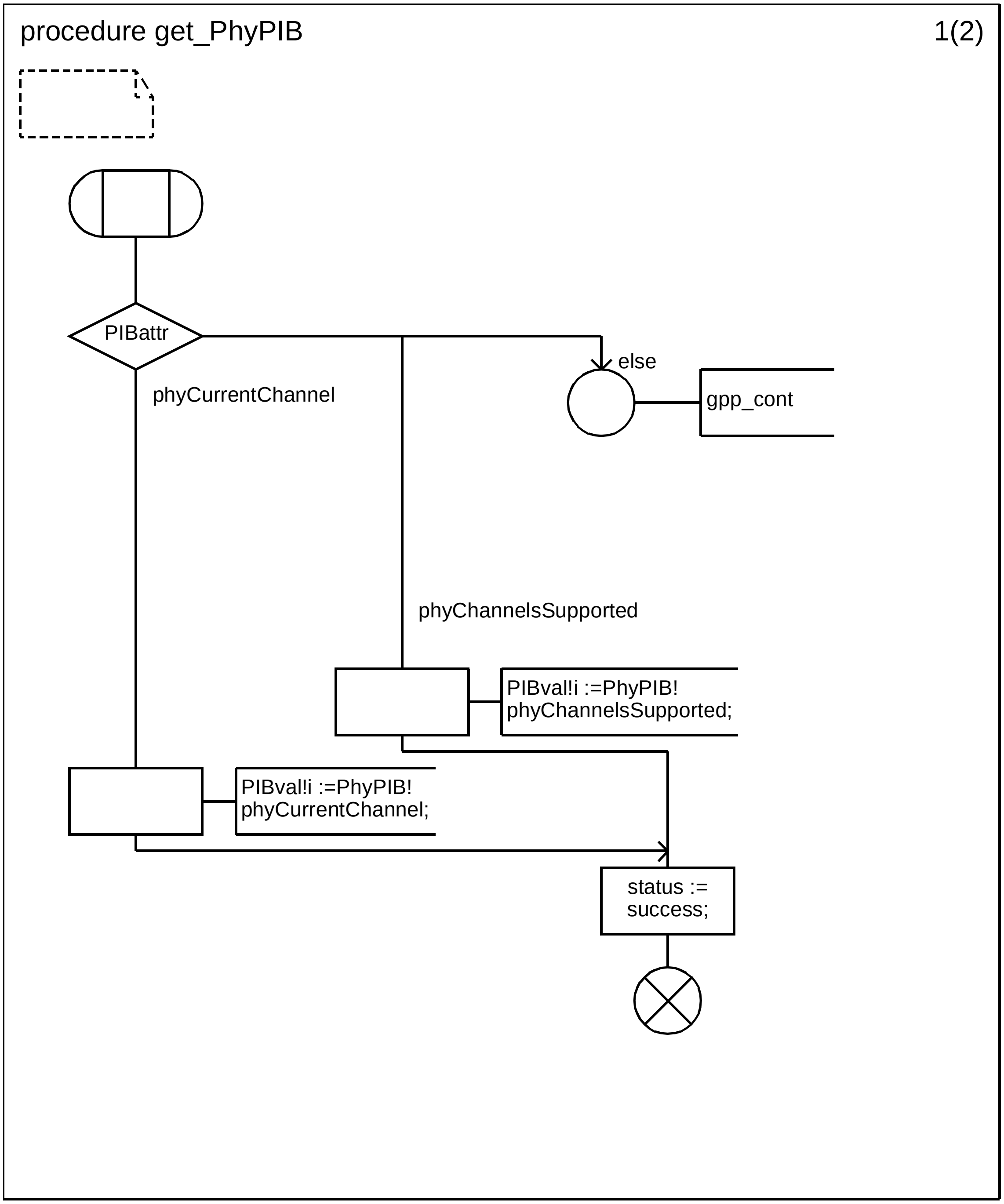
D.2.1.23.5 Procedure construct_PPDU

.



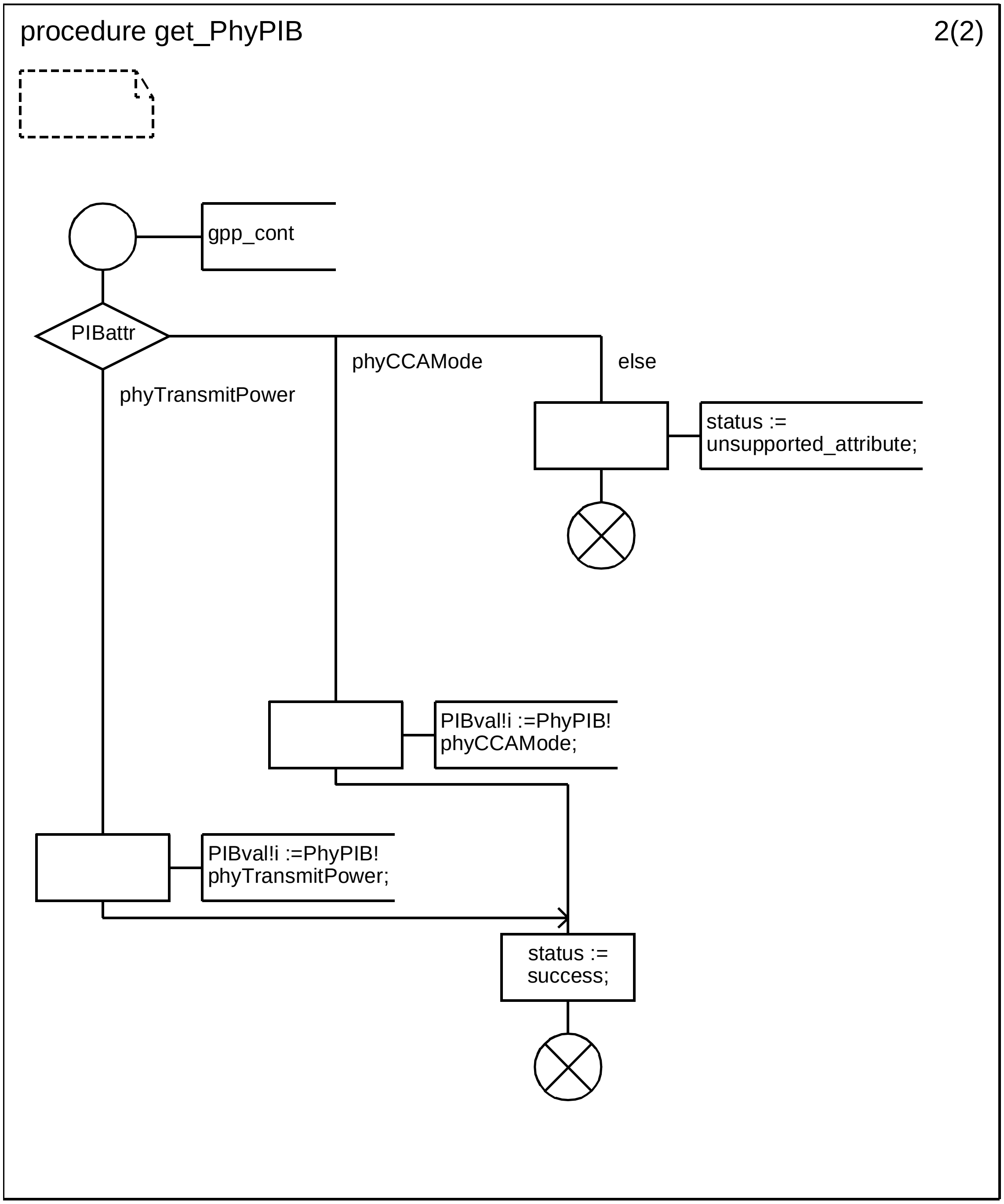
D.2.1.23.6 Procedure get_PhyPIB (1)

.



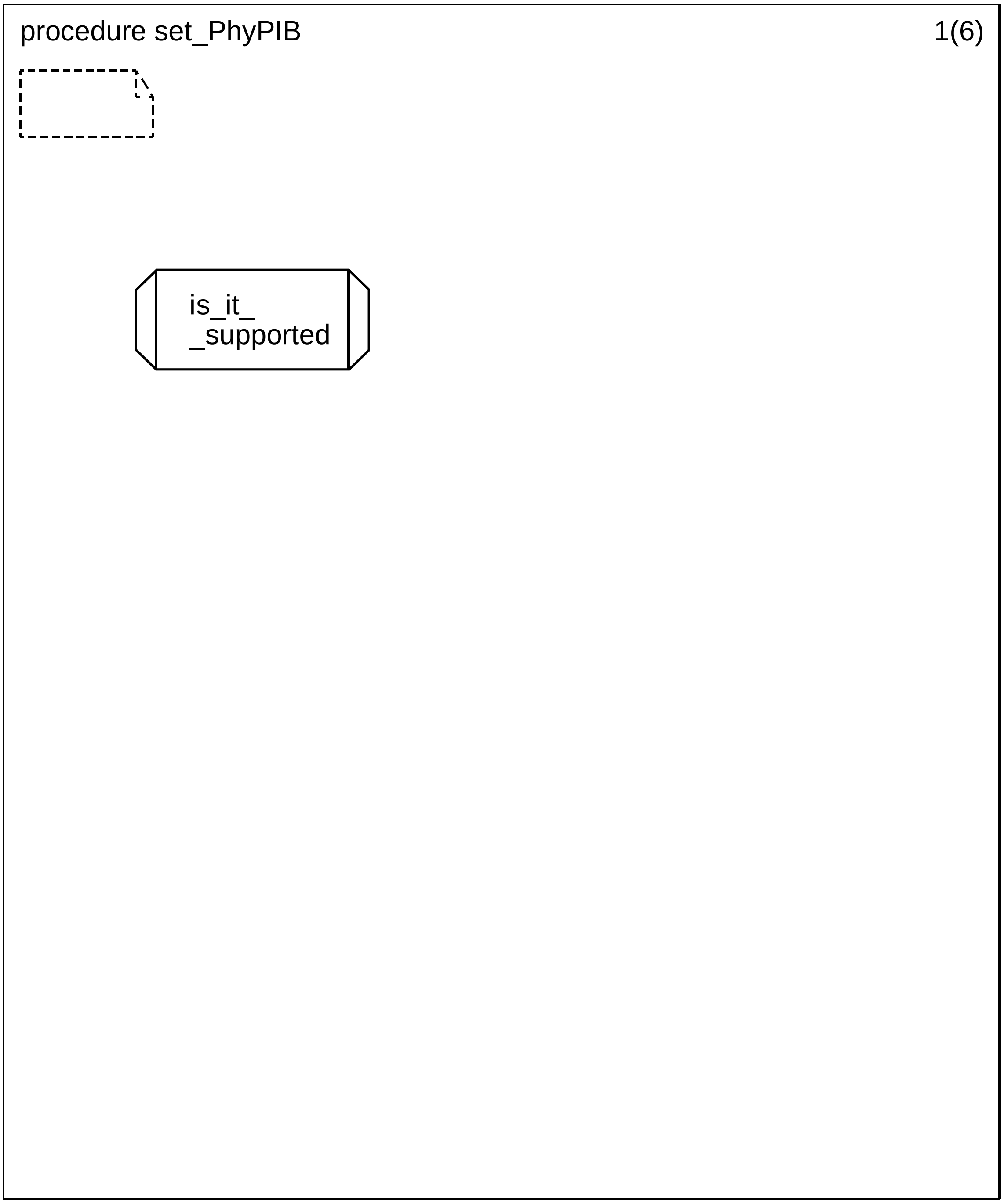
D.2.1.23.7 Procedure get_PhyPIB (2)

.



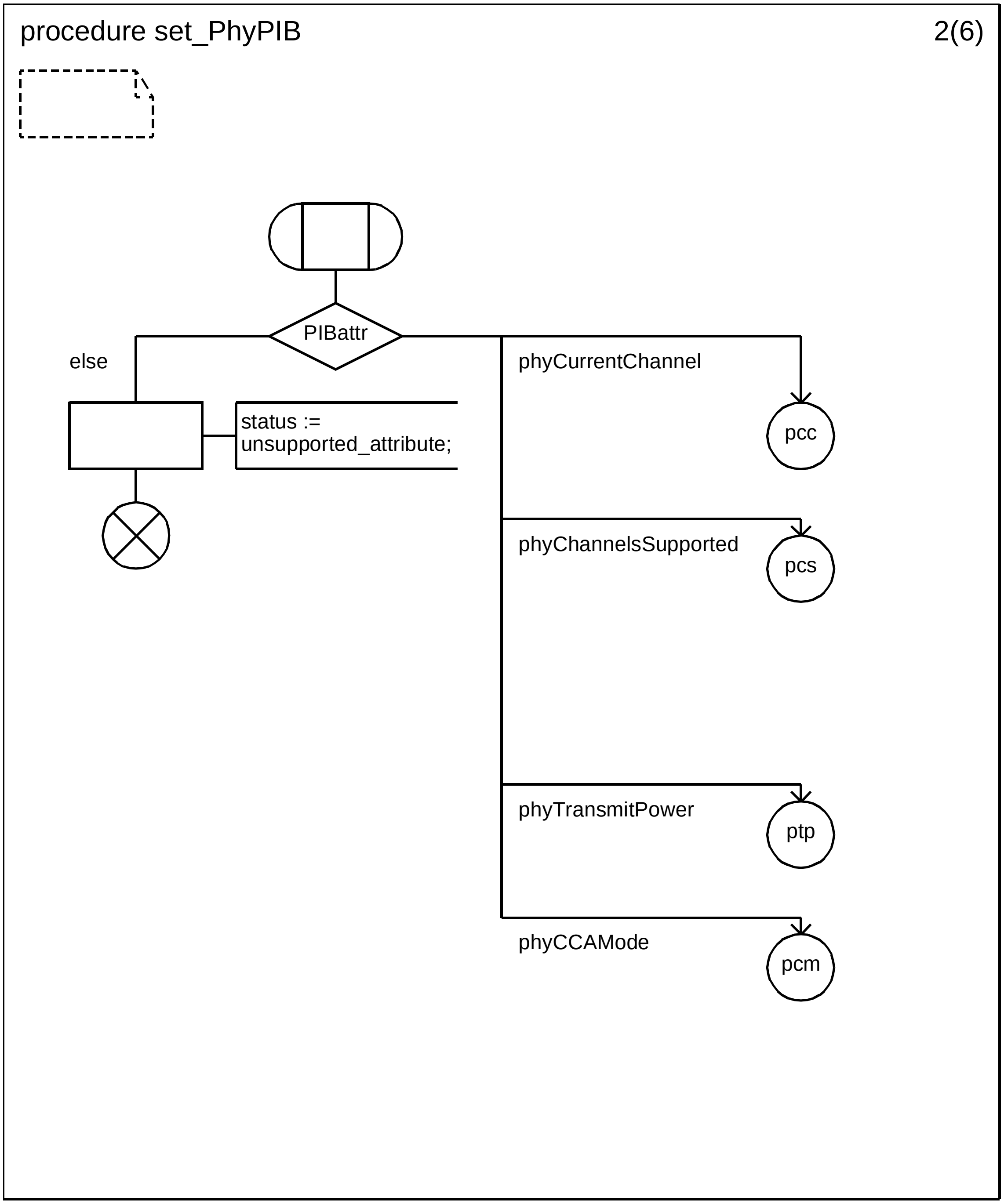
D.2.1.23.8 Procedure set_PhyPIB (1)

.



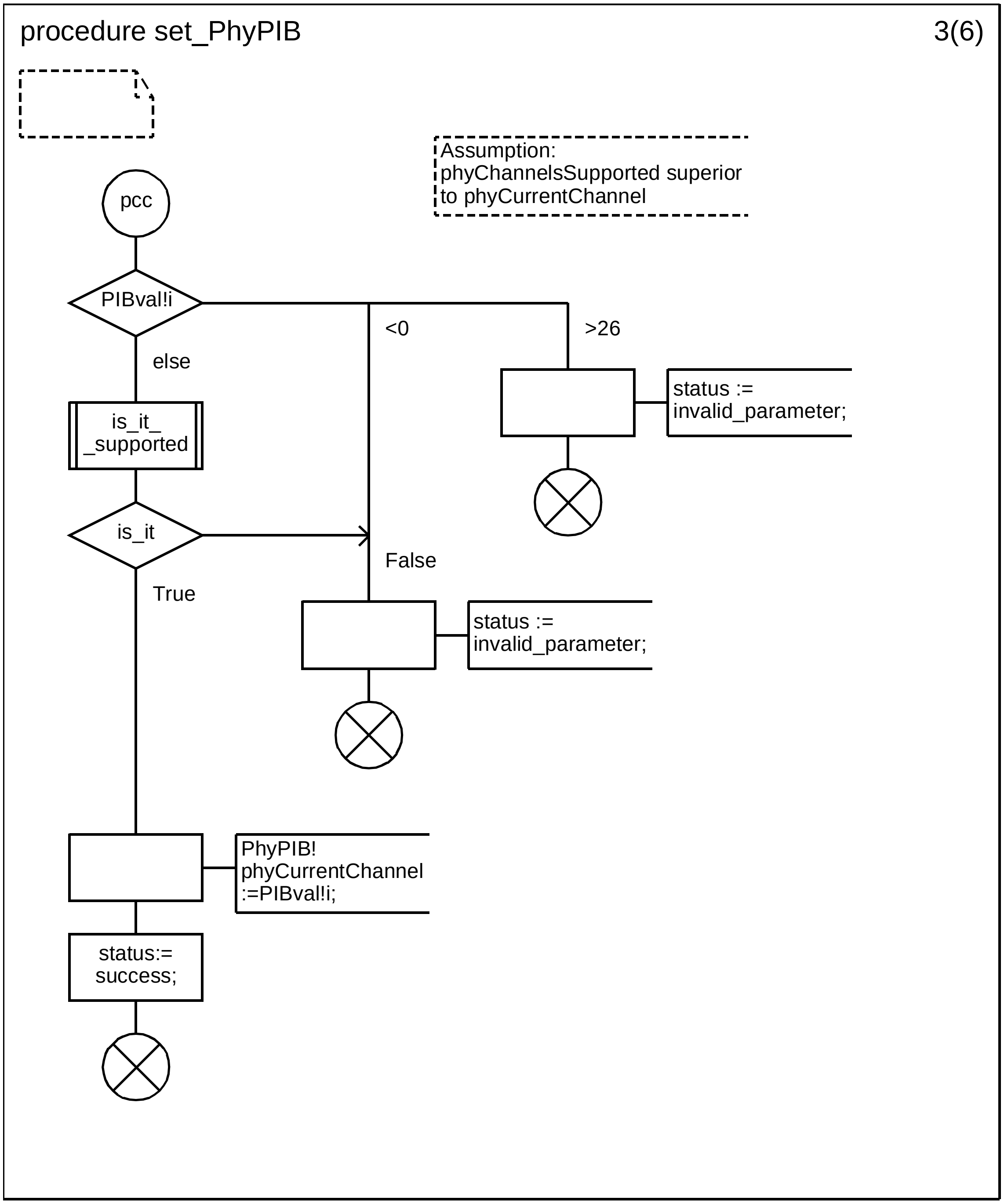
D.2.1.23.9 Procedure set_PhyPIB (2)

.



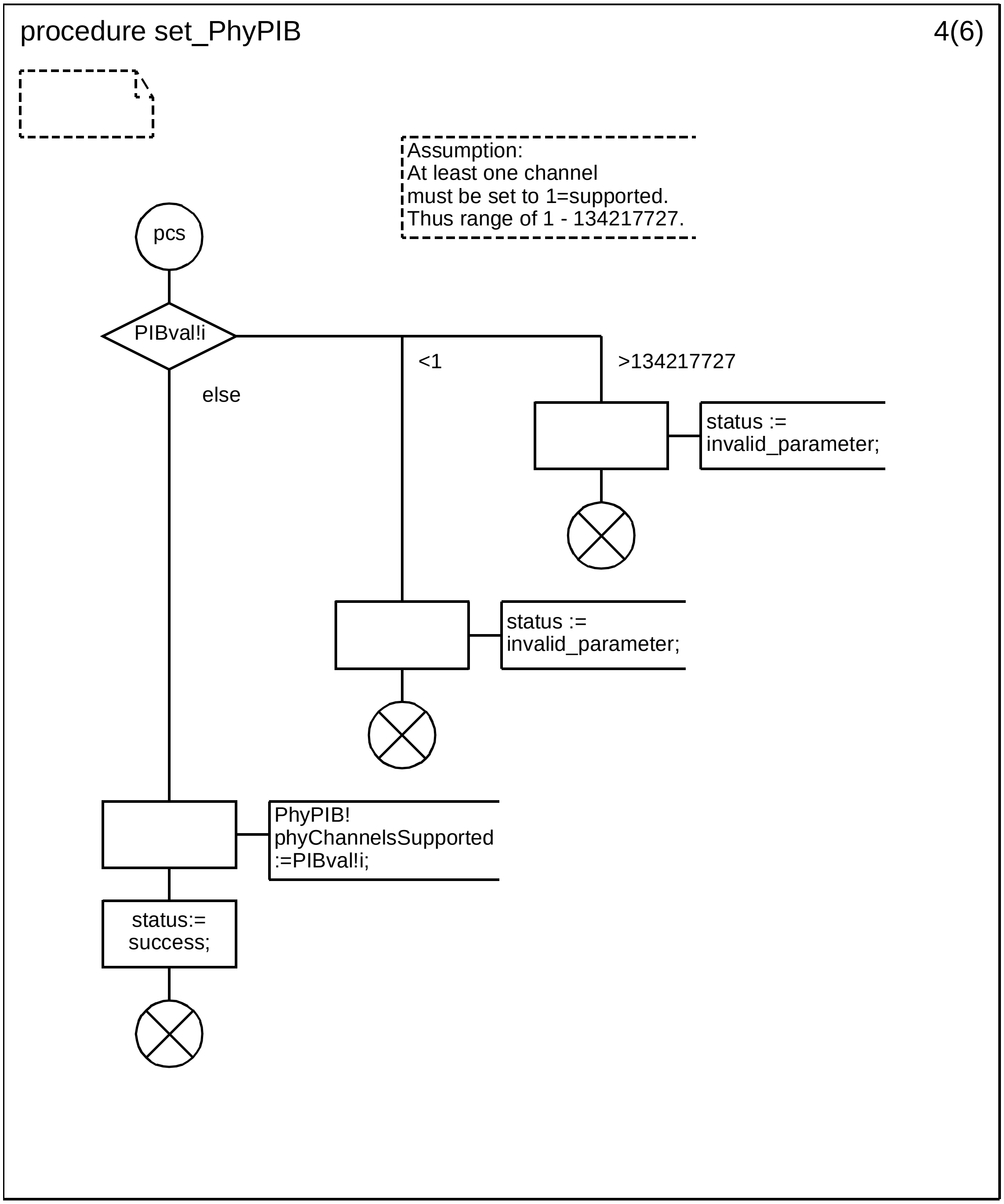
D.2.1.23.10 Procedure set_PhyPIB (3)

.



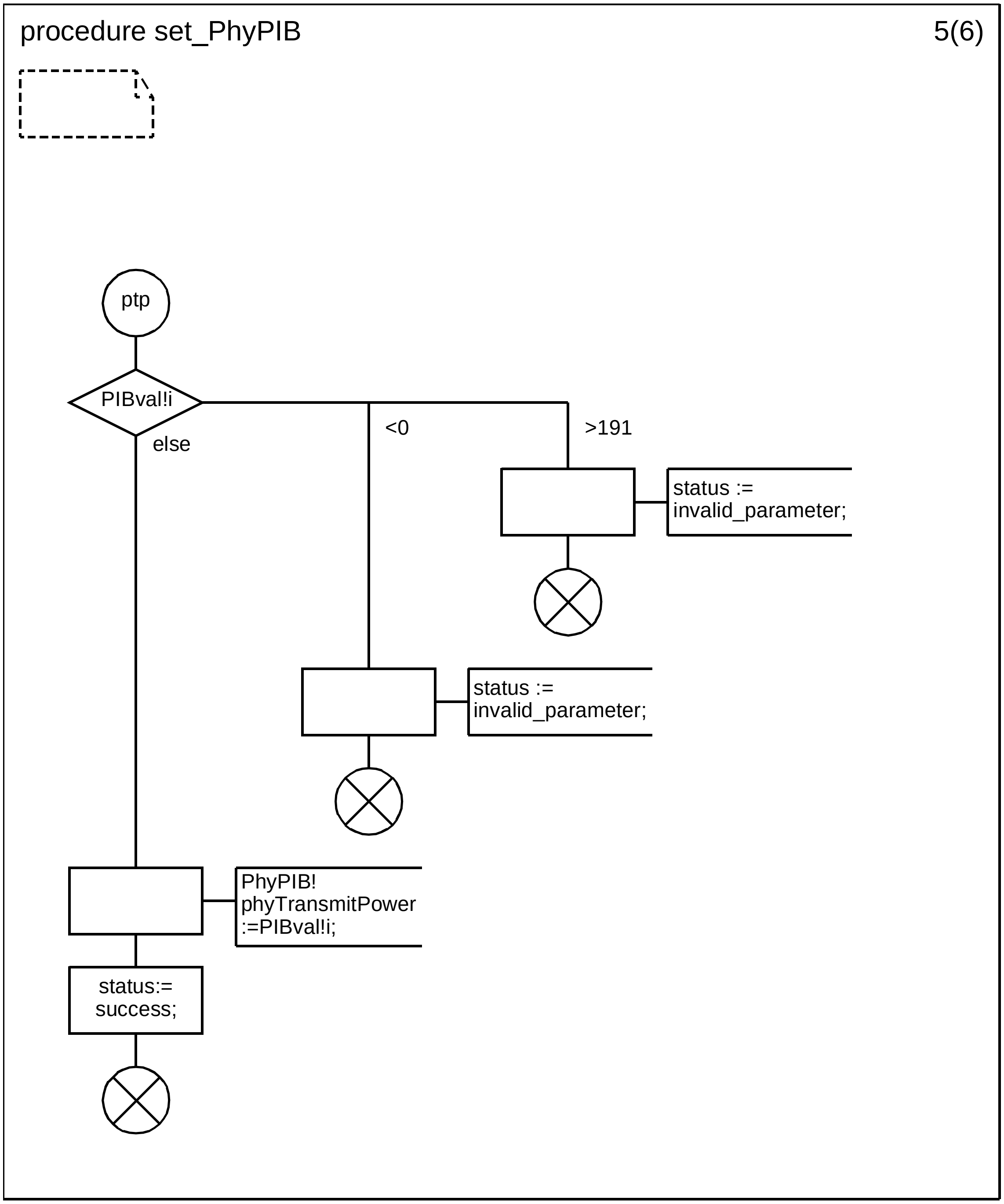
D.2.1.23.11 Procedure set_PhyPIB (4)

.



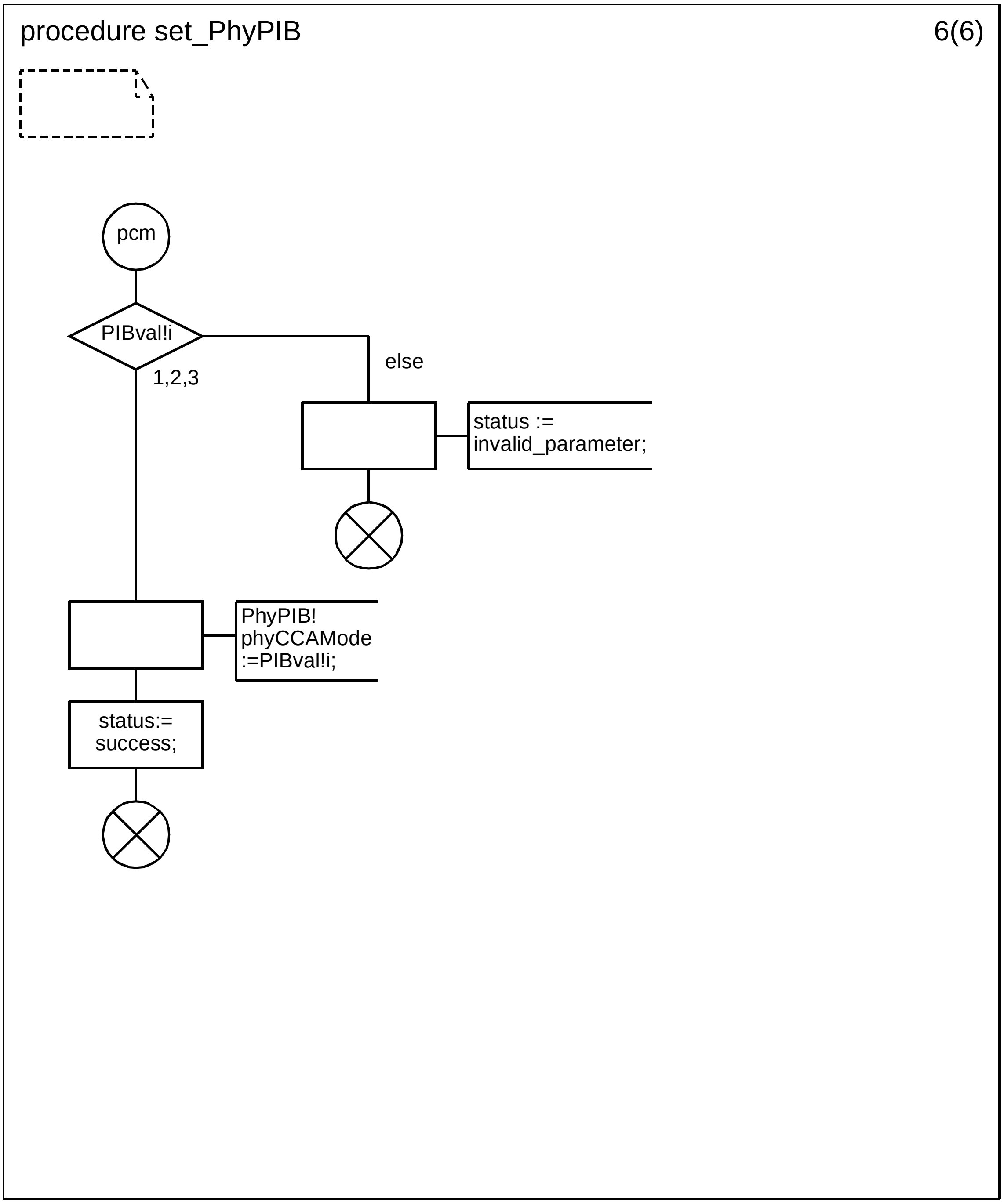
D.2.1.23.12 Procedure set_PhyPIB (5)

.



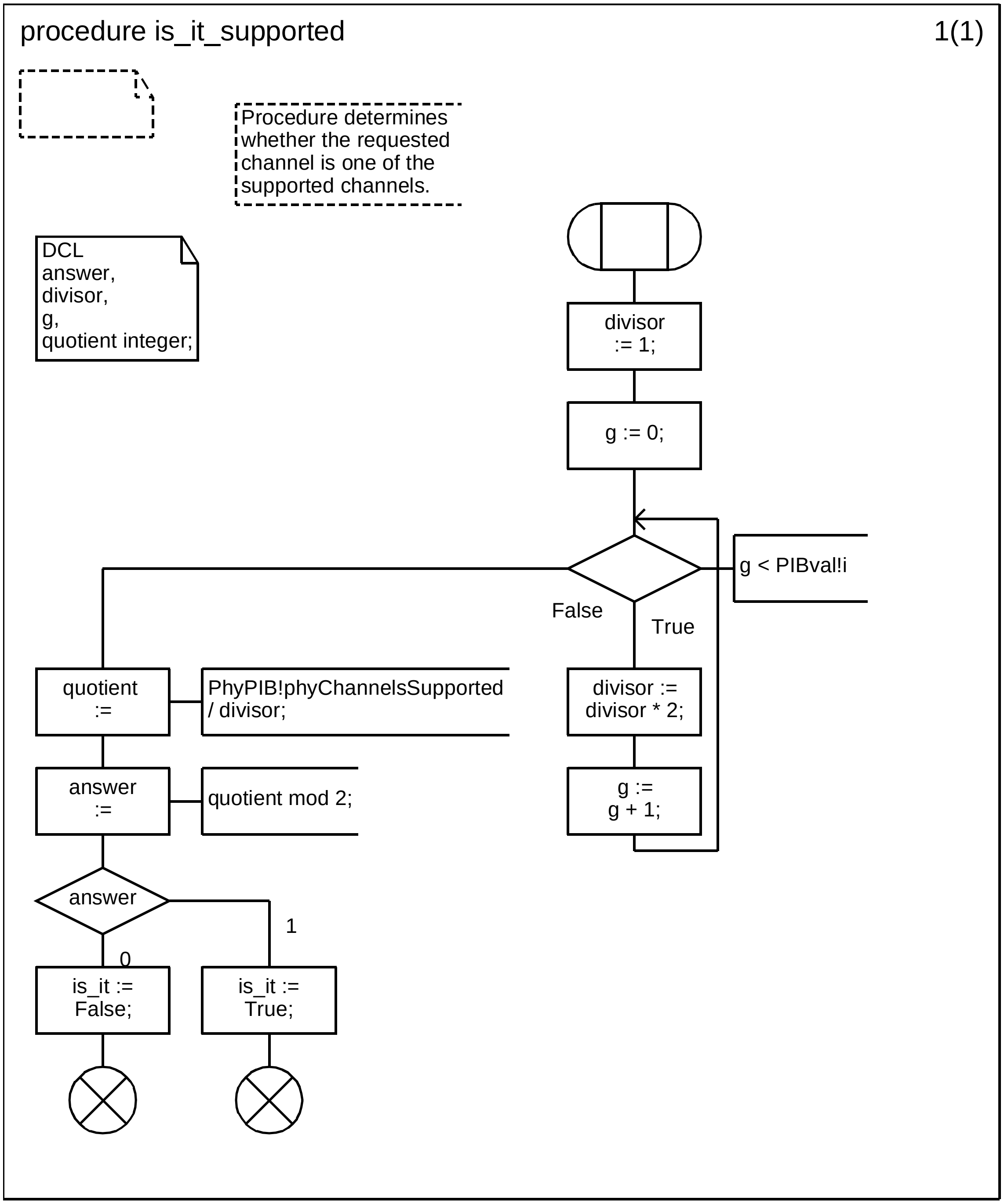
D.2.1.23.13 Procedure set_PhyPIB (6)

.



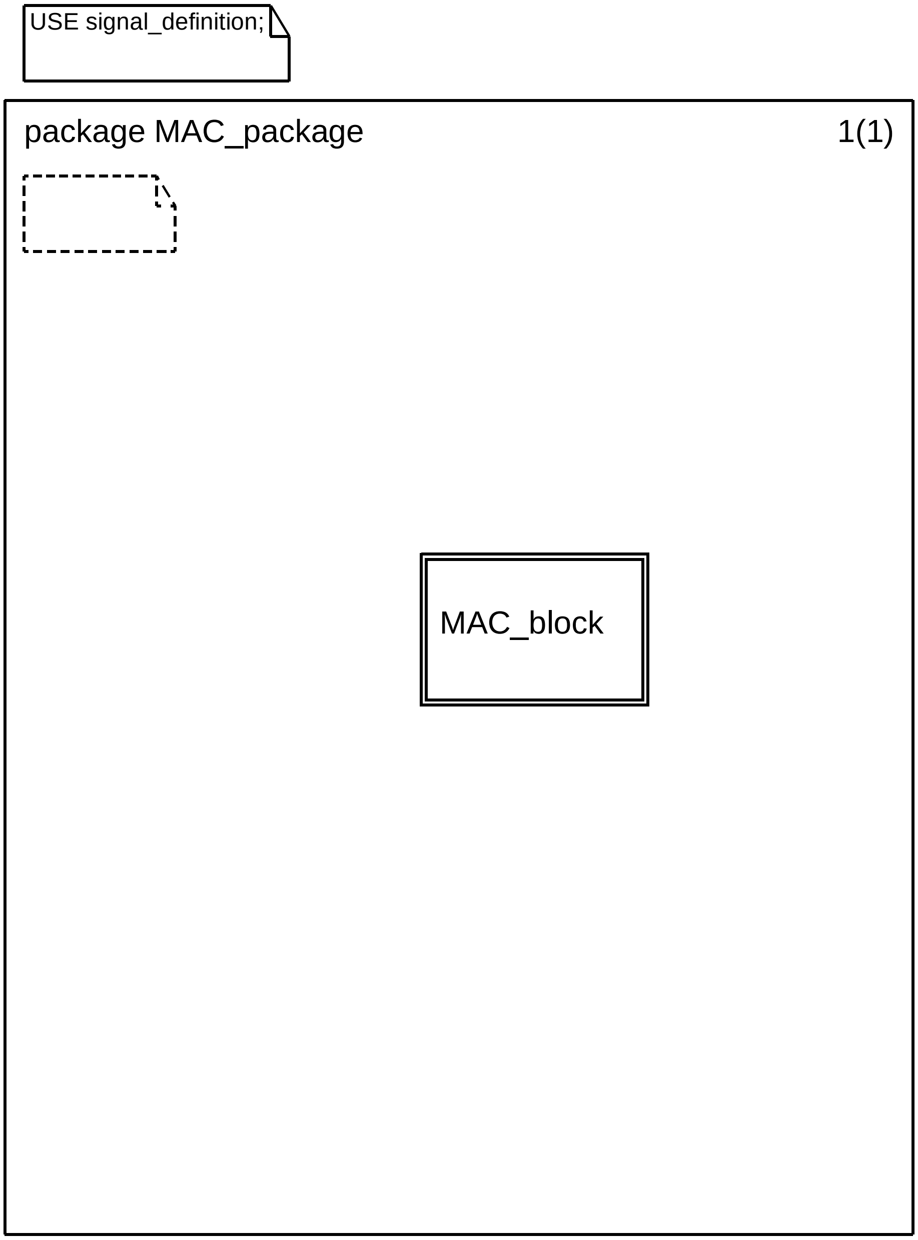
D.2.1.23.13.1 Procedure is_it_supported

.

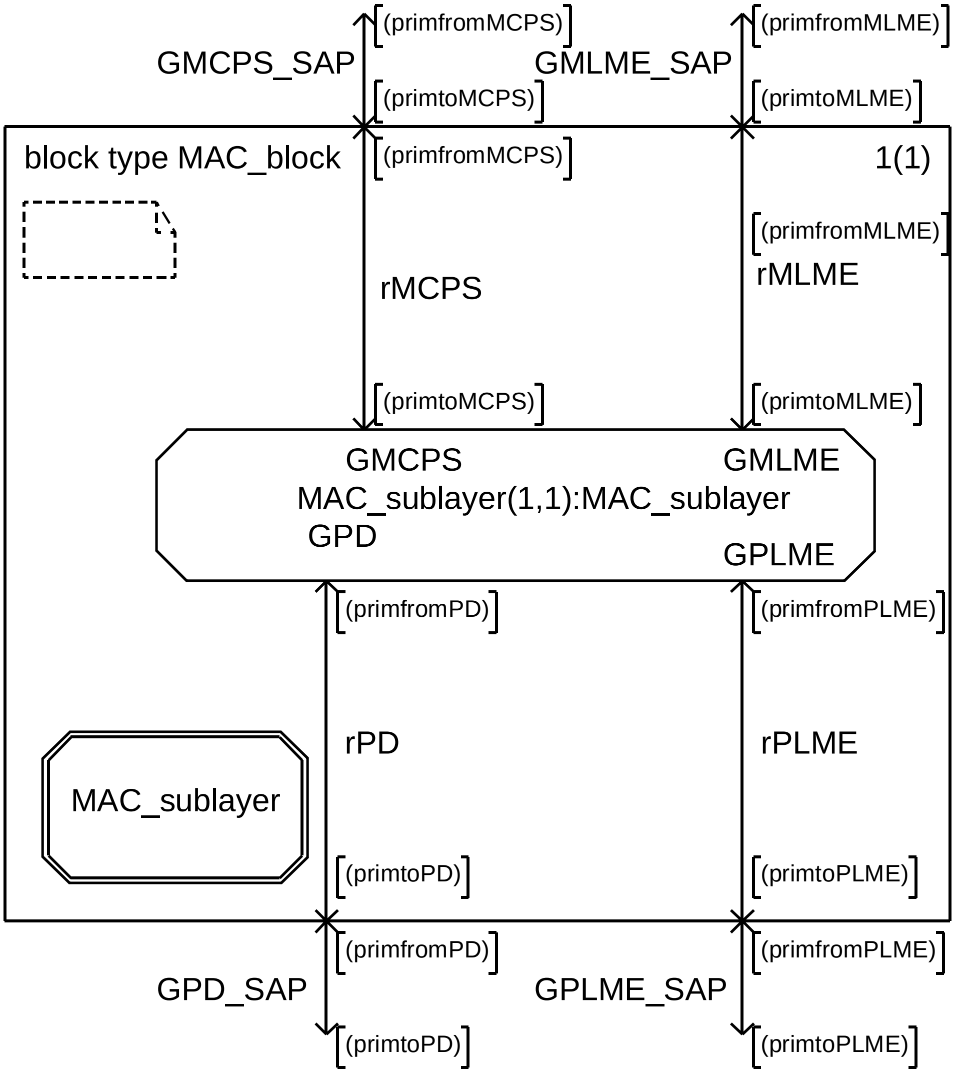


D.3 IEEE 802.15.4 MAC sublayer package

.

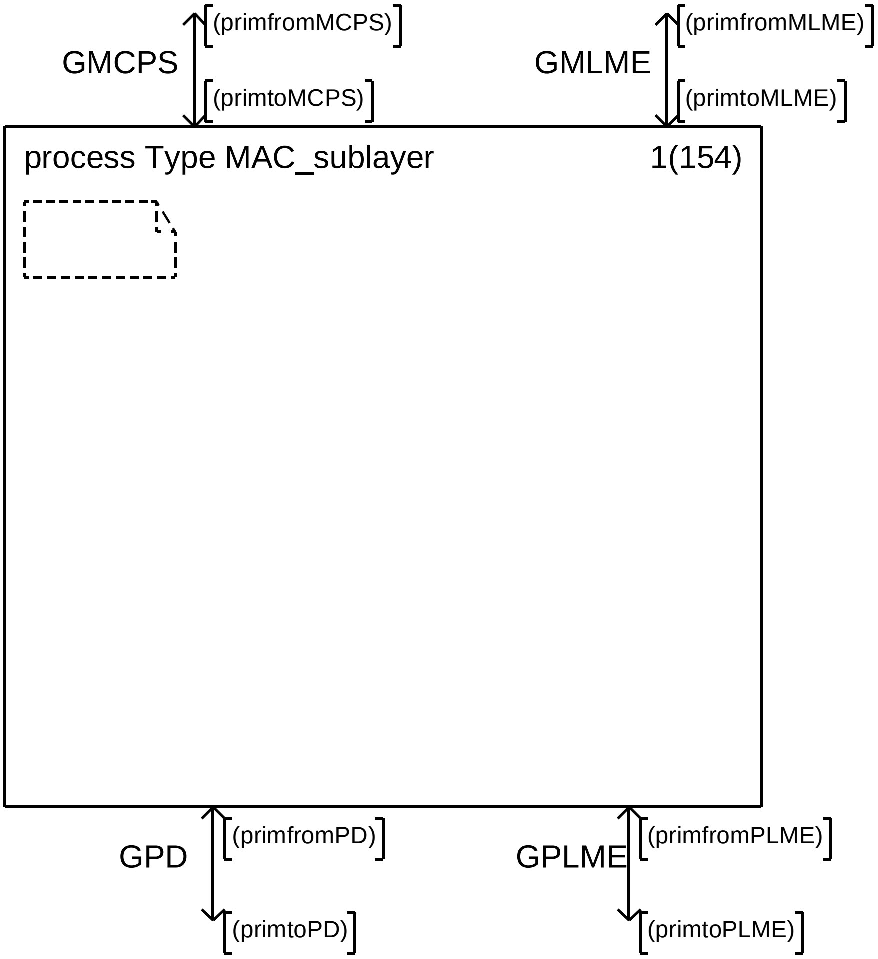


D.3.1 Block type MAC_block



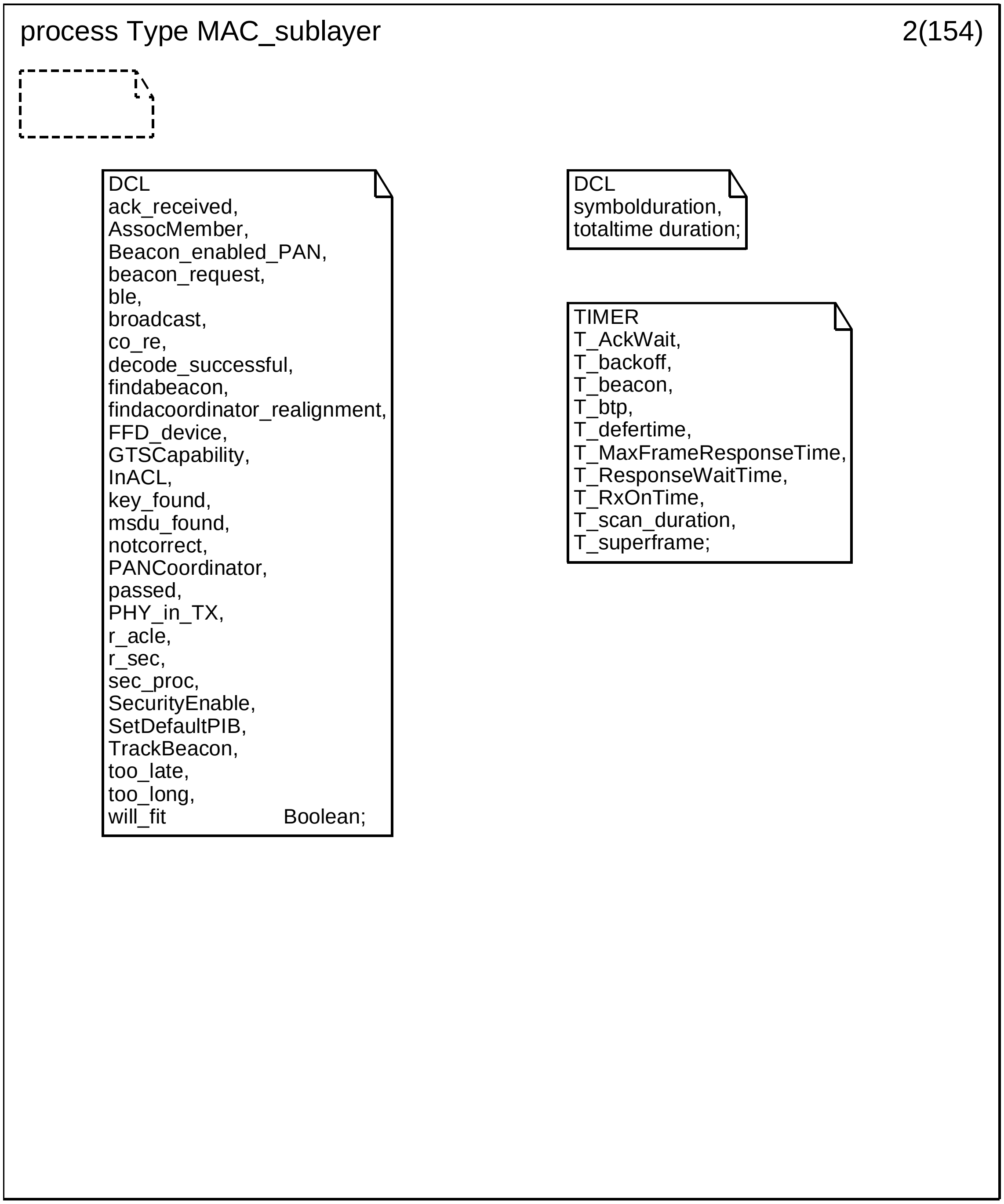
D.3.1.1 Process type MAC_sublayer (1)

.



D.3.1.2 Process type MAC_sublayer (2)

.



process Type MAC_sublayer 2(154)

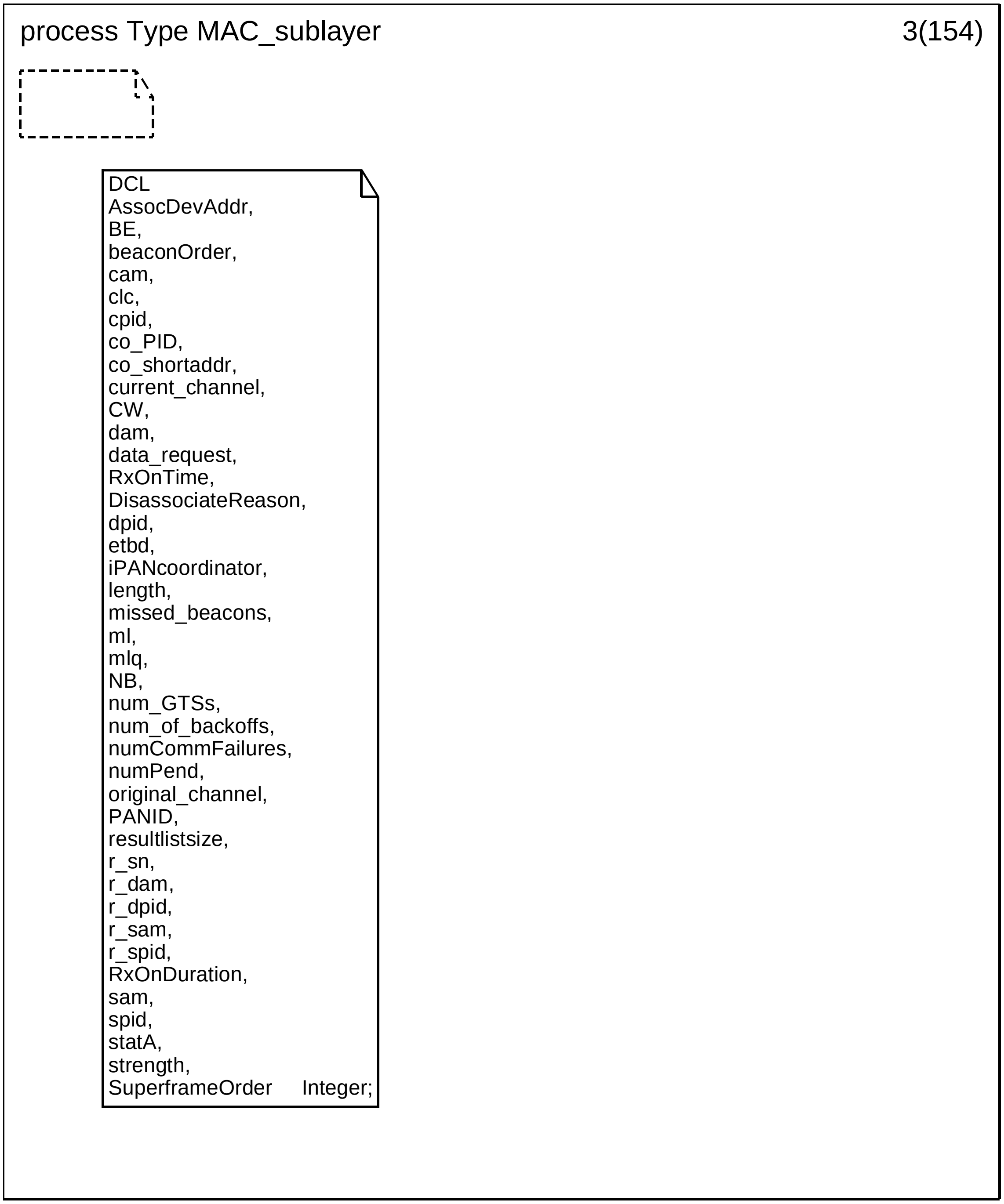
```
DCL
ack_received,
AssocMember,
Beacon_enabled_PAN,
beacon_request,
ble,
broadcast,
co_re,
decode_successful,
findabeacon,
findacoordinator_realignment,
FFD_device,
GTSCapability,
InACL,
key_found,
msdu_found,
notcorrect,
PANCoordinator,
passed,
PHY_in_TX,
r_acle,
r_sec,
sec_proc,
SecurityEnable,
SetDefaultPIB,
TrackBeacon,
too_late,
too_long,
will_fit                Boolean;
```

```
DCL
symbolduration,
totaltime duration;
```

```
TIMER
T_AckWait,
T_backoff,
T_beacon,
T_btp,
T_defertime,
T_MaxFrameResponseTime,
T_ResponseWaitTime,
T_RxOnTime,
T_scan_duration,
T_superframe;
```

D.3.1.3 Process type MAC_sublayer (3)

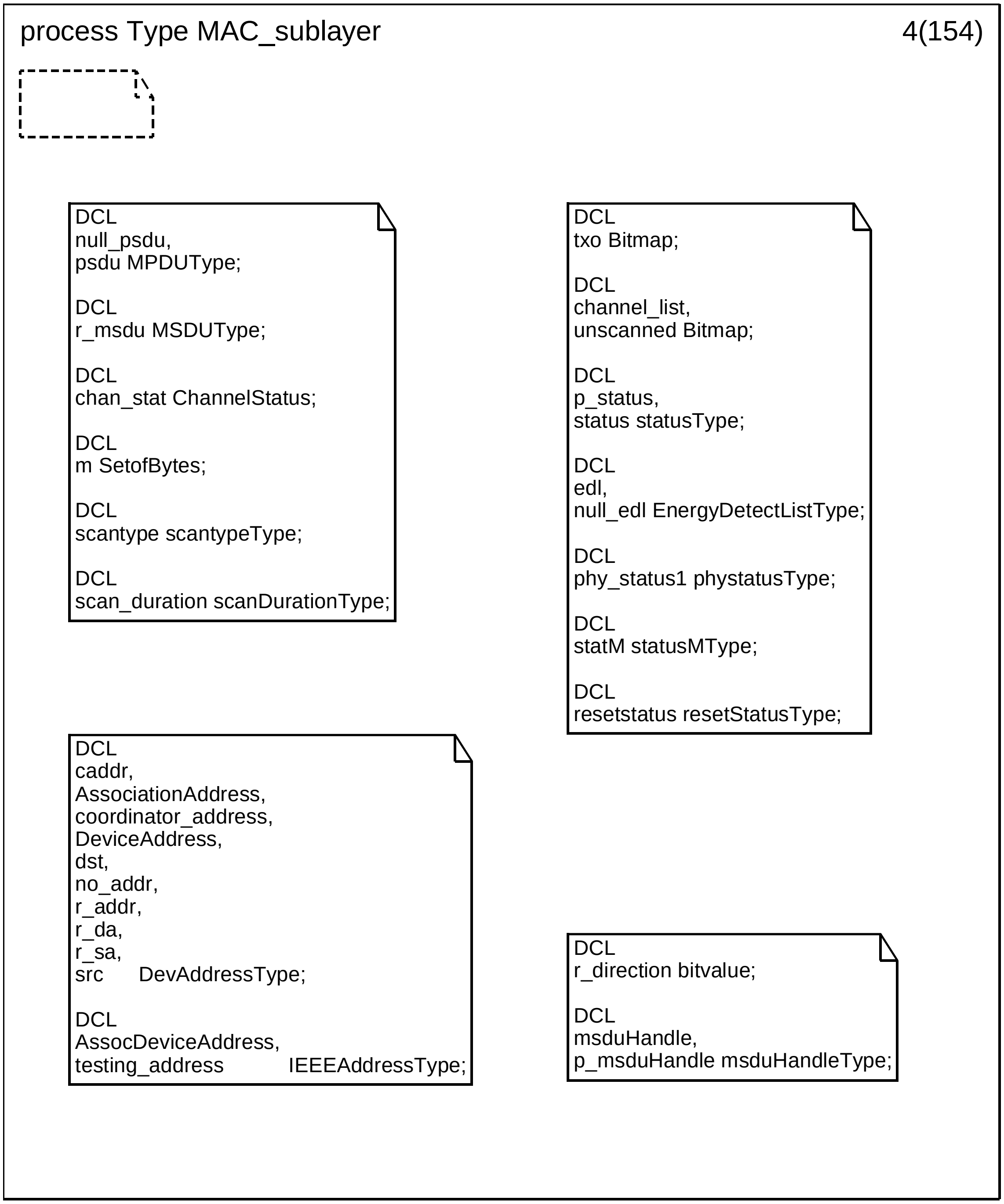
.



```
DCL
AssocDevAddr,
BE,
beaconOrder,
cam,
clc,
cpid,
co_PID,
co_shortaddr,
current_channel,
CW,
dam,
data_request,
RxOnTime,
DisassociateReason,
dpid,
etbd,
iPANcoordinator,
length,
missed_beacons,
ml,
mlq,
NB,
num_GTSs,
num_of_backoffs,
numCommFailures,
numPend,
original_channel,
PANID,
resultlistsize,
r_sn,
r_dam,
r_dpid,
r_sam,
r_spid,
RxOnDuration,
sam,
spid,
statA,
strength,
SuperframeOrder     Integer;
```

D.3.1.4 Process type MAC_sublayer (4)

.



```
DCL
null_psdu,
psdu MPDUType;

DCL
r_msdu MSDUType;

DCL
chan_stat ChannelStatus;

DCL
m SetofBytes;

DCL
scantype scantypeType;

DCL
scan_duration scanDurationType;
```

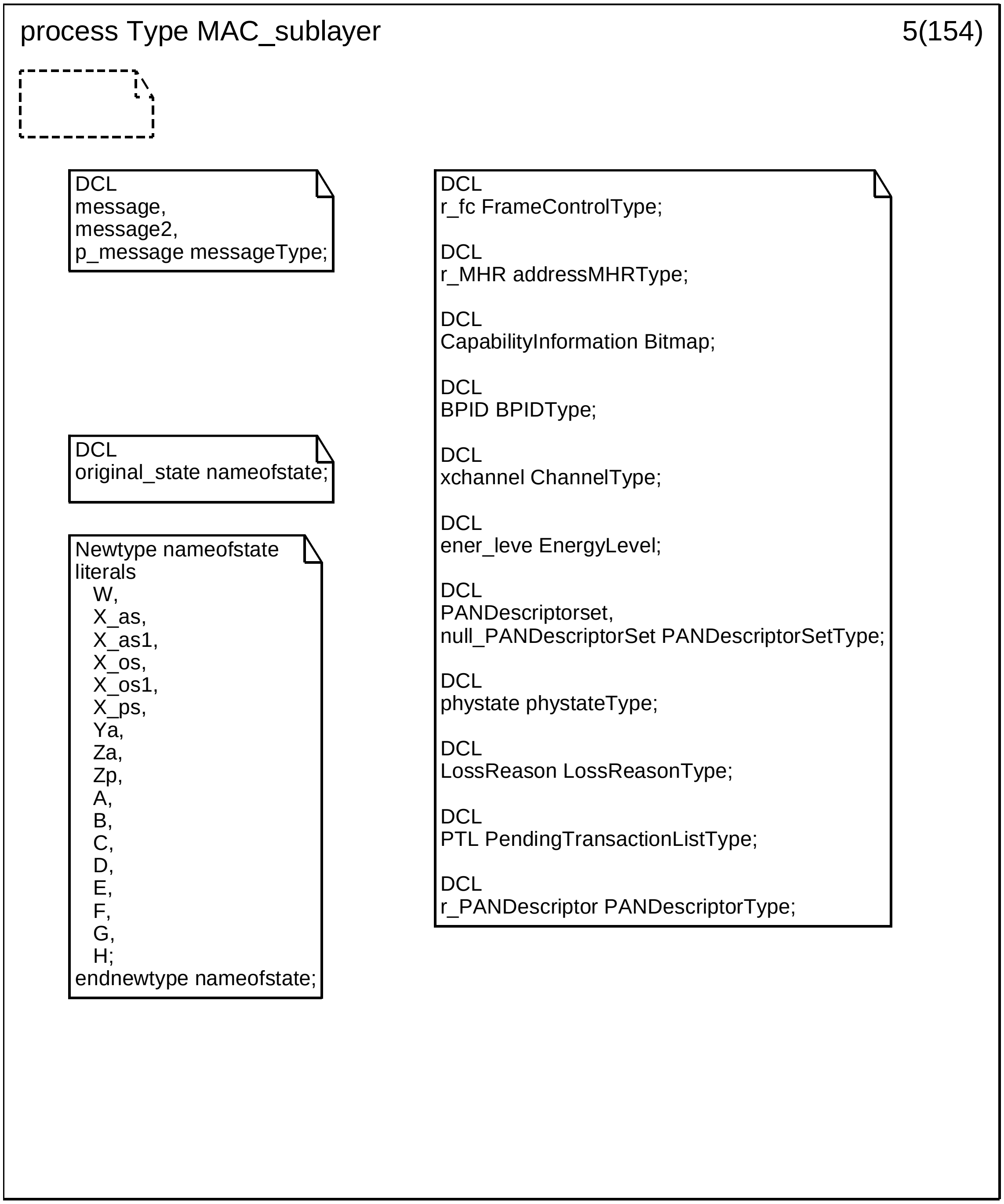
```
DCL
txo Bitmap;

DCL
channel_list,
unscanned Bitmap;

DCL
p_status,
status statusType;

DCL
edl,
null_edl EnergyDetectListType;

DCL
phy_status1 physstatusType;

DCL
statM statusMType;

DCL
resetstatus resetStatusType;
```

```
DCL
caddr,
AssociationAddress,
coordinator_address,
DeviceAddress,
dst,
no_addr,
r_addr,
r_da,
r_sa,
src      DevAddressType;

DCL
AssocDeviceAddress,
testing_address          IEEEAddressType;
```

```
DCL
r_direction bitvalue;

DCL
msduHandle,
p_msduHandle msduHandleType;
```

D.3.1.5 Process type MAC_sublayer (5)

.



process Type MAC_sublayer 5(154)

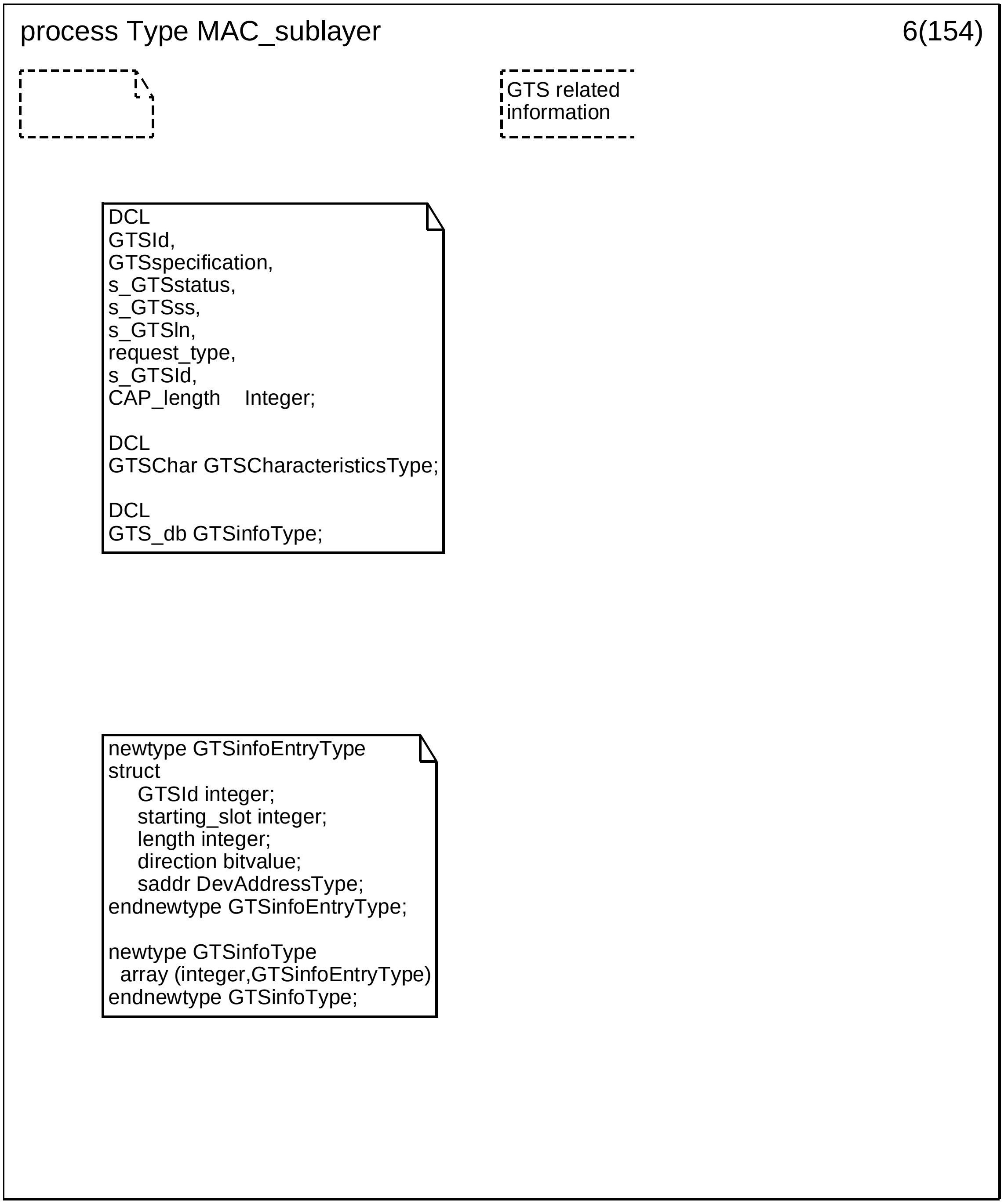
```
DCL
message,
message2,
p_message messageType;
```

```
DCL
original_state nameofstate;
```

```
Newtype nameofstate
literals
   W,
   X_as,
   X_as1,
   X_os,
   X_os1,
   X_ps,
   Ya,
   Za,
   Zp,
   A,
   B,
   C,
   D,
   E,
   F,
   G,
   H;
endnewtype nameofstate;
```

```
DCL
r_fc FrameControlType;

DCL
r_MHR addressMHRType;

DCL
CapabilityInformation Bitmap;

DCL
BPID BPIDType;

DCL
xchannel ChannelType;

DCL
ener_leve EnergyLevel;

DCL
PANDescriptorset,
null_PANDescriptorSet PANDescriptorSetType;

DCL
phystate phystateType;

DCL
LossReason LossReasonType;

DCL
PTL PendingTransactionListType;

DCL
r_PANDescriptor PANDescriptorType;
```

D.3.1.6 Process type MAC_sublayer (6)



process Type MAC_sublayer 6(154)

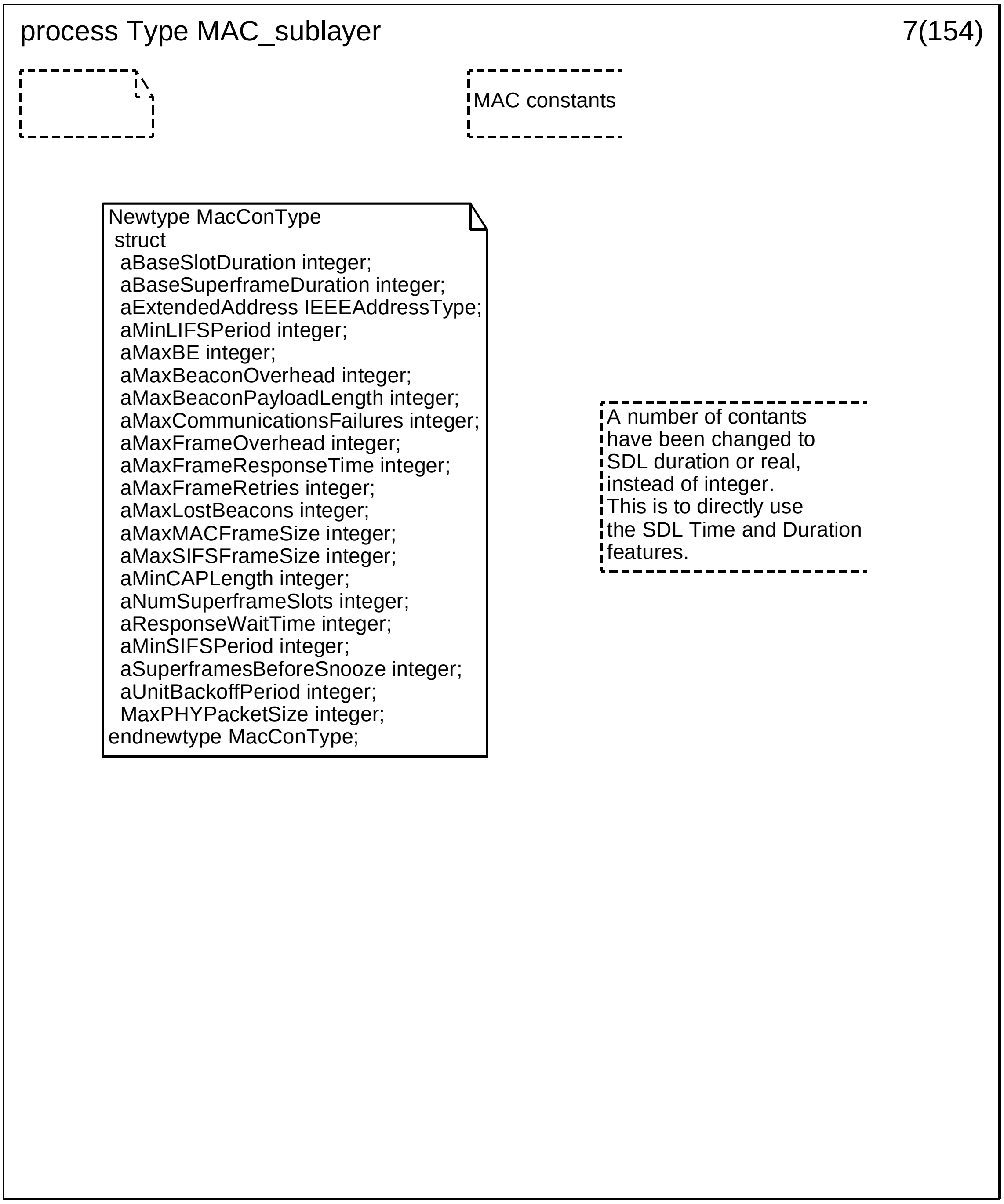
GTS related
information

```
DCL
GTSId,
GTSspecification,
s_GTSstatus,
s_GTSss,
s_GTSln,
request_type,
s_GTSId,
CAP_length    Integer;

DCL
GTSChar GTSCharacteristicsType;

DCL
GTS_db GTSinfoType;
```

```
newtype GTSinfoEntryType
struct
     GTSId integer;
     starting_slot integer;
     length integer;
     direction bitvalue;
     saddr DevAddressType;
endnewtype GTSinfoEntryType;

newtype GTSinfoType
  array (integer,GTSinfoEntryType)
endnewtype GTSinfoType;
```

D.3.1.7 Process type MAC_sublayer (7)

.

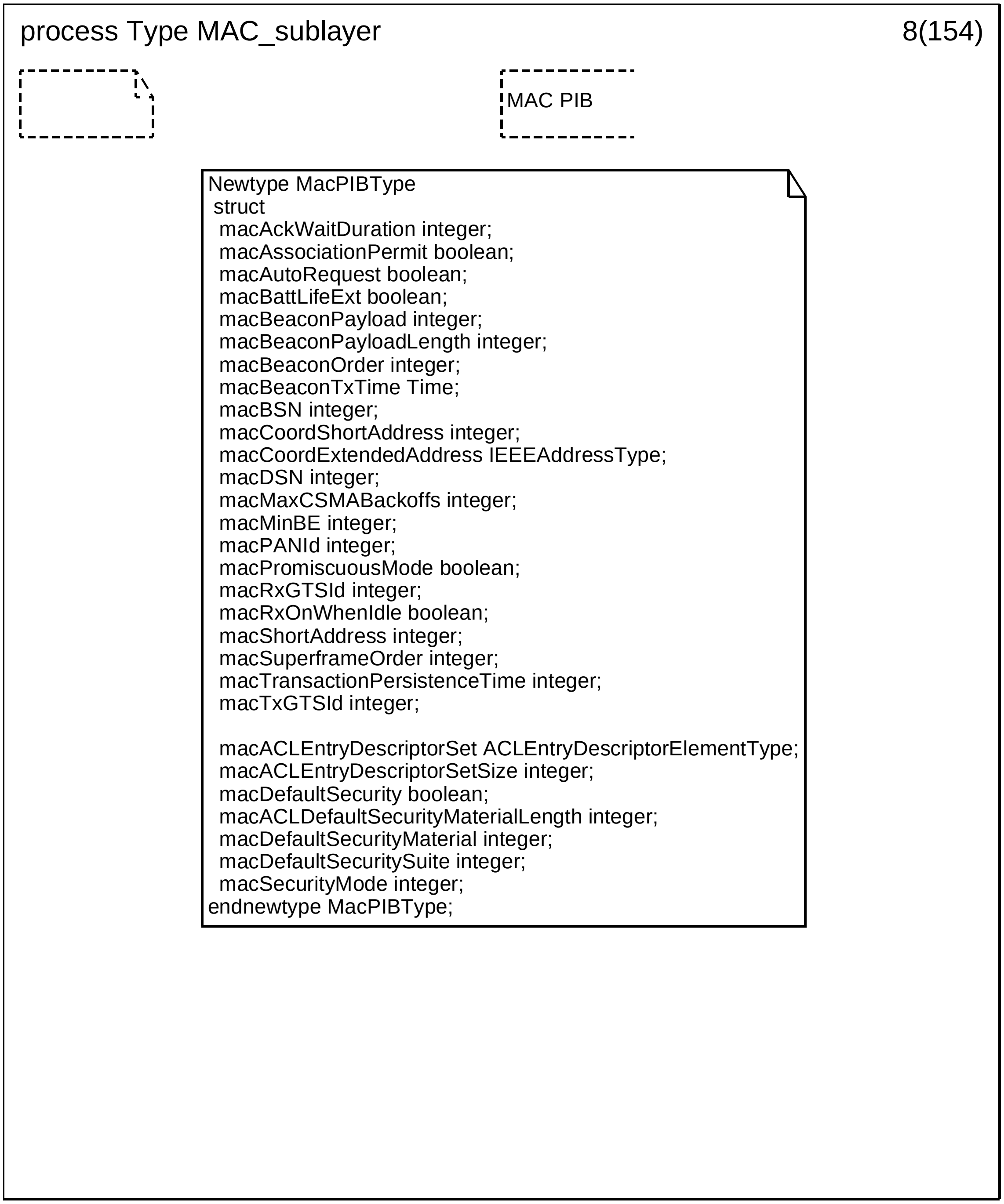


```
Newtype MacConType
 struct
  aBaseSlotDuration integer;
  aBaseSuperframeDuration integer;
  aExtendedAddress IEEEAddressType;
  aMinLIFSPeriod integer;
  aMaxBE integer;
  aMaxBeaconOverhead integer;
  aMaxBeaconPayloadLength integer;
  aMaxCommunicationsFailures integer;
  aMaxFrameOverhead integer;
  aMaxFrameResponseTime integer;
  aMaxFrameRetries integer;
  aMaxLostBeacons integer;
  aMaxMACFrameSize integer;
  aMaxSIFSFrameSize integer;
  aMinCAPLength integer;
  aNumSuperframeSlots integer;
  aResponseWaitTime integer;
  aMinSIFSPeriod integer;
  aSuperframesBeforeSnooze integer;
  aUnitBackoffPeriod integer;
  MaxPHYPacketSize integer;
endnewtype MacConType;
```

A number of contants have been changed to SDL duration or real, instead of integer. This is to directly use the SDL Time and Duration features.

D.3.1.8 Process type MAC_sublayer (8)

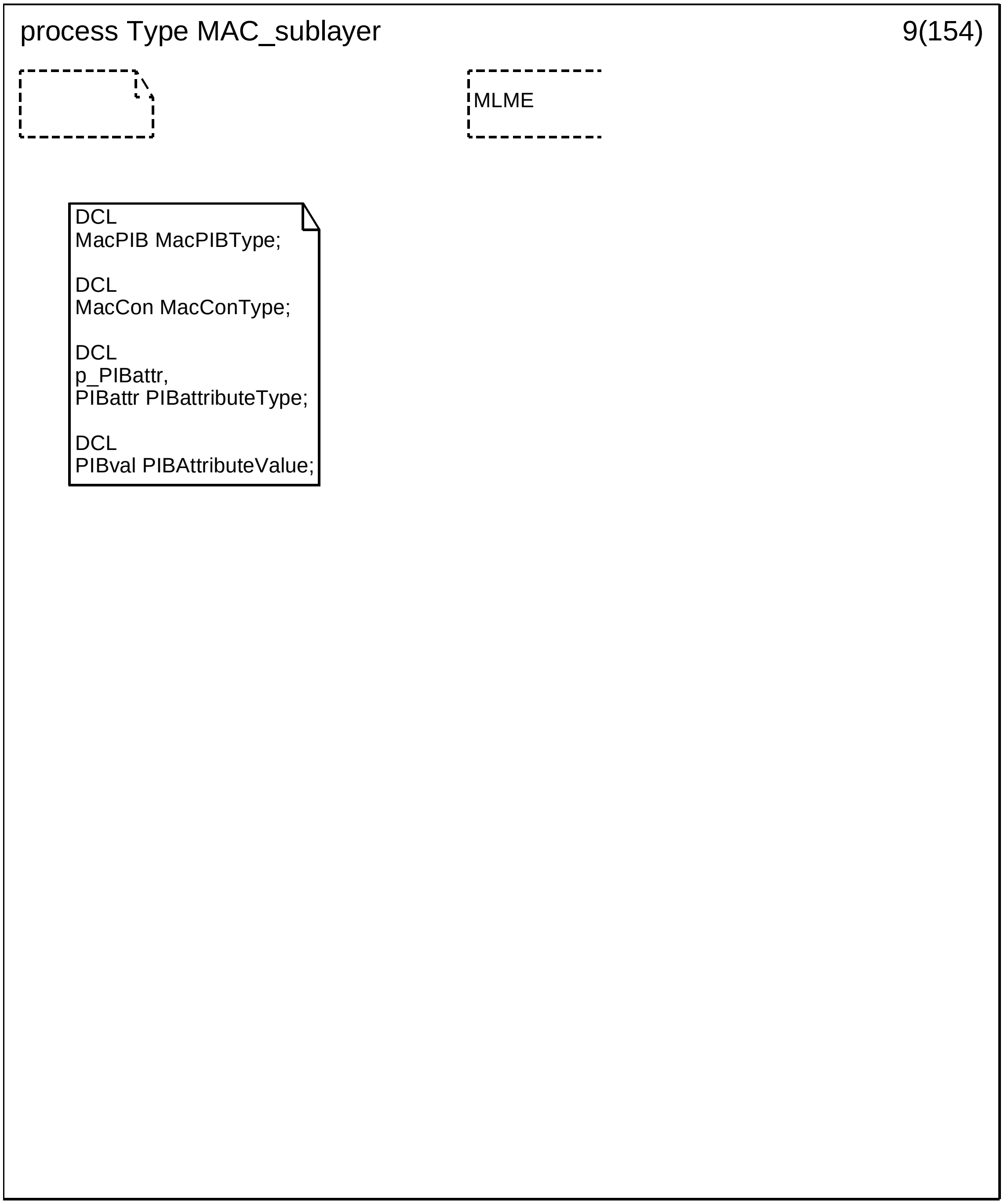
.



```
Newtype MacPIBType
 struct
  macAckWaitDuration integer;
  macAssociationPermit boolean;
  macAutoRequest boolean;
  macBattLifeExt boolean;
  macBeaconPayload integer;
  macBeaconPayloadLength integer;
  macBeaconOrder integer;
  macBeaconTxTime Time;
  macBSN integer;
  macCoordShortAddress integer;
  macCoordExtendedAddress IEEEAddressType;
  macDSN integer;
  macMaxCSMABackoffs integer;
  macMinBE integer;
  macPANId integer;
  macPromiscuousMode boolean;
  macRxGTSId integer;
  macRxOnWhenIdle boolean;
  macShortAddress integer;
  macSuperframeOrder integer;
  macTransactionPersistenceTime integer;
  macTxGTSId integer;

  macACLEntryDescriptorSet ACLEntryDescriptorElementType;
  macACLEntryDescriptorSetSize integer;
  macDefaultSecurity boolean;
  macACLDefaultSecurityMaterialLength integer;
  macDefaultSecurityMaterial integer;
  macDefaultSecuritySuite integer;
  macSecurityMode integer;
endnewtype MacPIBType;
```

D.3.1.9 Process type MAC_sublayer (9)

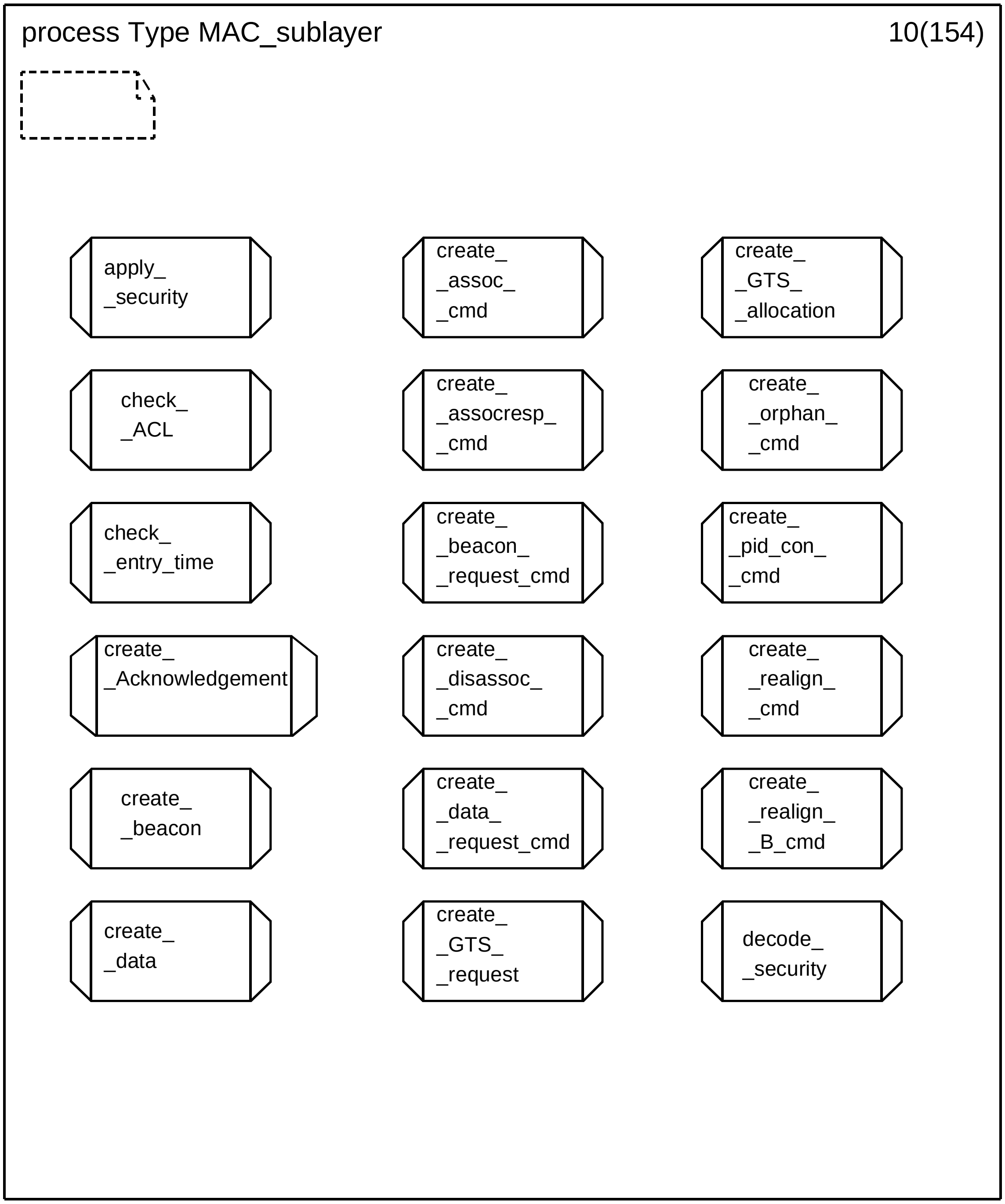
.



```
DCL
MacPIB MacPIBType;

DCL
MacCon MacConType;

DCL
p_PIBattr,
PIBattr PIBattributeType;

DCL
PIBval PIBAttributeValue;
```

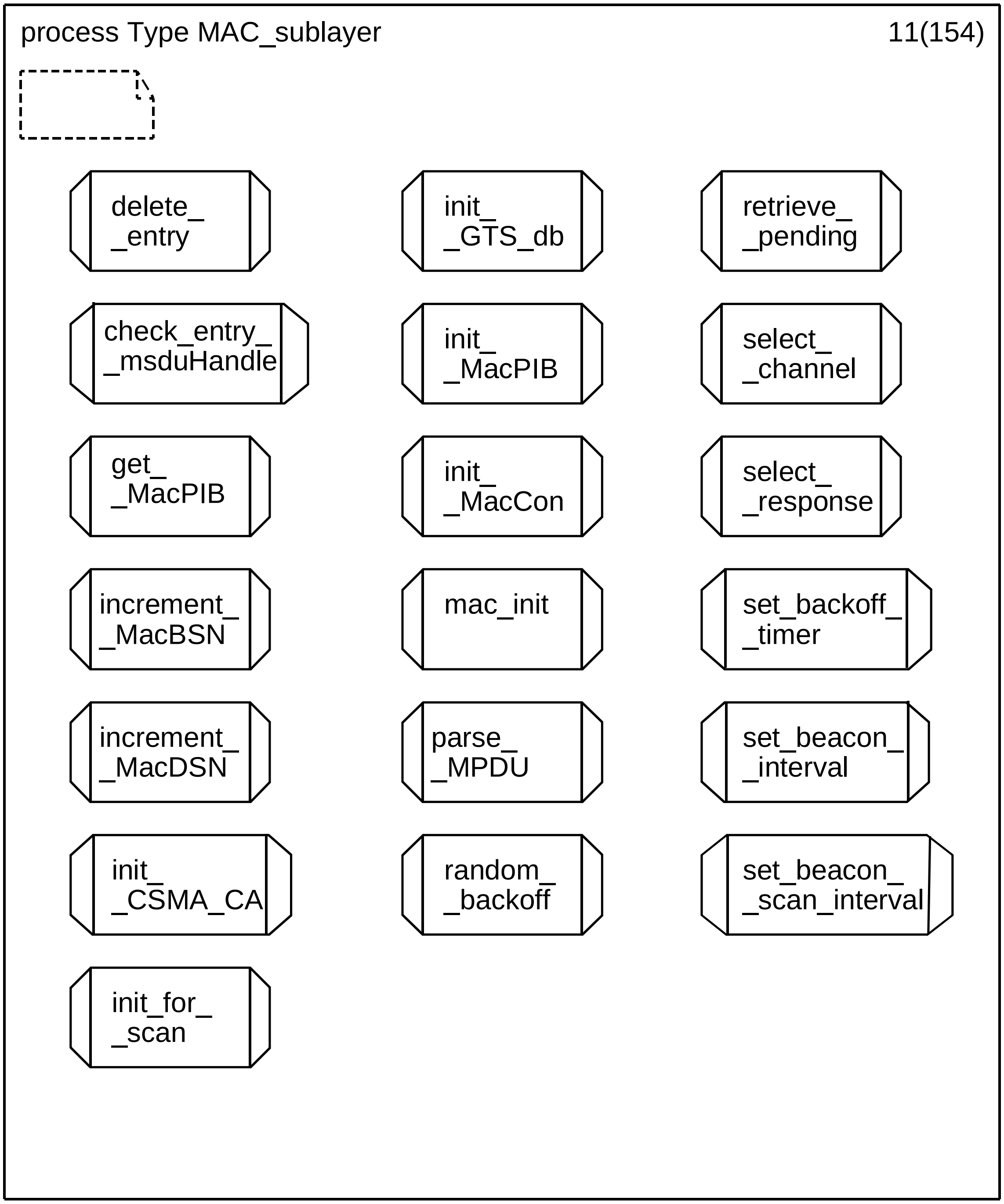
D.3.1.10 Process type MAC_sublayer (10)

.

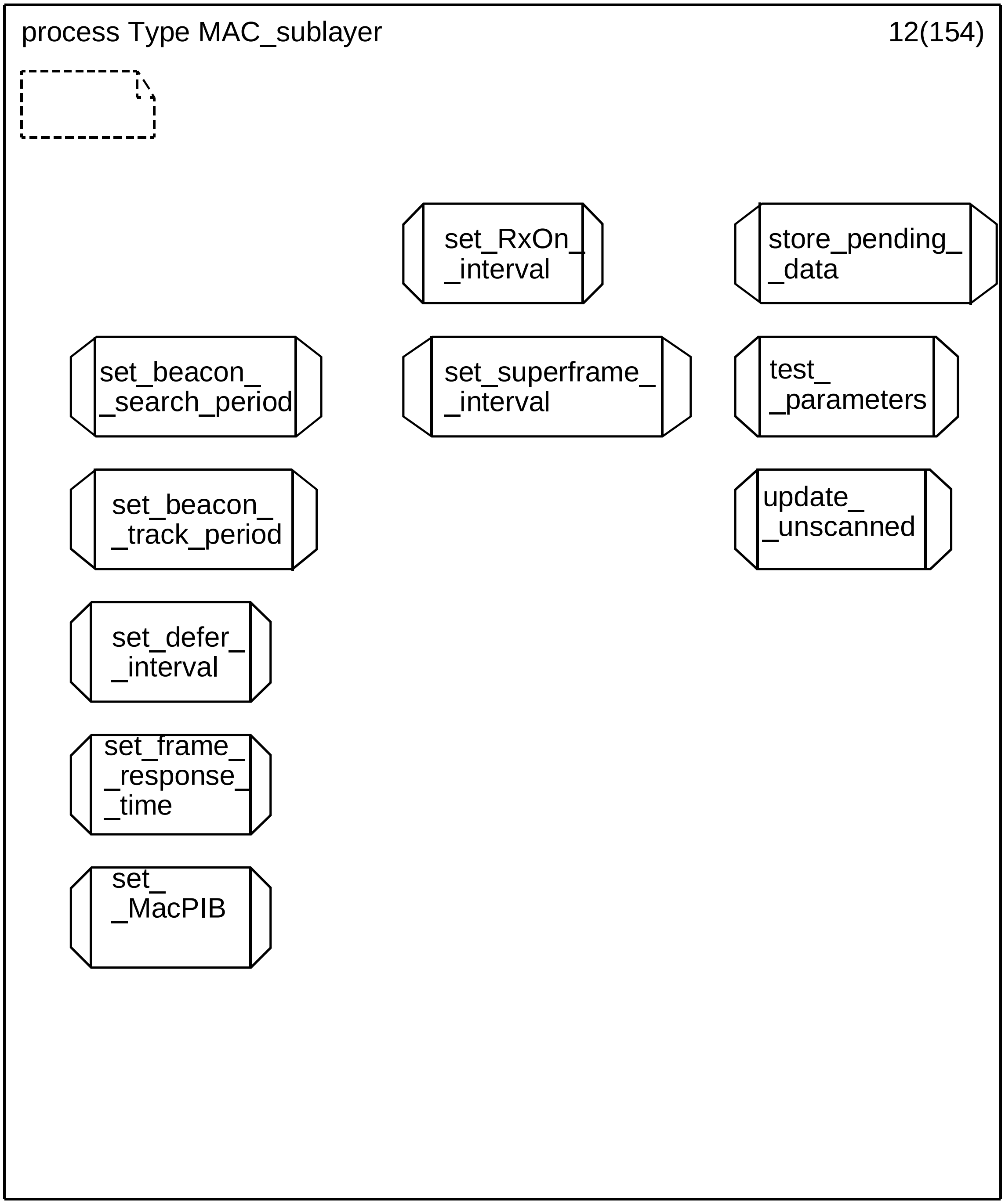


D.3.1.11 Process type MAC_sublayer (11)

.

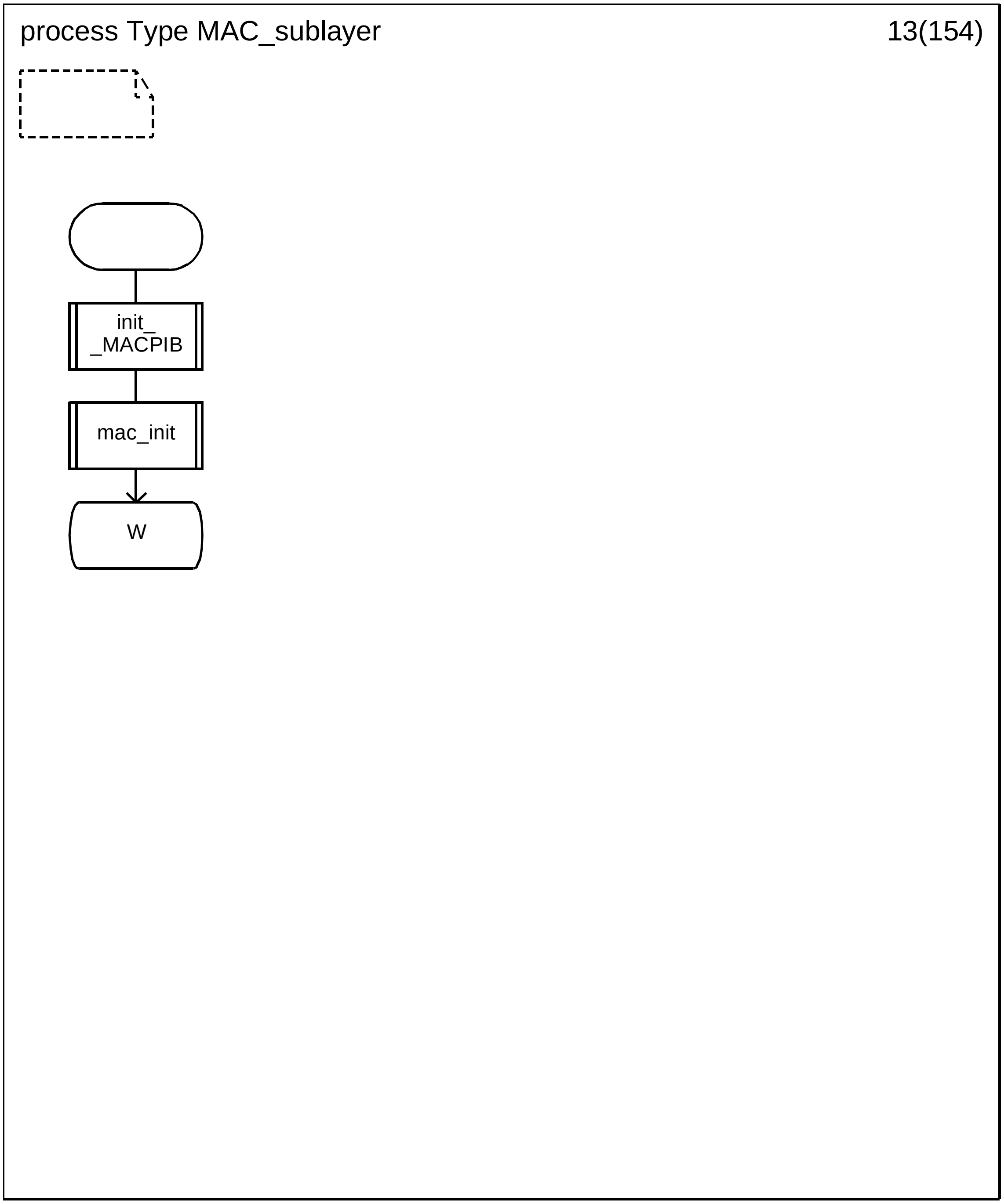


D.3.1.12 Process type MAC_sublayer (12)



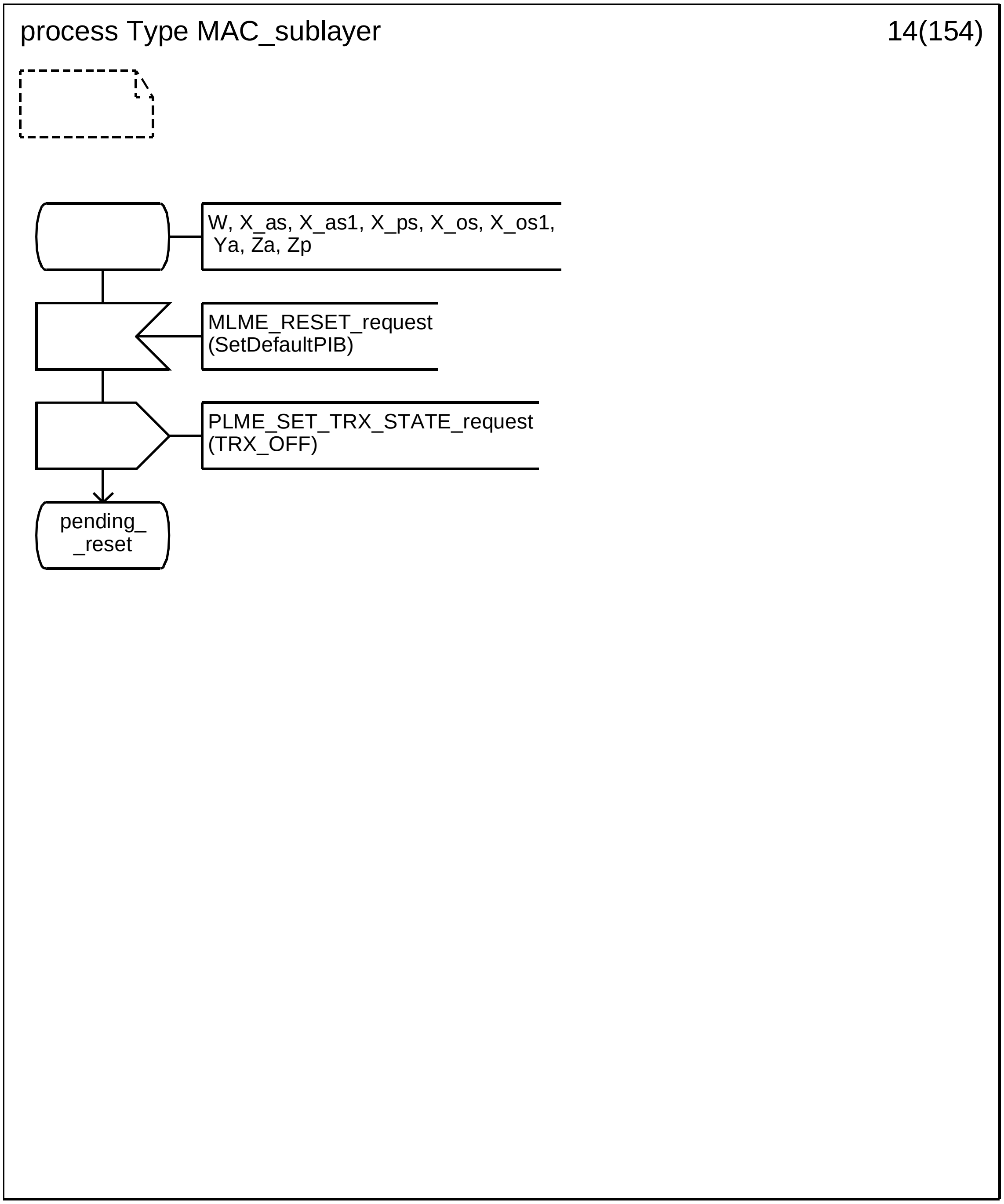
D.3.1.13 Process type MAC_sublayer (13)

.

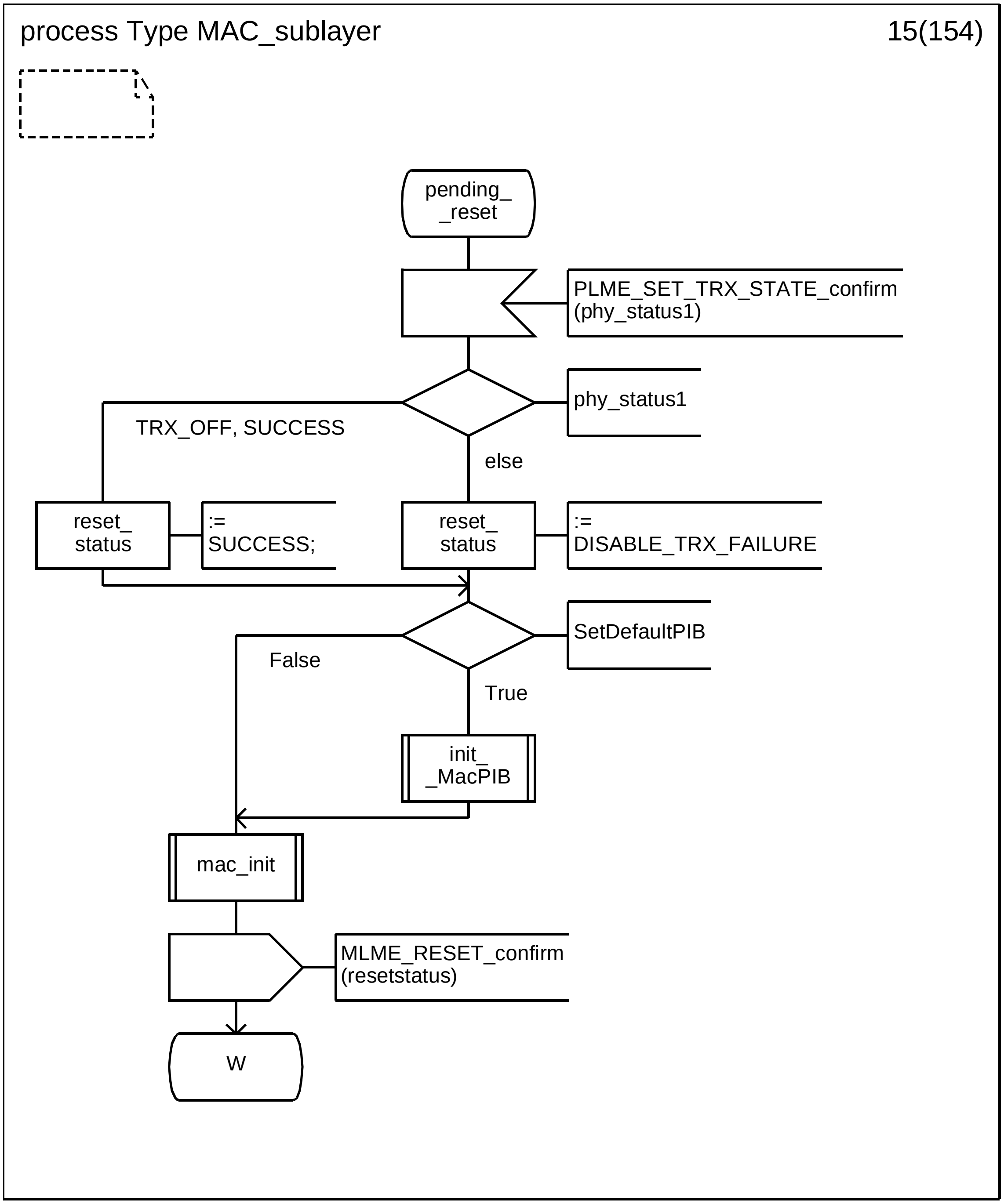


D.3.1.14 Process type MAC_sublayer (14)

.

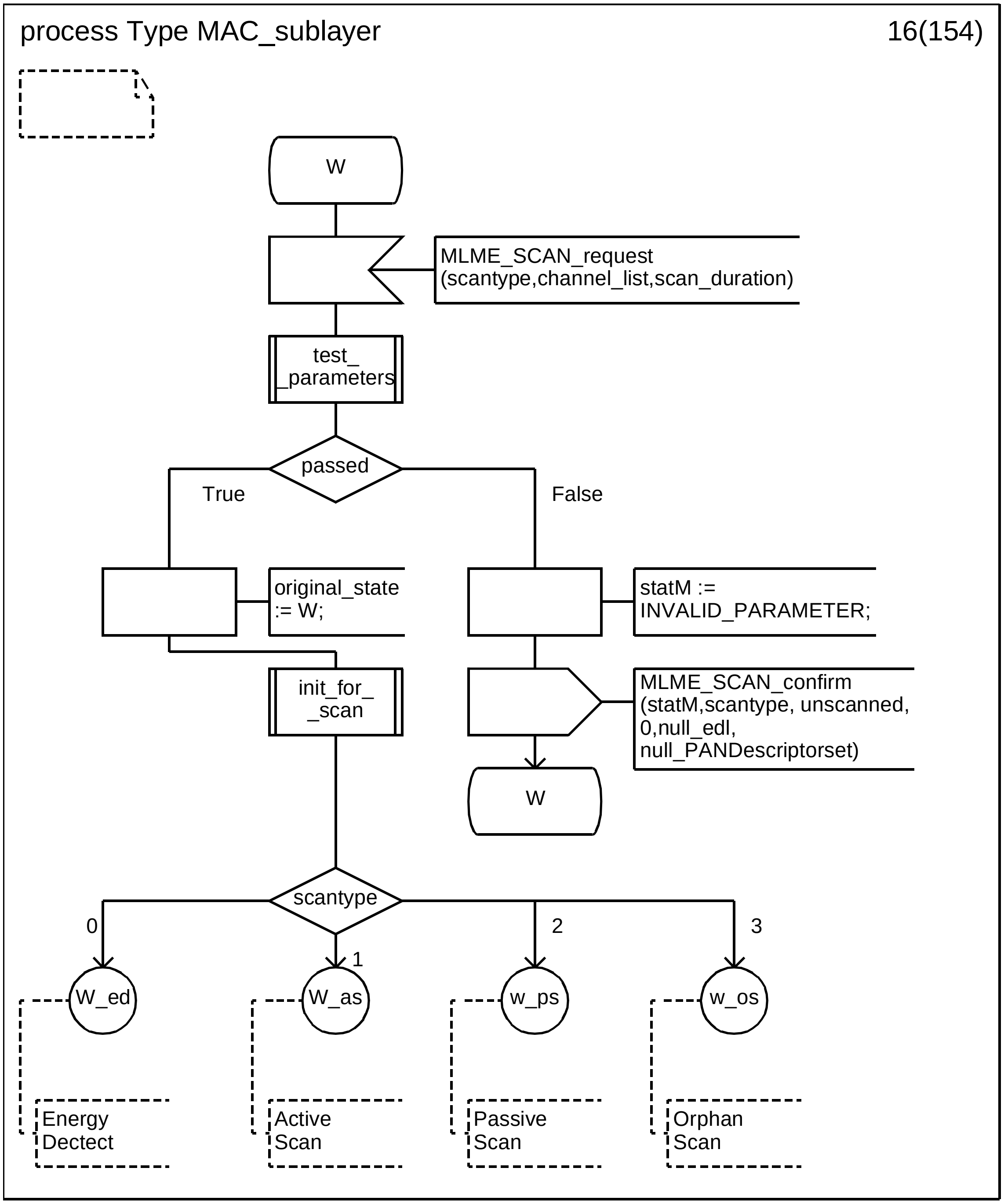

process Type MAC_sublayer
14(154)
W, X_as, X_as1, X_ps, X_os, X_os1,
Ya, Za, Zp
MLME_RESET_request
(SetDefaultPIB)
PLME_SET_TRX_STATE_request
(TRX_OFF)
pending_
_reset

D.3.1.15 Process type MAC_sublayer (15)

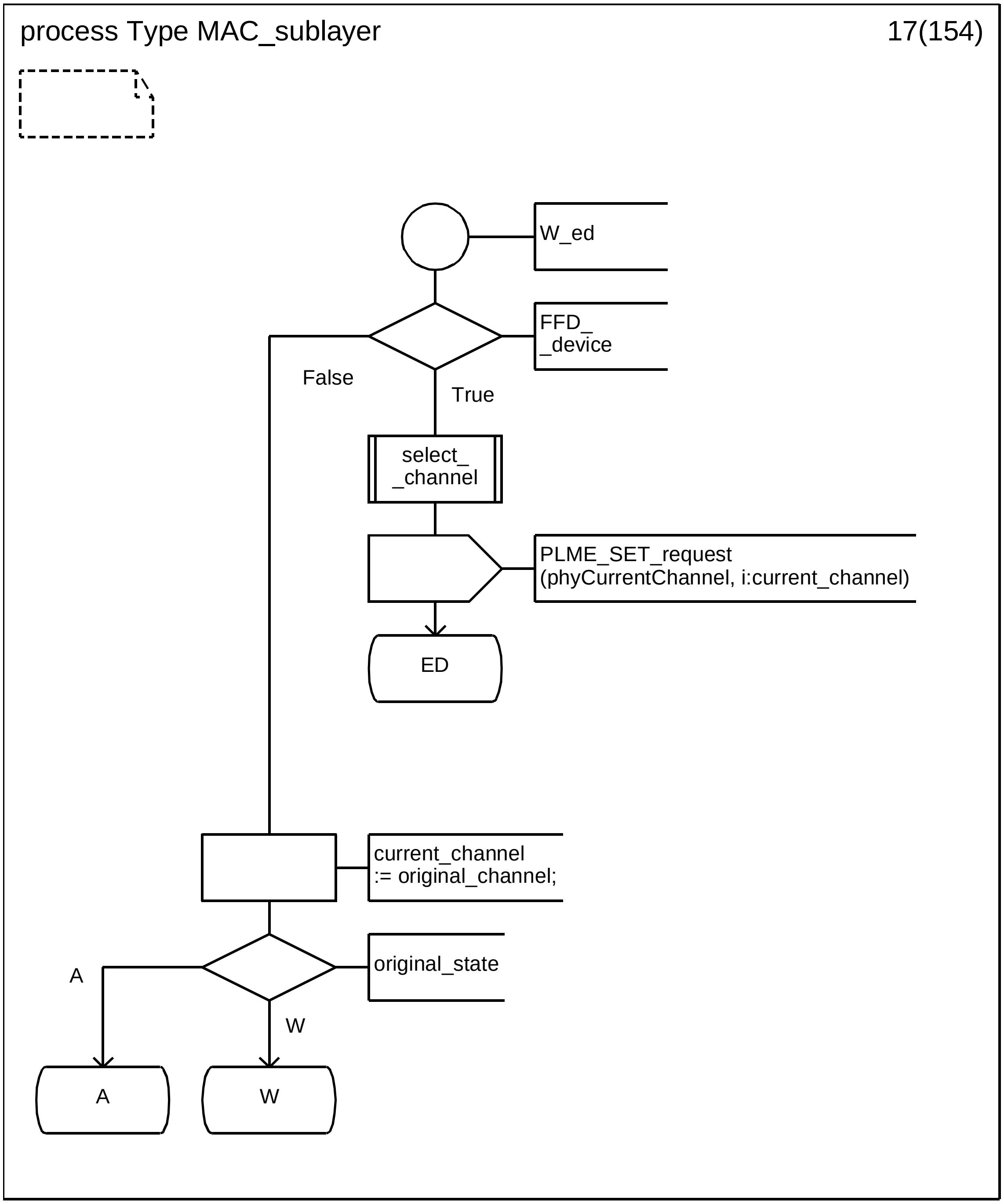


D.3.1.16 Process type MAC_sublayer (16)

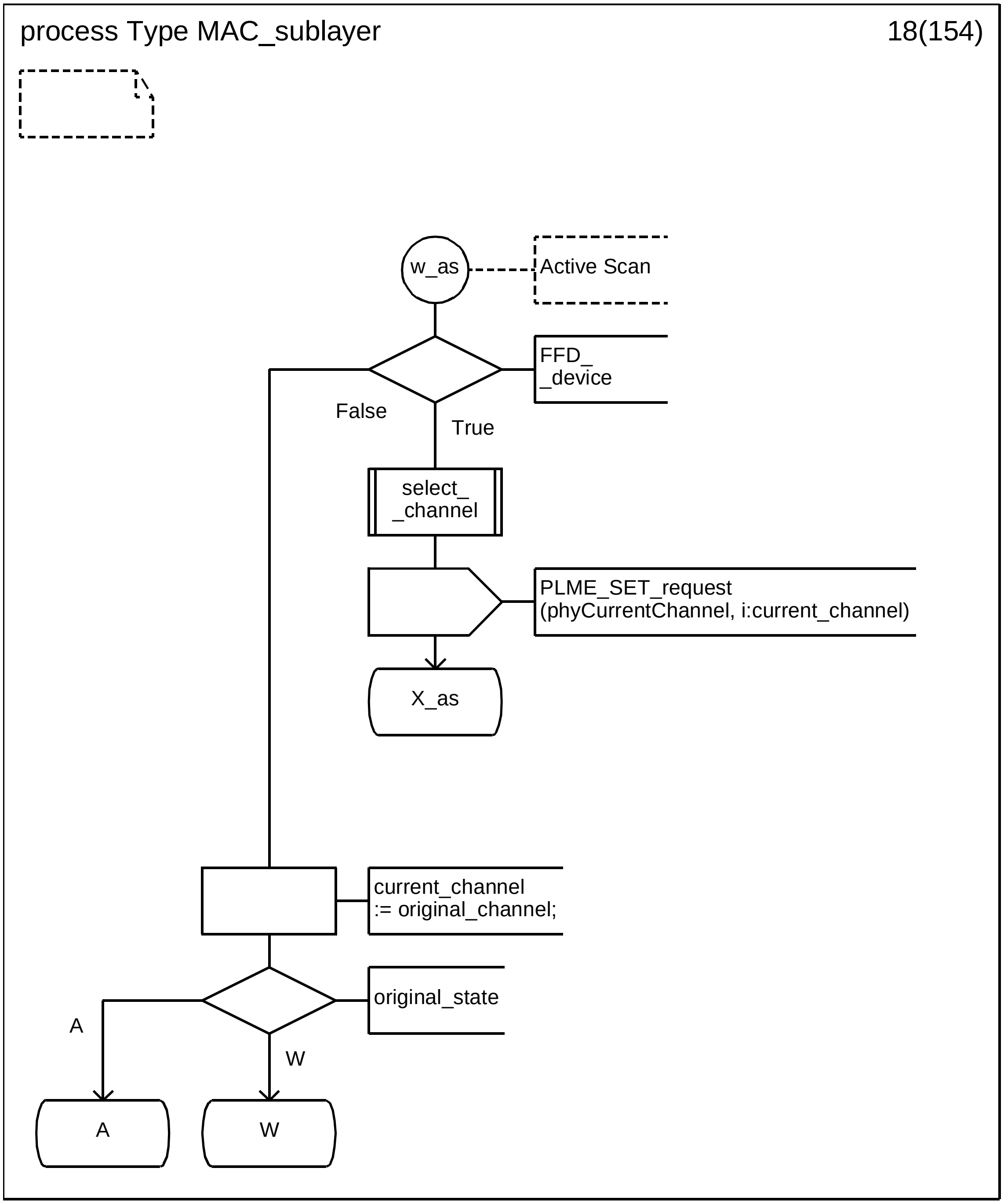
.



D.3.1.17 Process type MAC_sublayer (17)

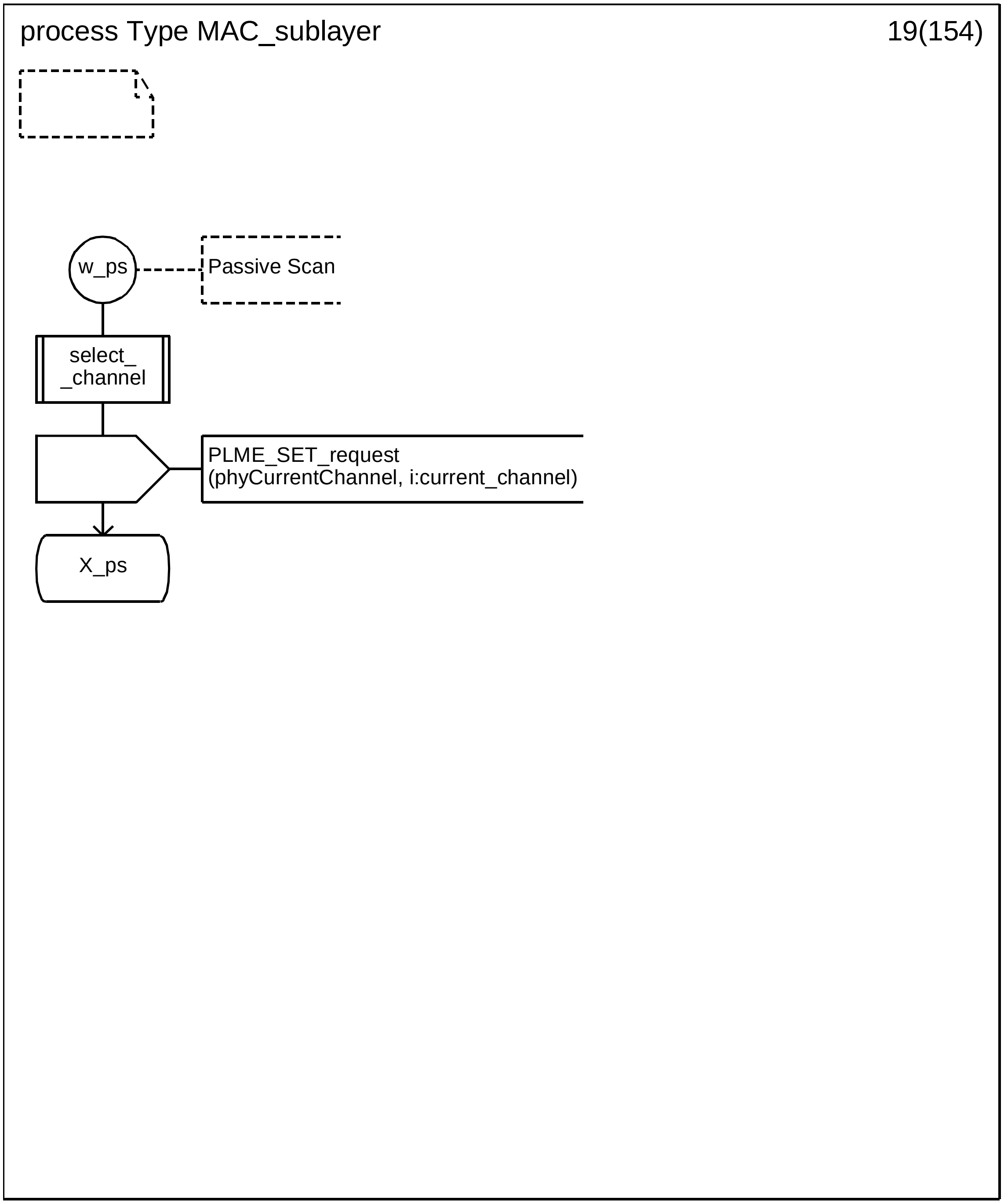


D.3.1.18 Process type MAC_sublayer (18)



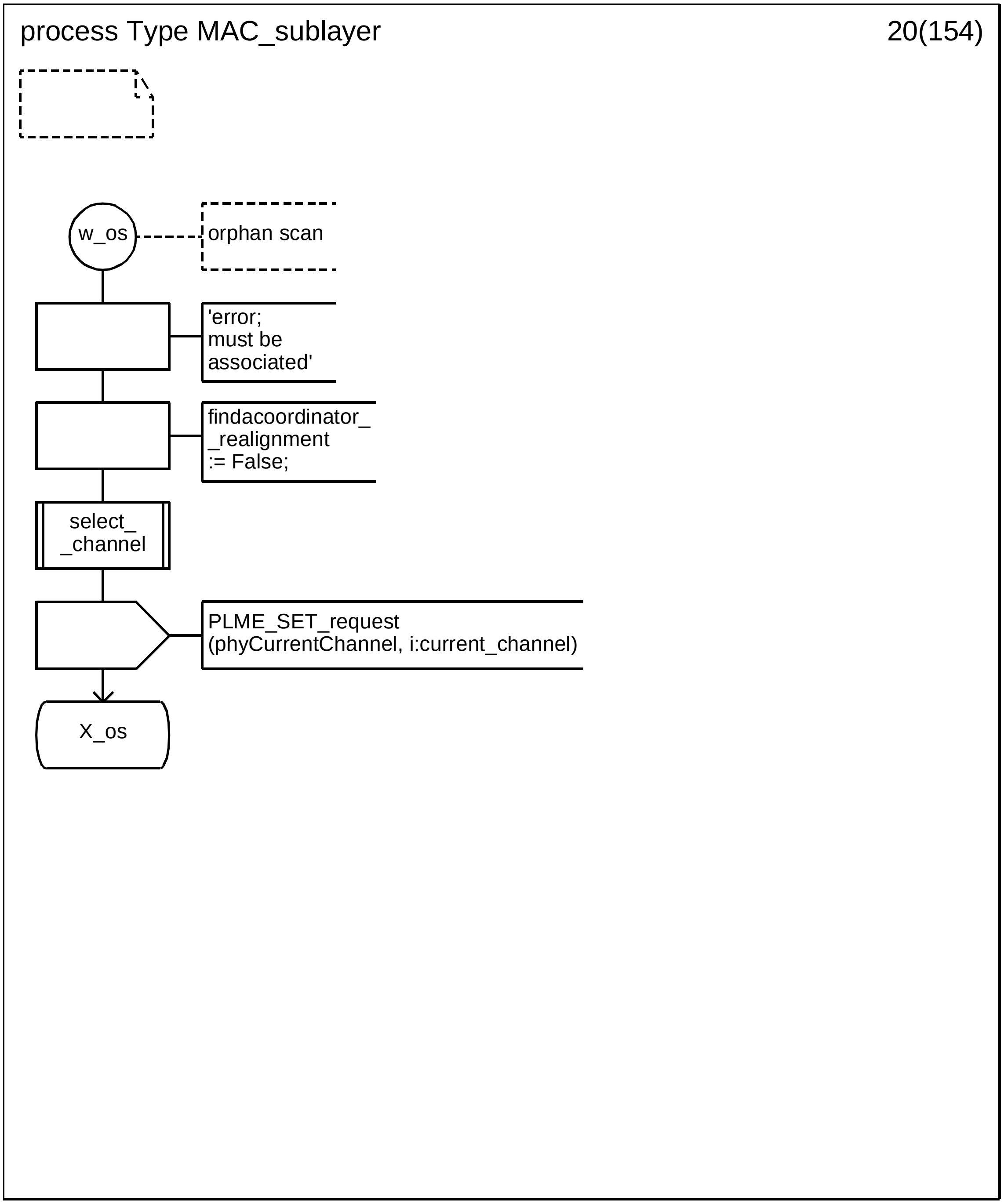
D.3.1.19 Process type MAC_sublayer (19)

.

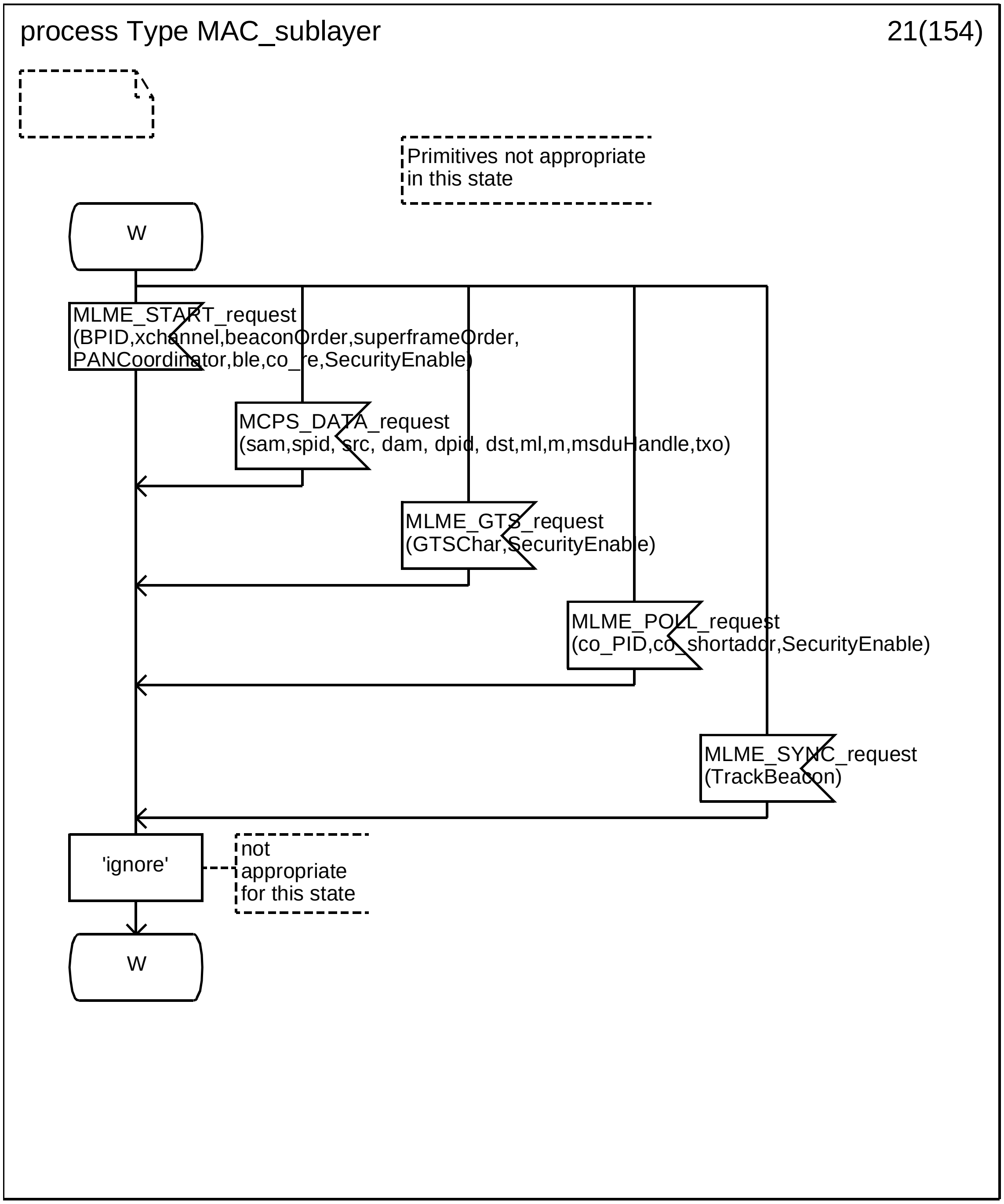

process Type MAC_sublayer
19(154)
w_ps
Passive Scan
select_
_channel
PLME_SET_request
(phyCurrentChannel, i:current_channel)
X_ps

D.3.1.20 Process type MAC_sublayer (20)

.

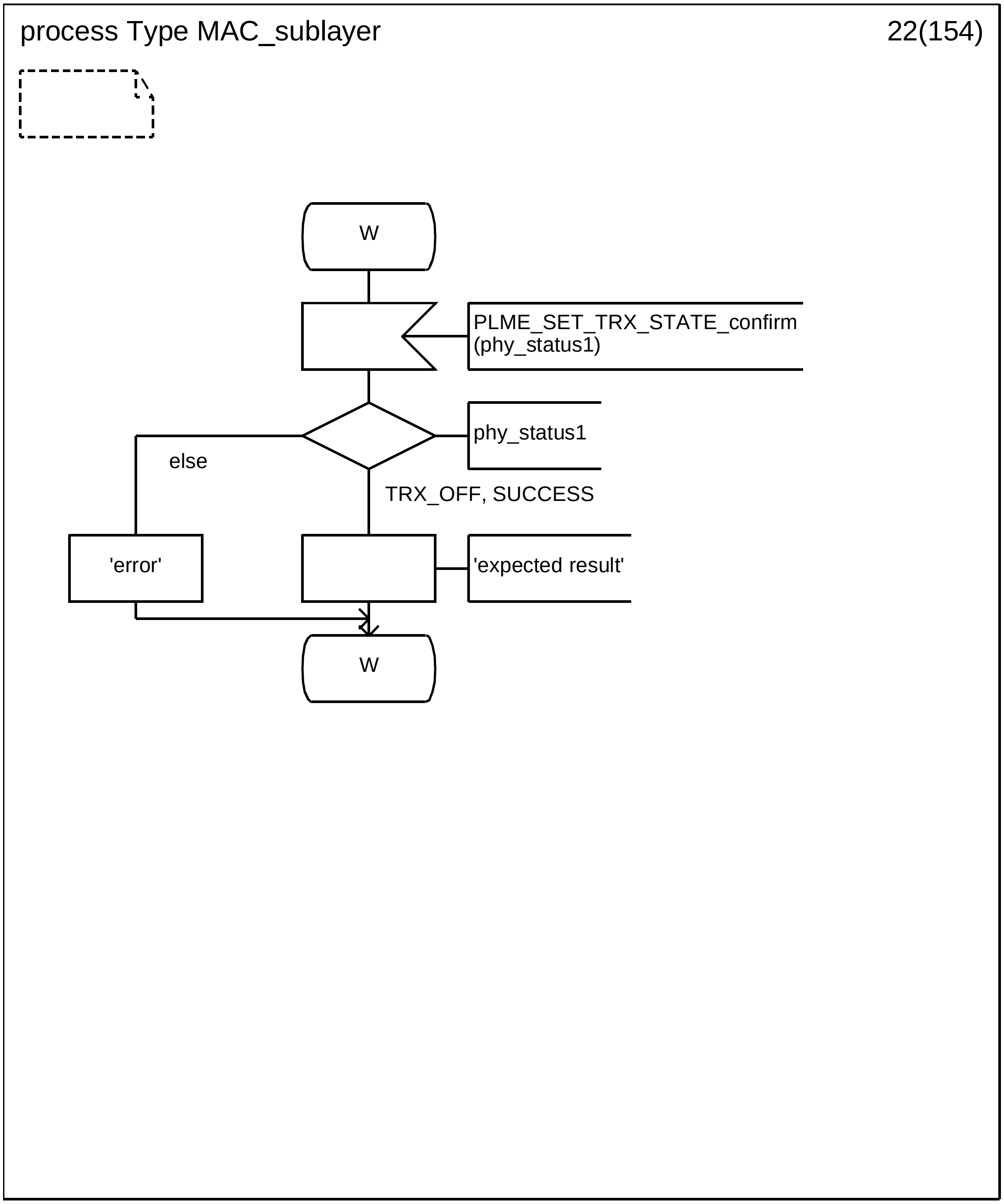


D.3.1.21 Process type MAC_sublayer (21)


process Type MAC_sublayer
21(154)
Primitives not appropriate in this state
W
MLME_START_request (BPID,xchannel,beaconOrder,superframeOrder, PANCoordinator,ble,co_re,SecurityEnable)
MCPS_DATA_request (sam,spid, src, dam, dpid, dst,ml,m,msduHandle,txo)
MLME_GTS_request (GTSChar,SecurityEnable)
MLME_POLL_request (co_PID,co_shortaddr,SecurityEnable)
MLME_SYNC_request (TrackBeacon)
'ignore'
not appropriate for this state
W

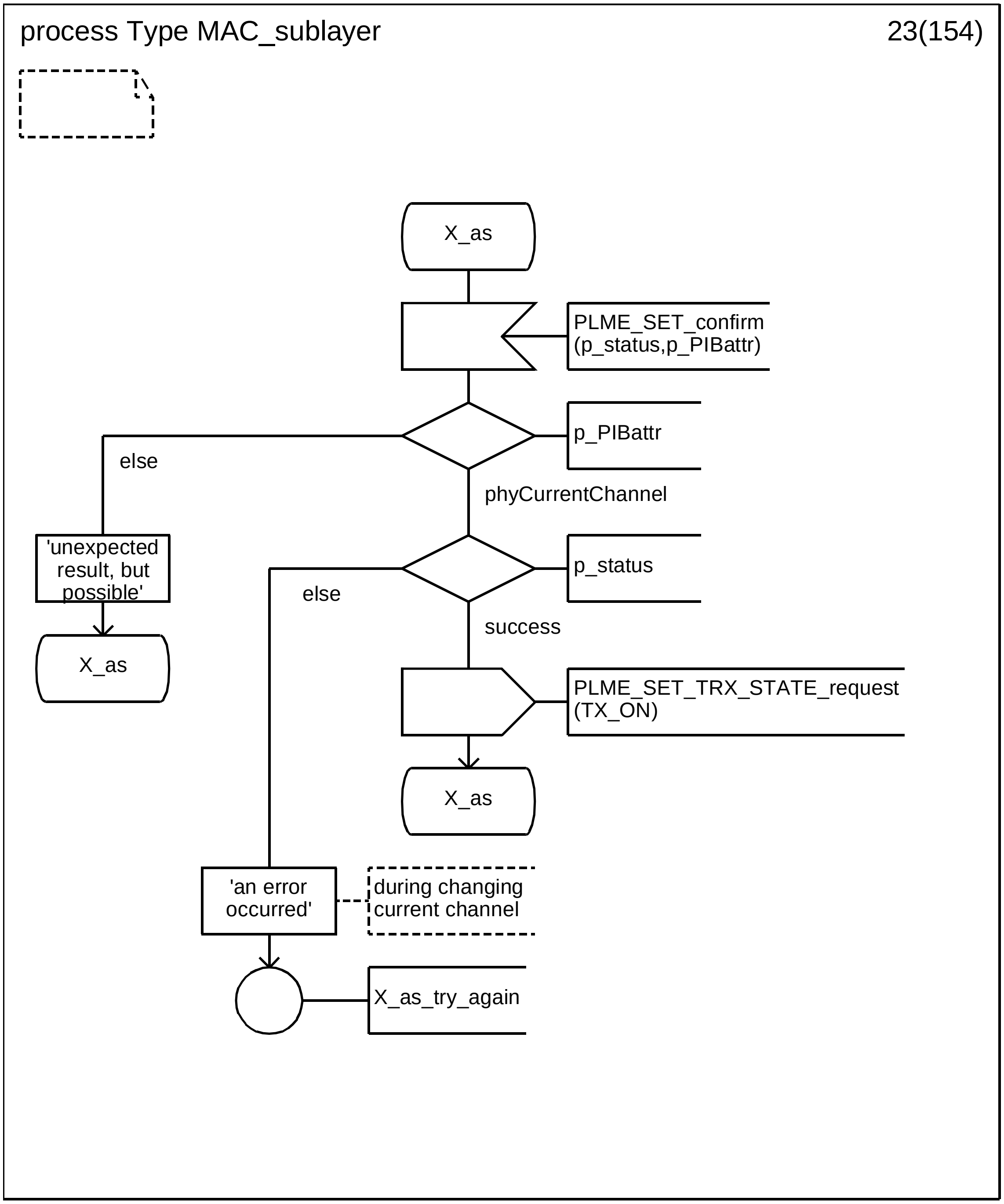
D.3.1.22 Process type MAC_sublayer (22)

.



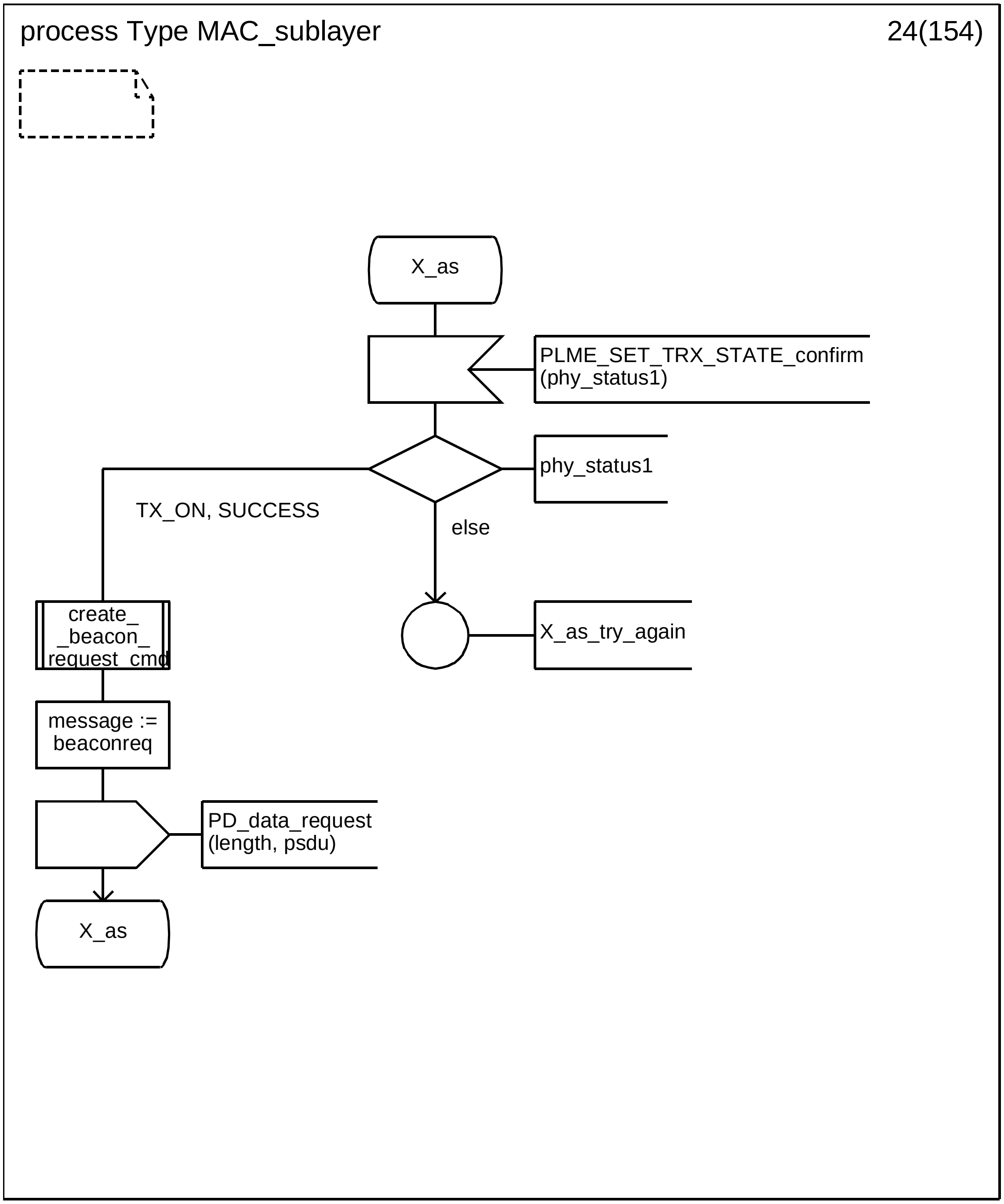
D.3.1.23 Process type MAC_sublayer (23)

.



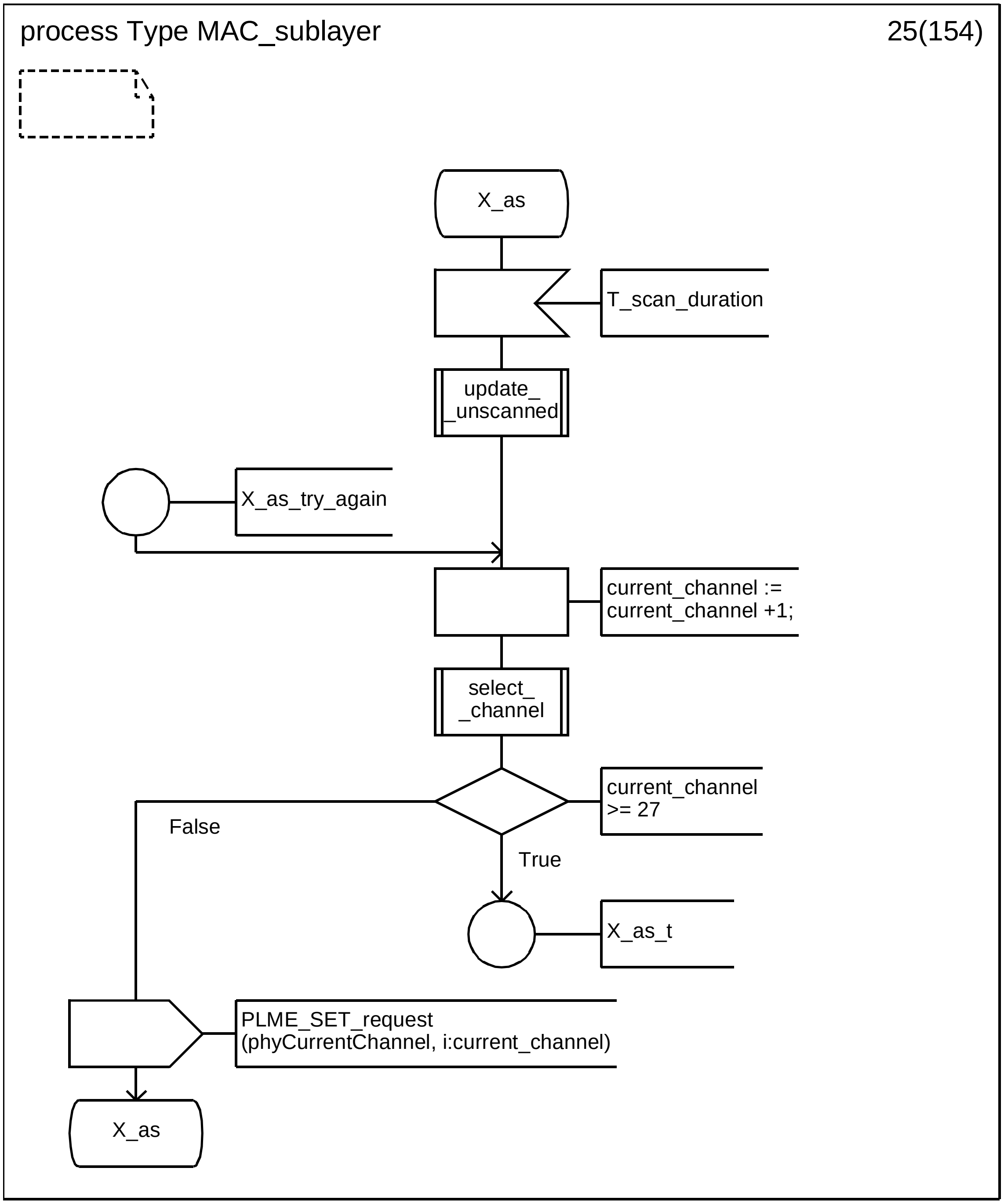
D.3.1.24 Process type MAC_sublayer (24)

.



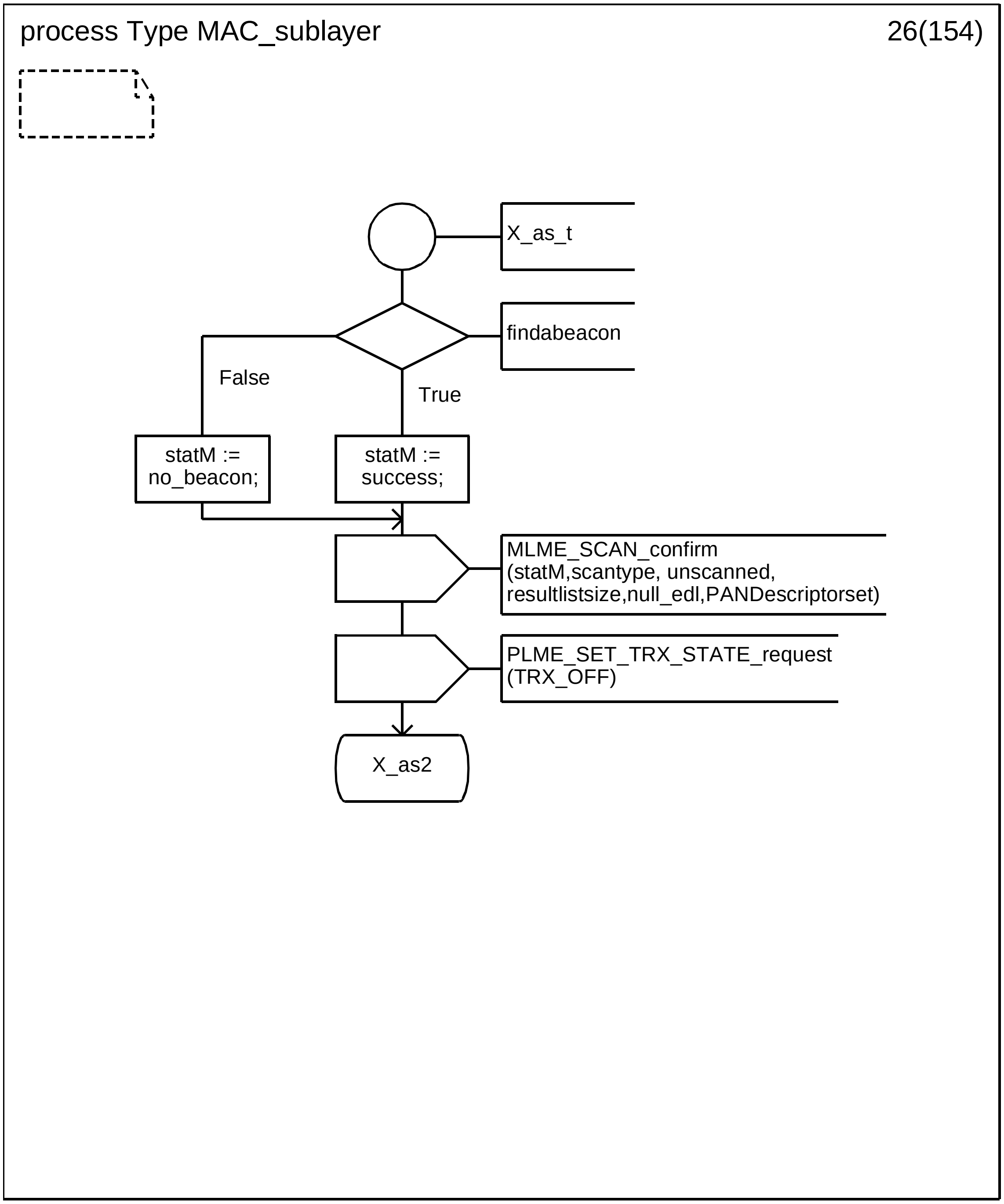
D.3.1.25 Process type MAC_sublayer (25)

.


process Type MAC_sublayer
25(154)
X_as
T_scan_duration
update_ _unscanned
X_as_try_again
current_channel := current_channel +1;
select_ _channel
current_channel >= 27
False
True
X_as_t
PLME_SET_request (phyCurrentChannel, i:current_channel)
X_as

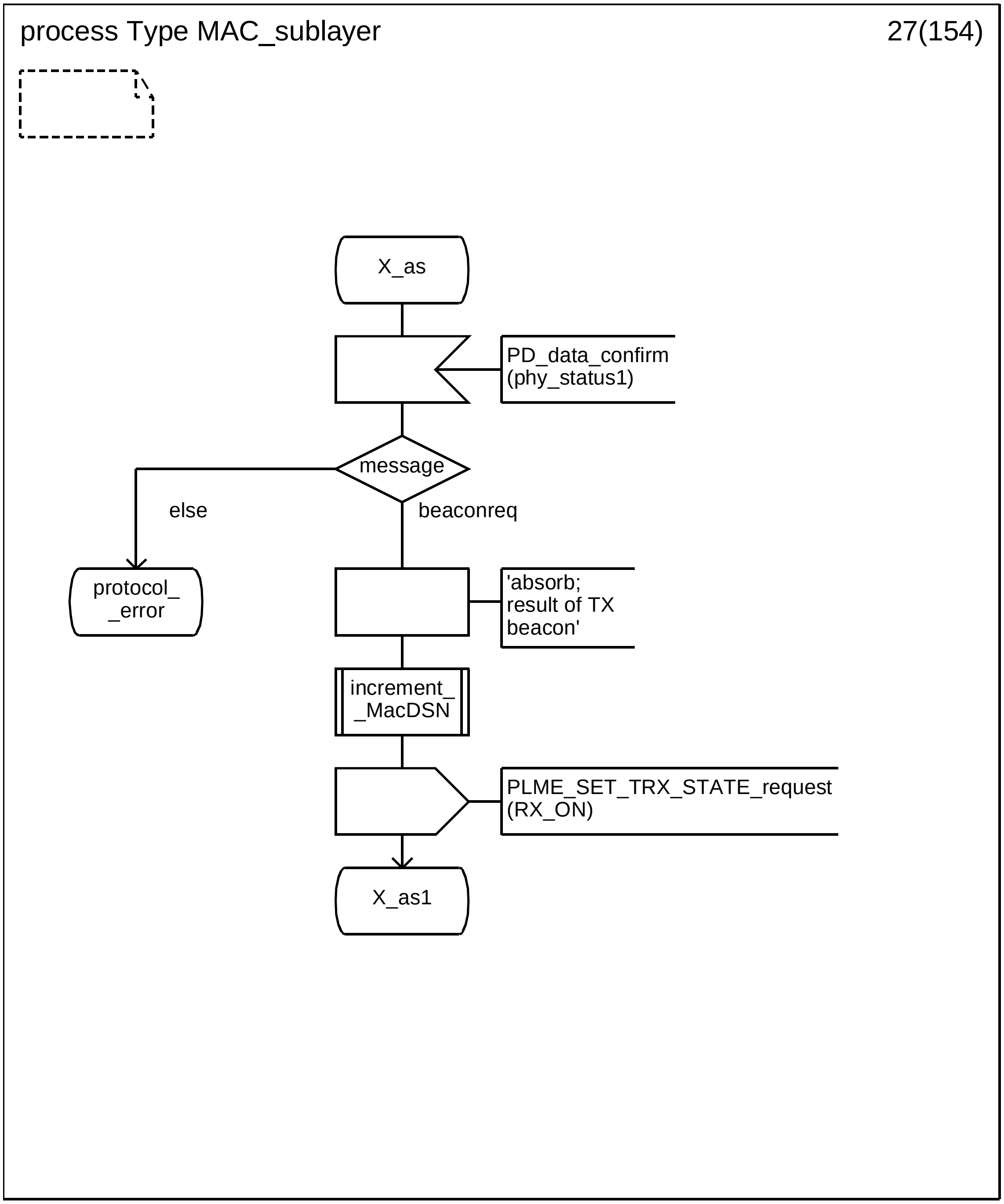
D.3.1.26 Process type MAC_sublayer (26)

.


process Type MAC_sublayer
26(154)
X_as_t
findabeacon
False
True
statM := no_beacon;
statM := success;
MLME_SCAN_confirm (statM,scantype, unscanned, resultlistsize,null_edl,PANDescriptorset)
PLME_SET_TRX_STATE_request (TRX_OFF)
X_as2

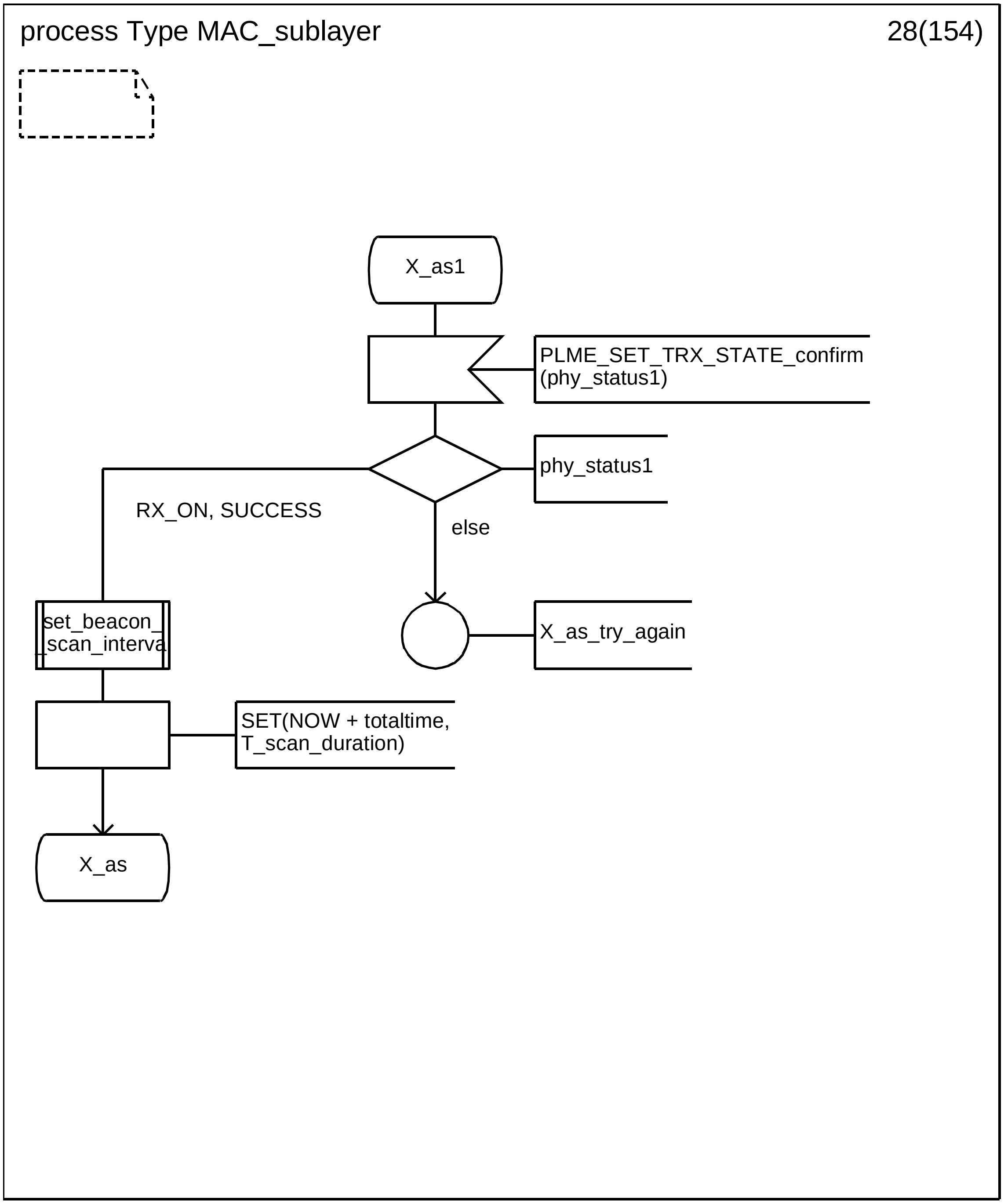
D.3.1.27 Process type MAC_sublayer (27)

.



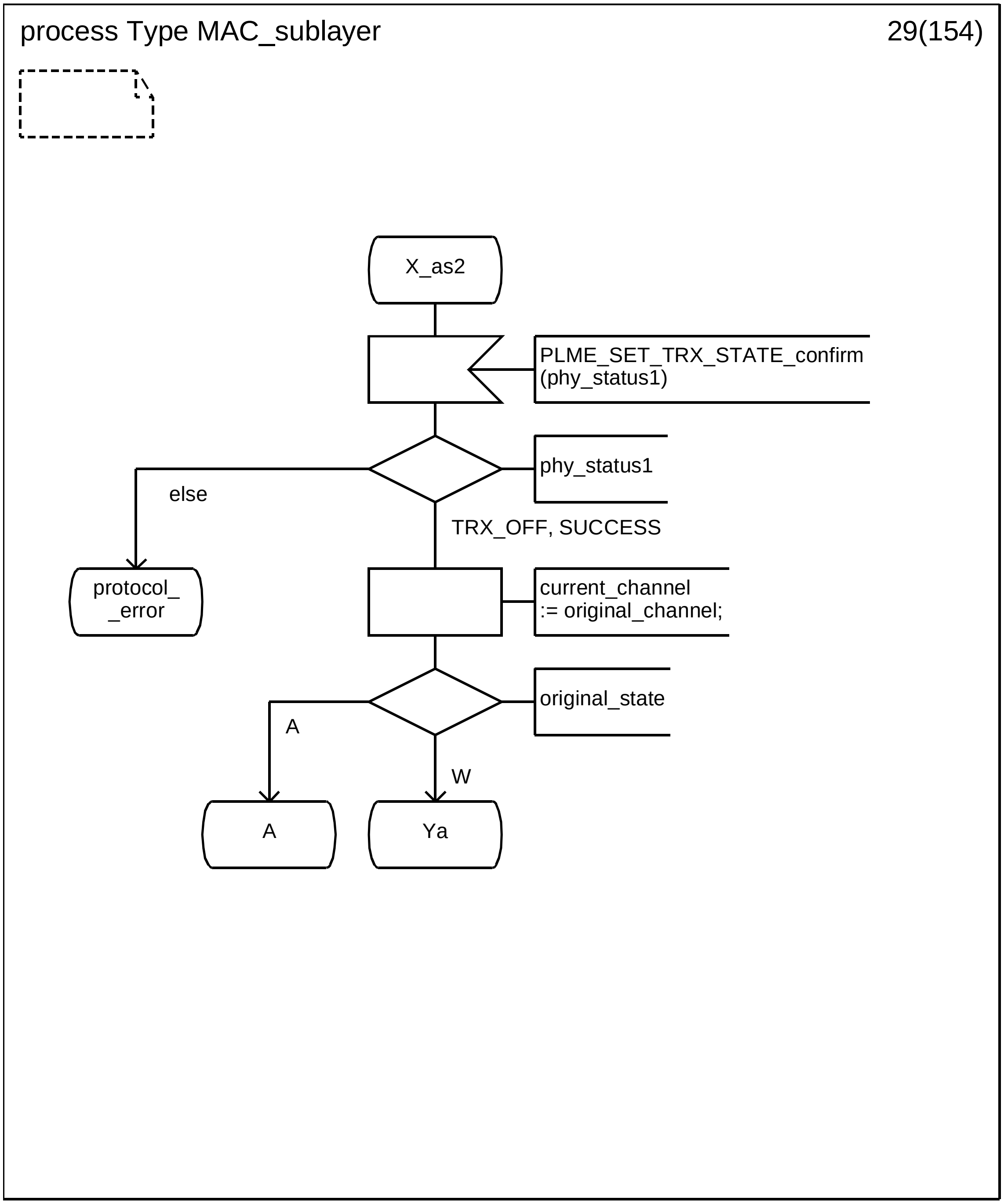
D.3.1.28 Process type MAC_sublayer (28)

.



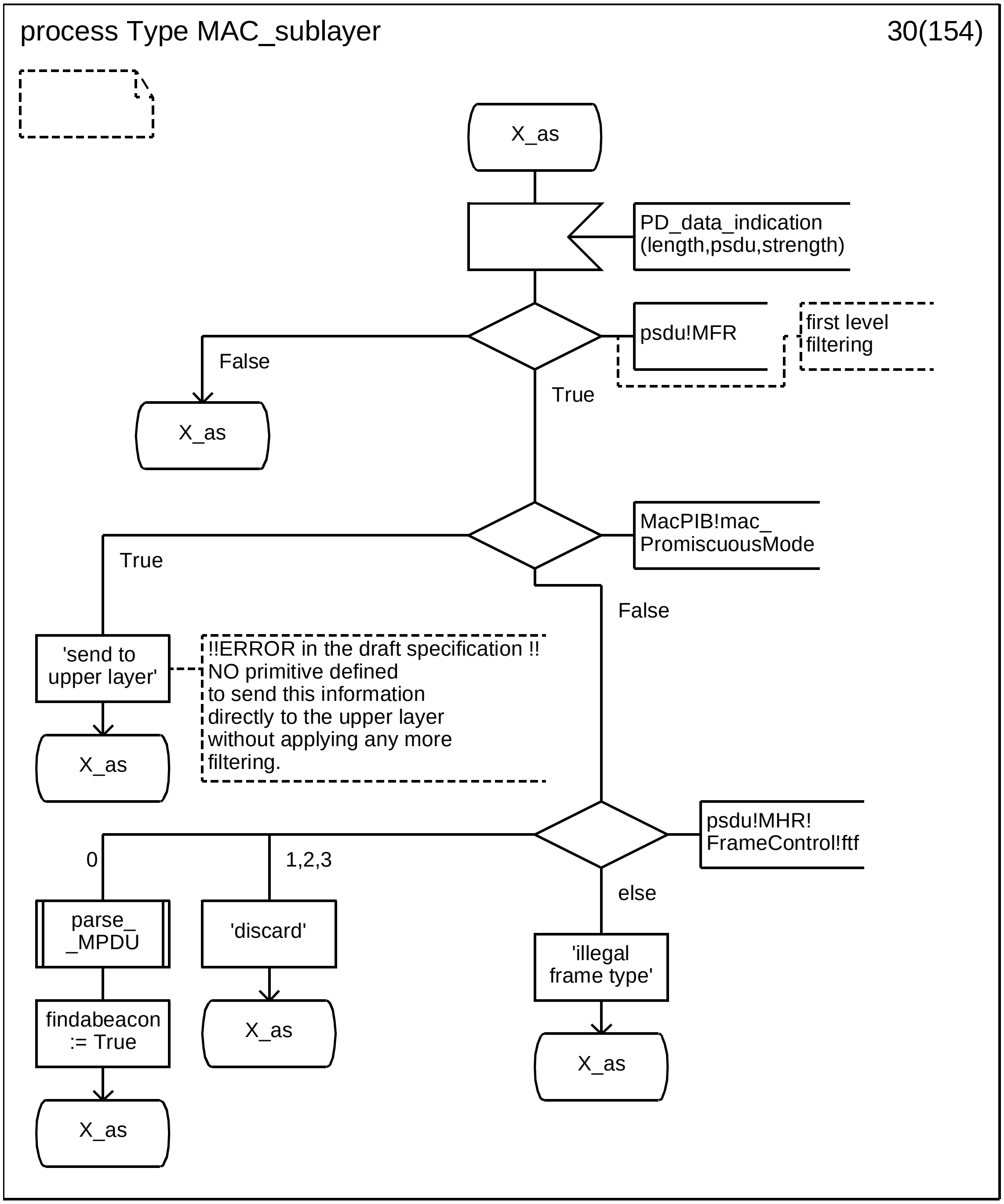
D.3.1.29 Process type MAC_sublayer (29)

.



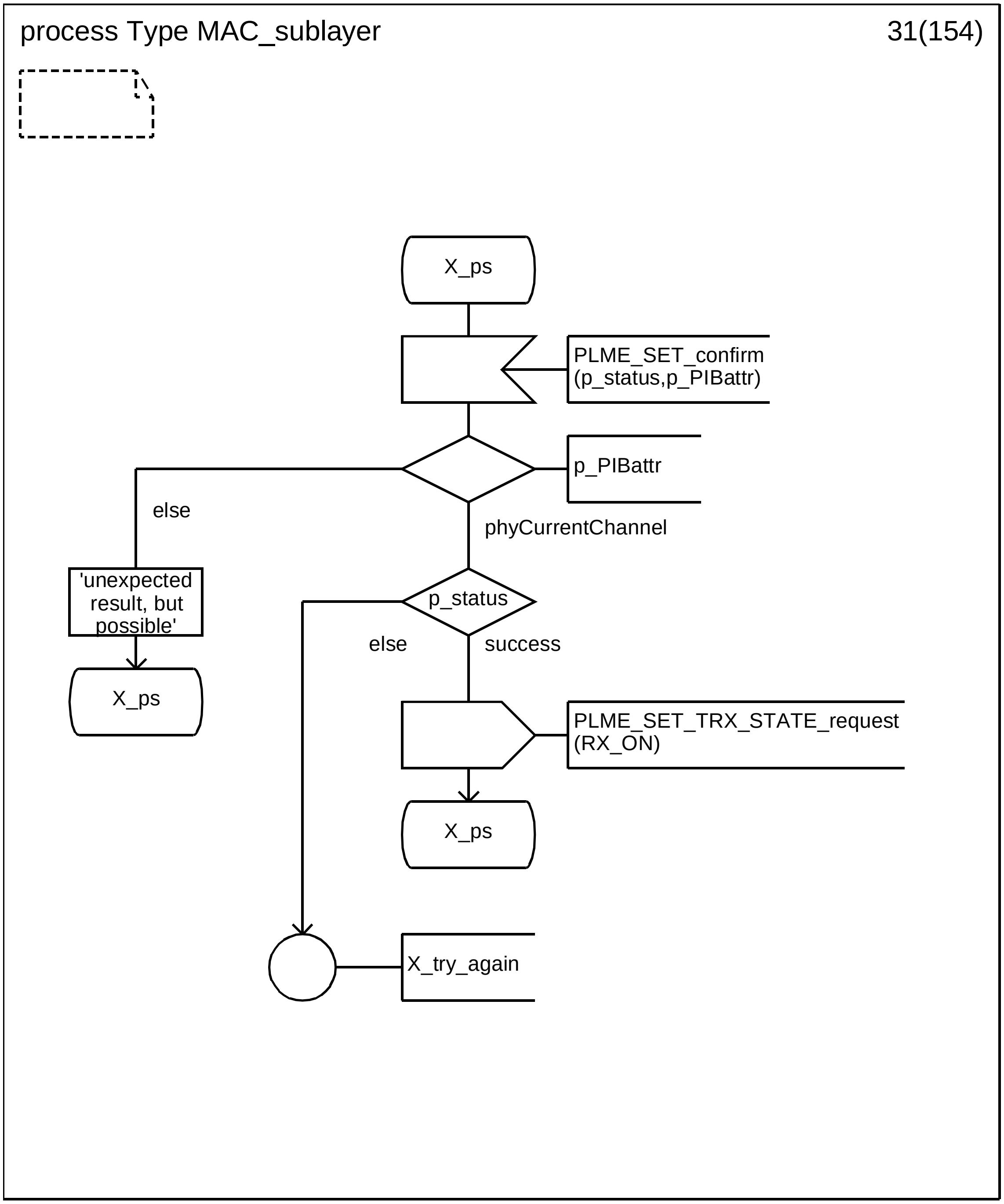
D.3.1.30 Process type MAC_sublayer (30)

.



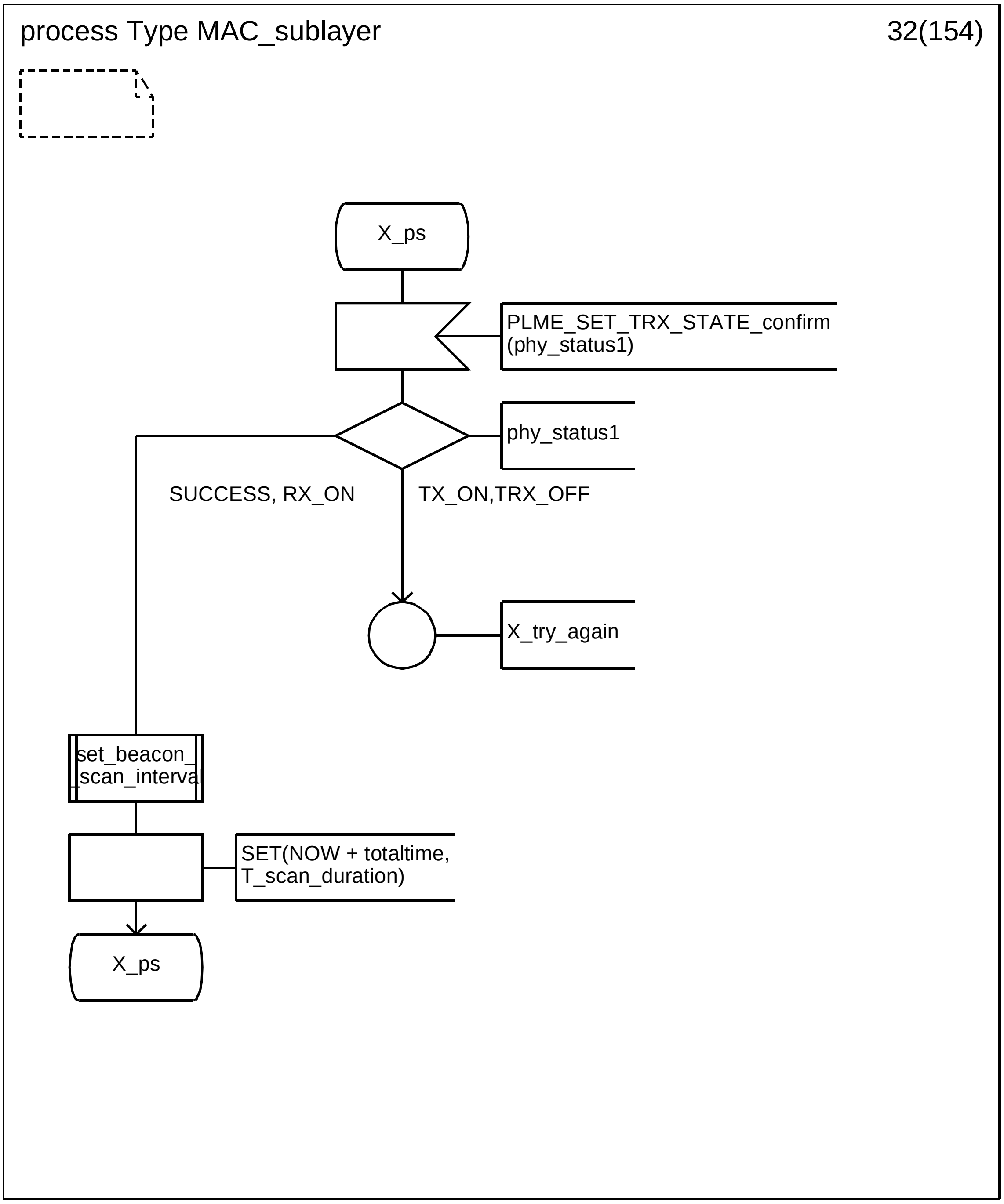
D.3.1.31 Process type MAC_sublayer (31)

.



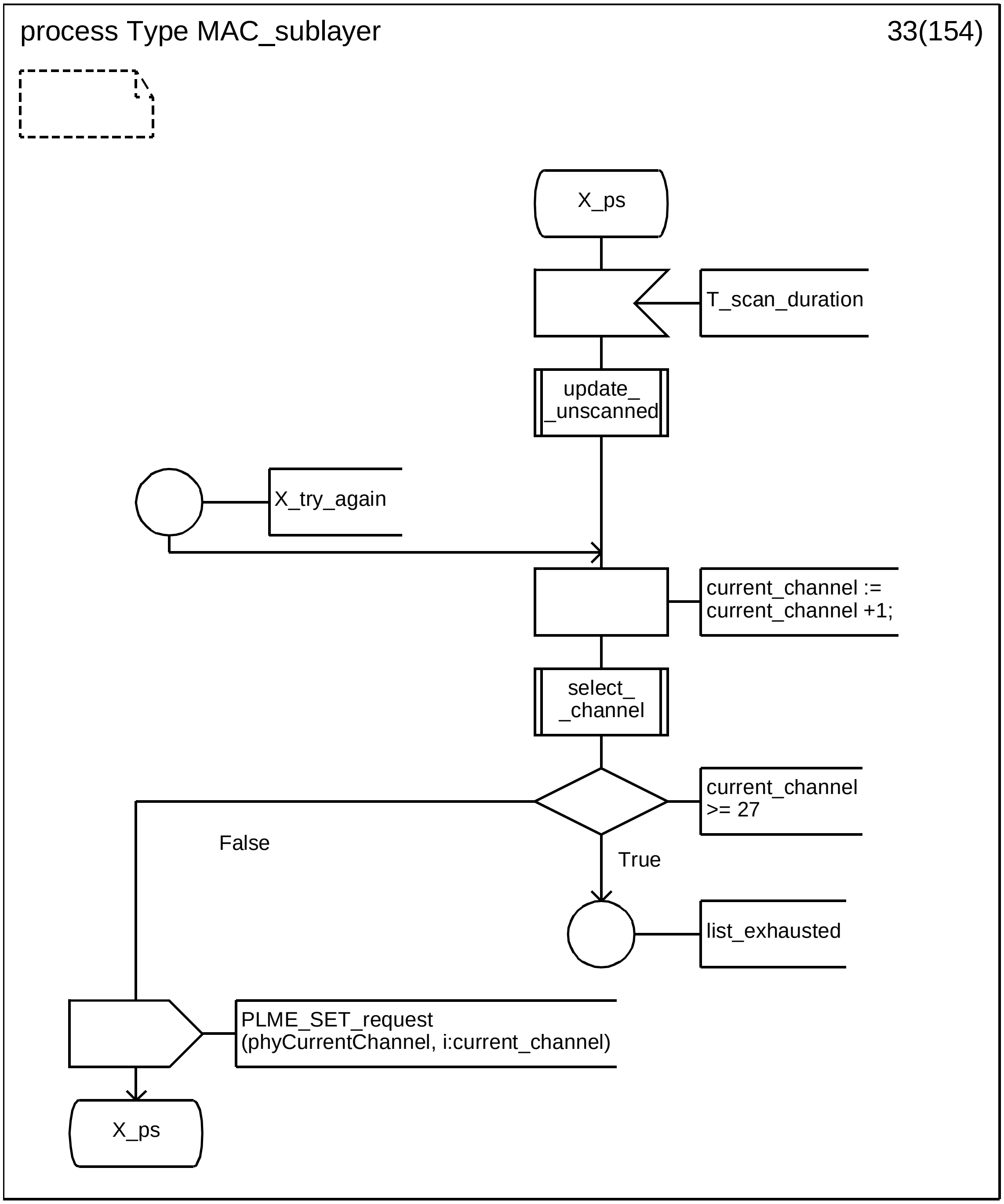
D.3.1.32 Process type MAC_sublayer (32)

.

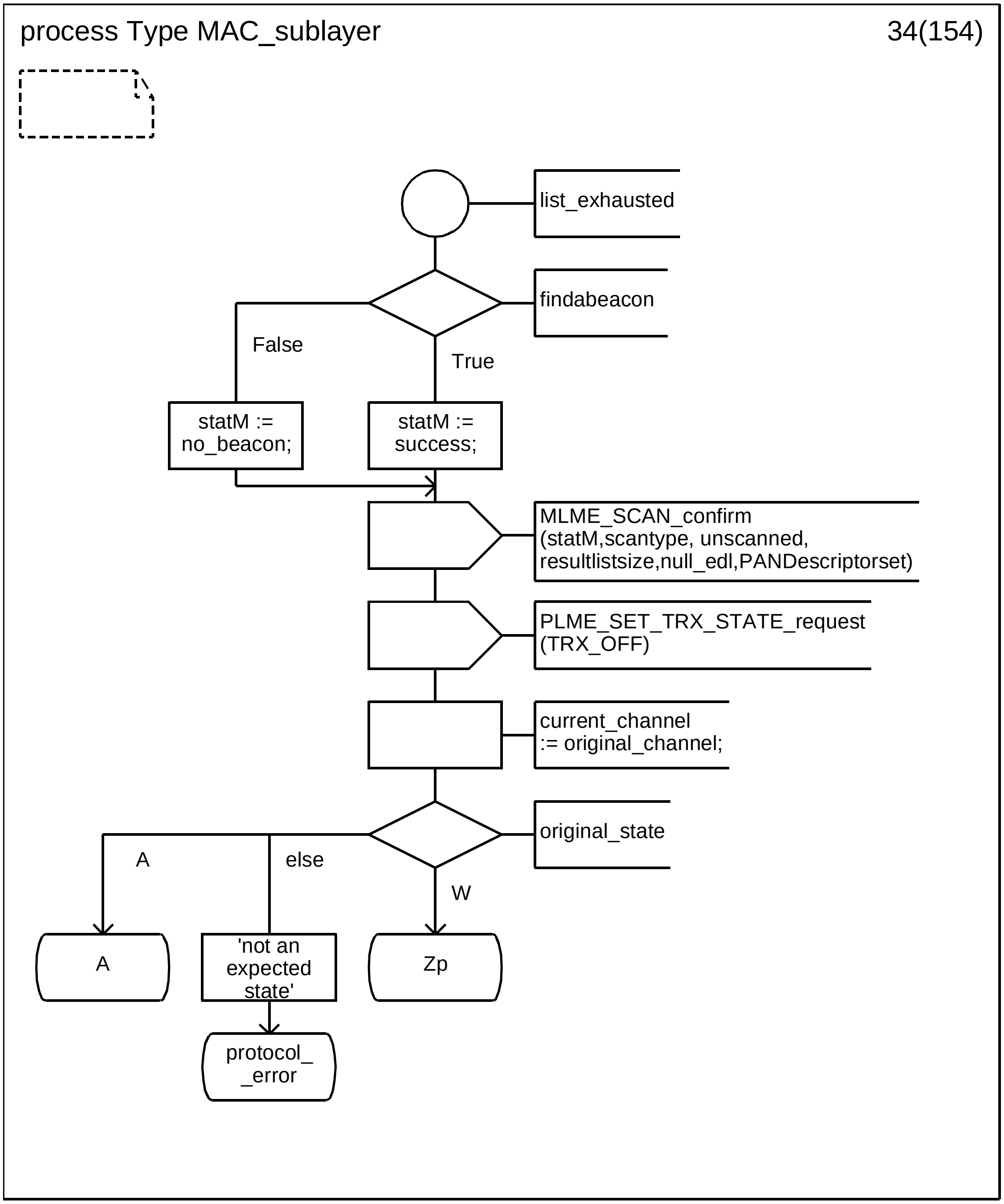


D.3.1.33 Process type MAC_sublayer (33)

.

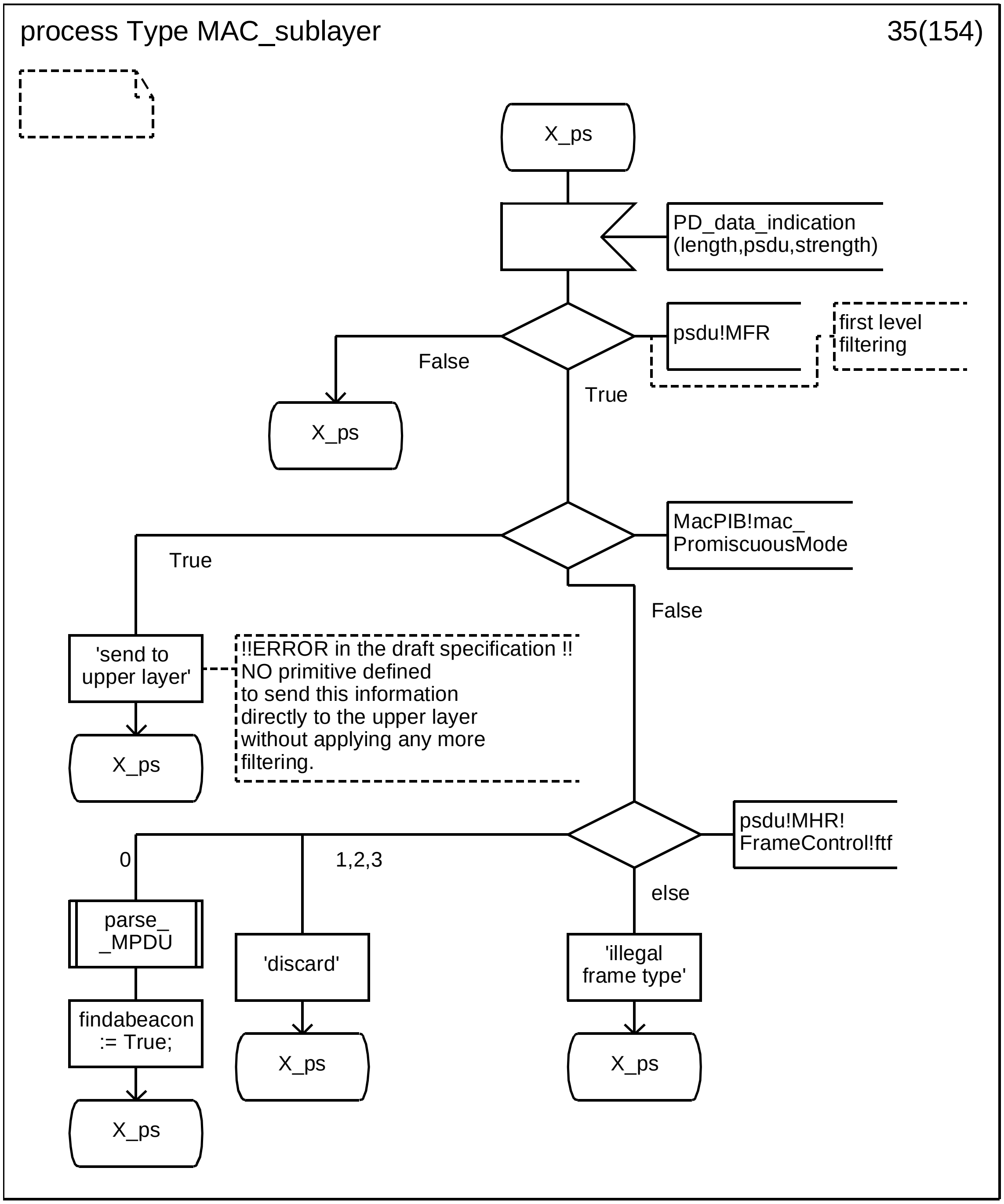


D.3.1.34 Process type MAC_sublayer (34)



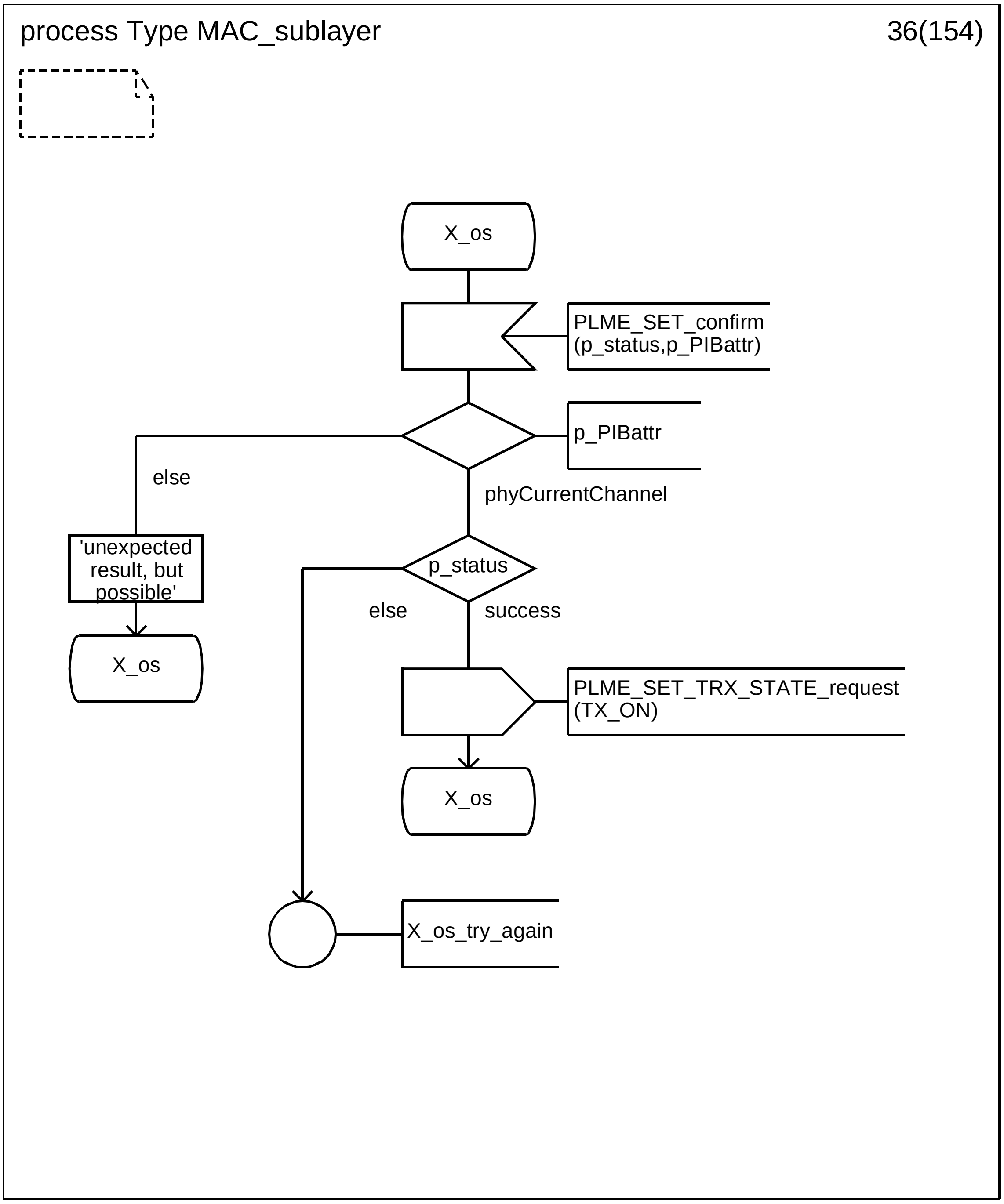
D.3.1.35 Process type MAC_sublayer (35)

.

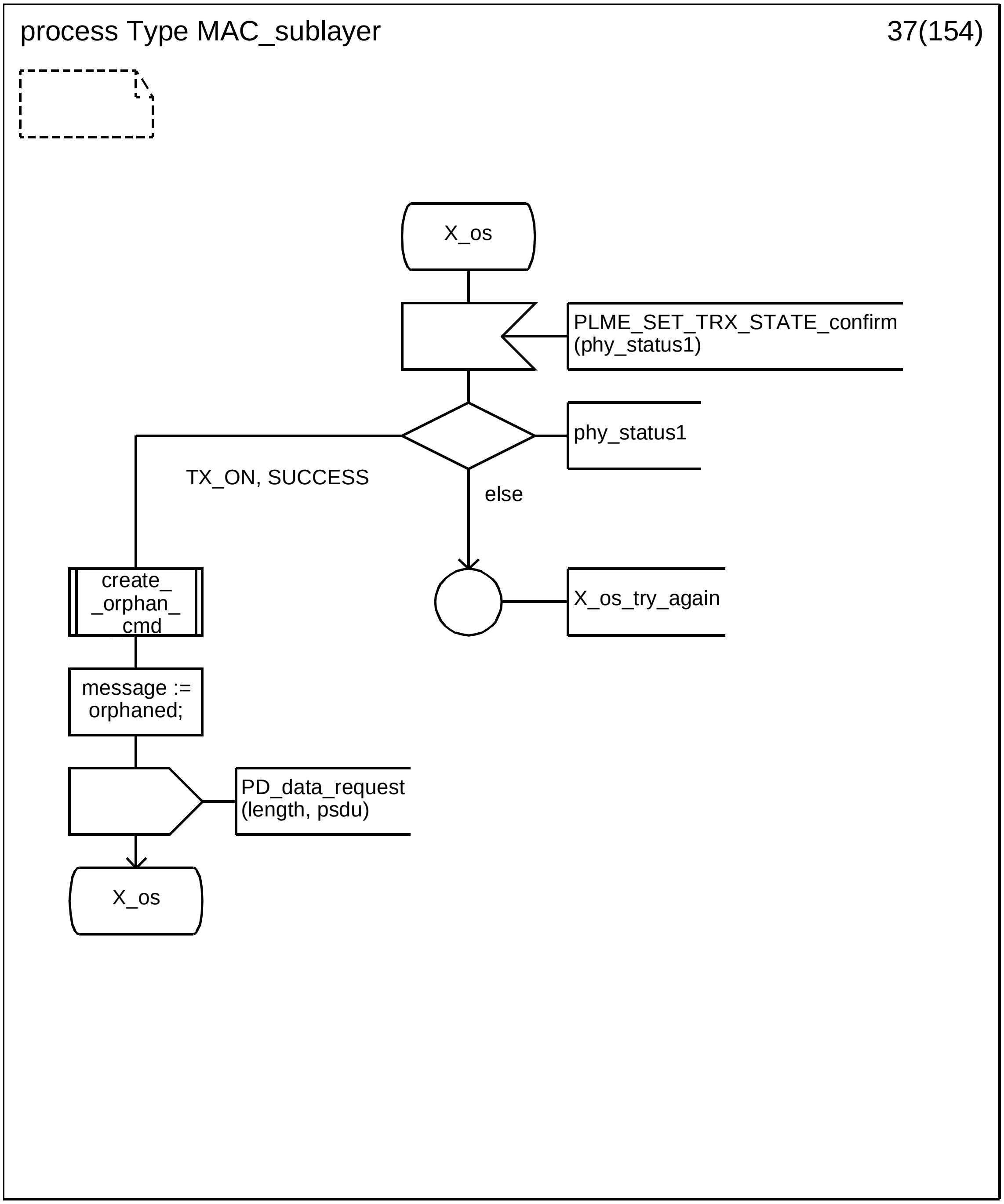


D.3.1.36 Process type MAC_sublayer (36)

.

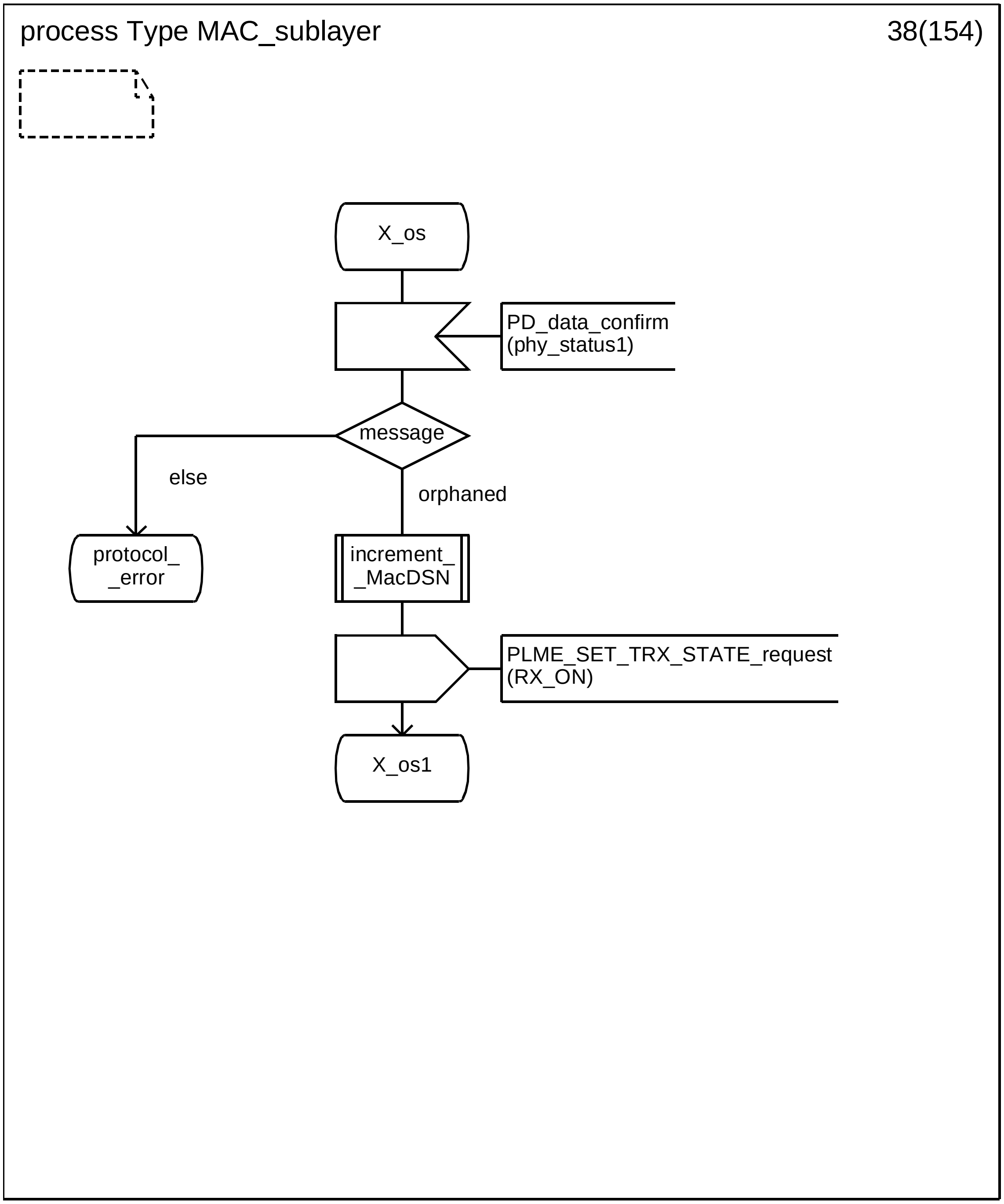


D.3.1.37 Process type MAC_sublayer (37)



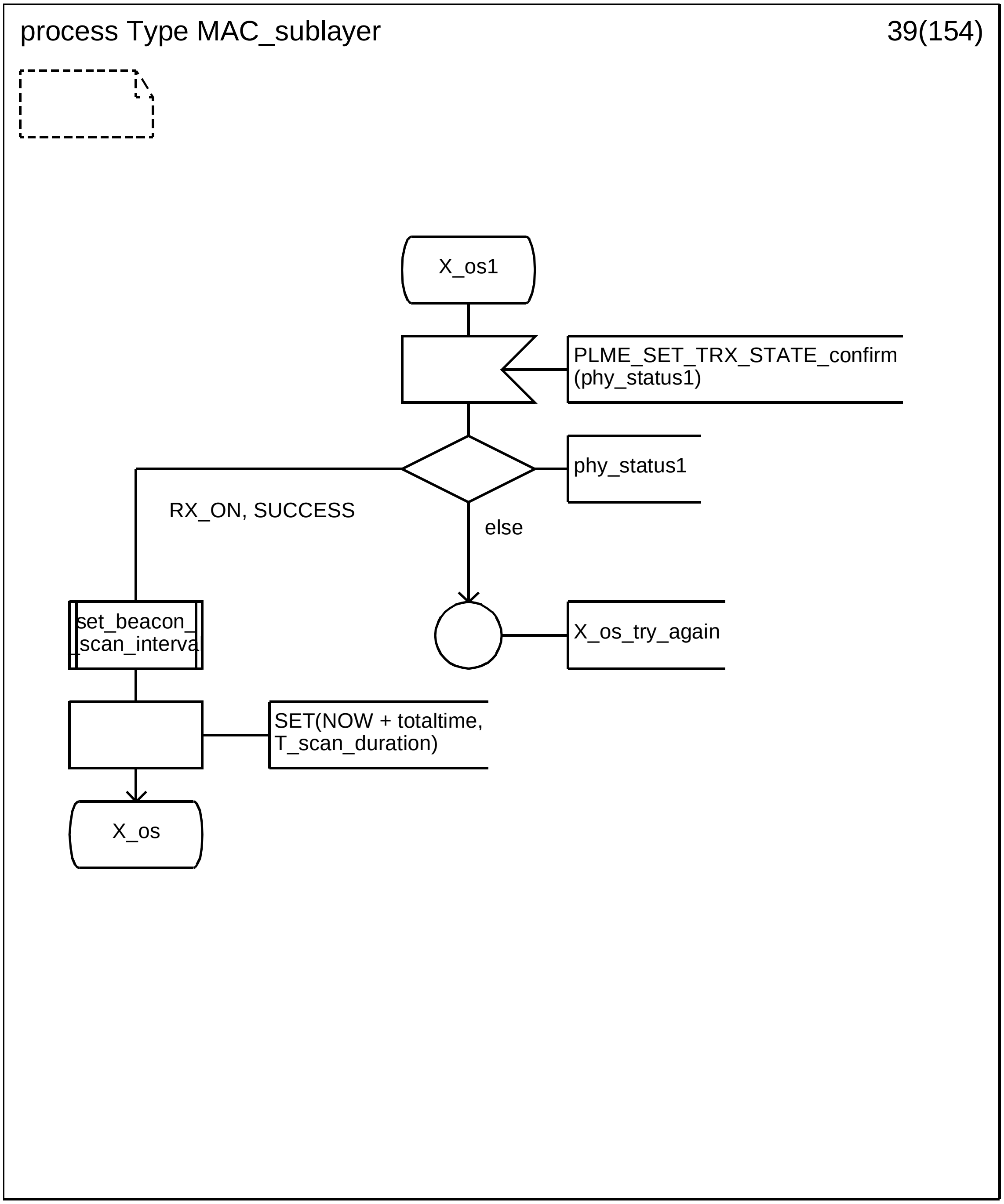
D.3.1.38 Process type MAC_sublayer (38)

.



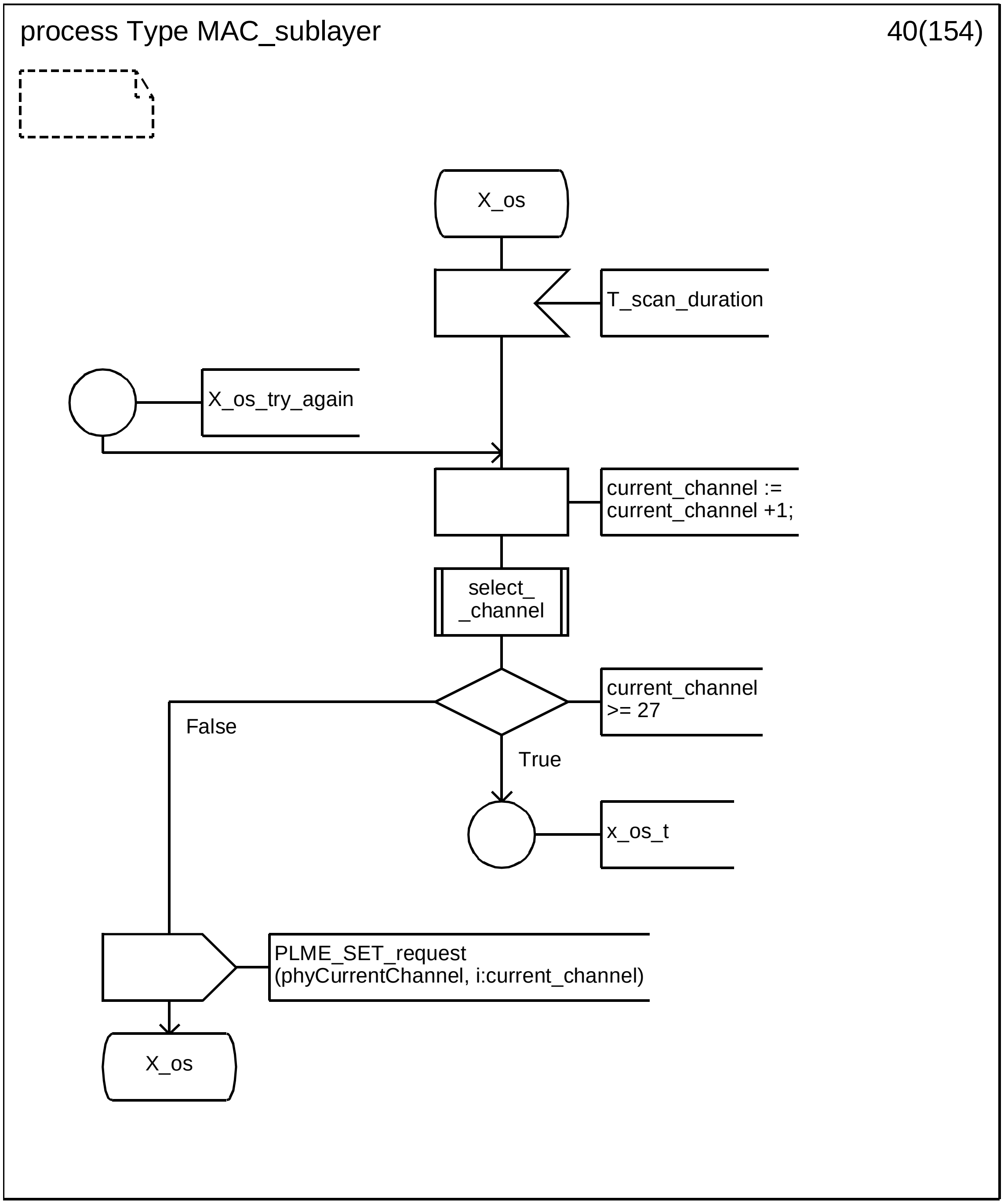
D.3.1.39 Process type MAC_sublayer (39)

.



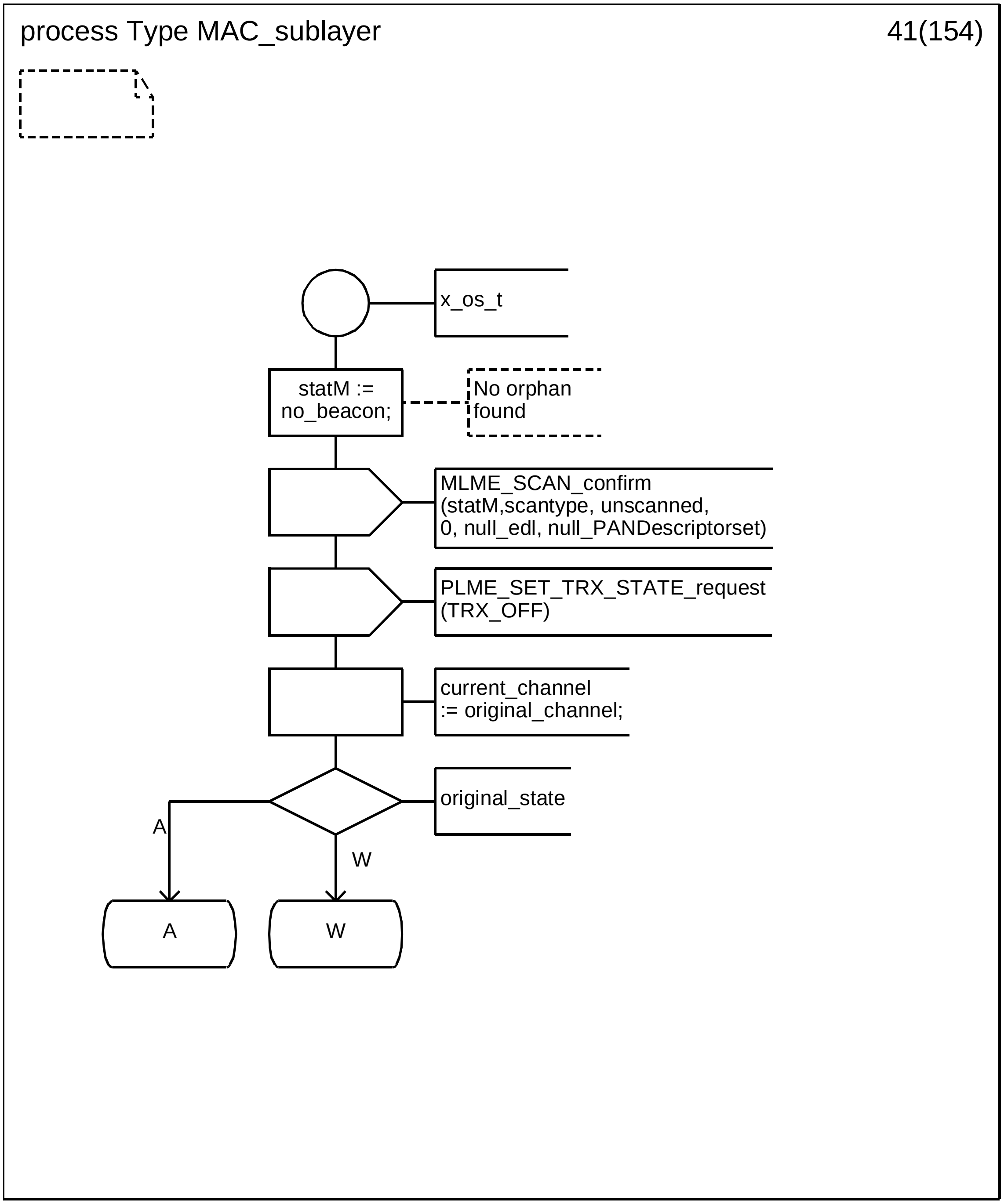
D.3.1.40 Process type MAC_sublayer (40)

.

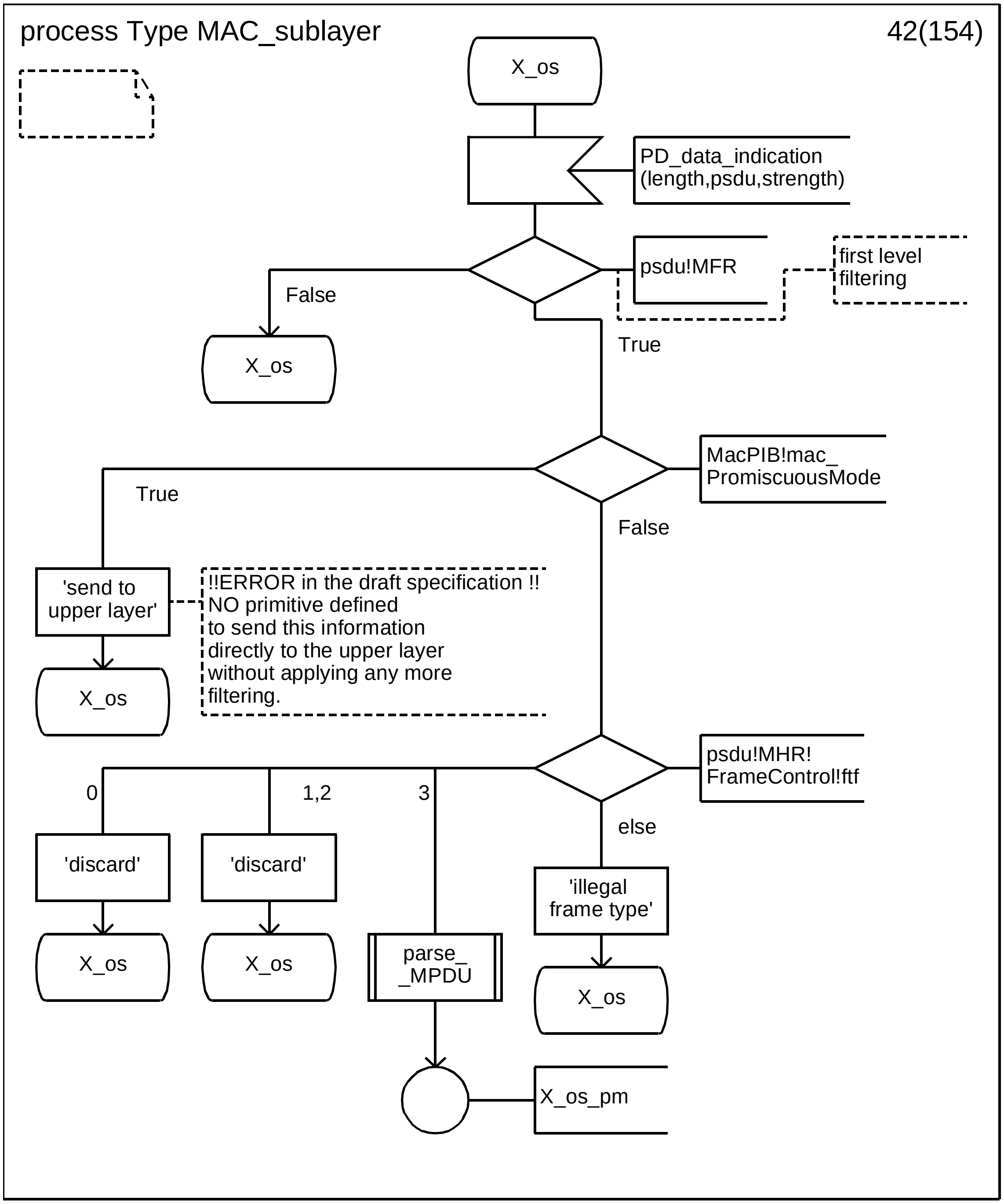


D.3.1.41 Process type MAC_sublayer (41)

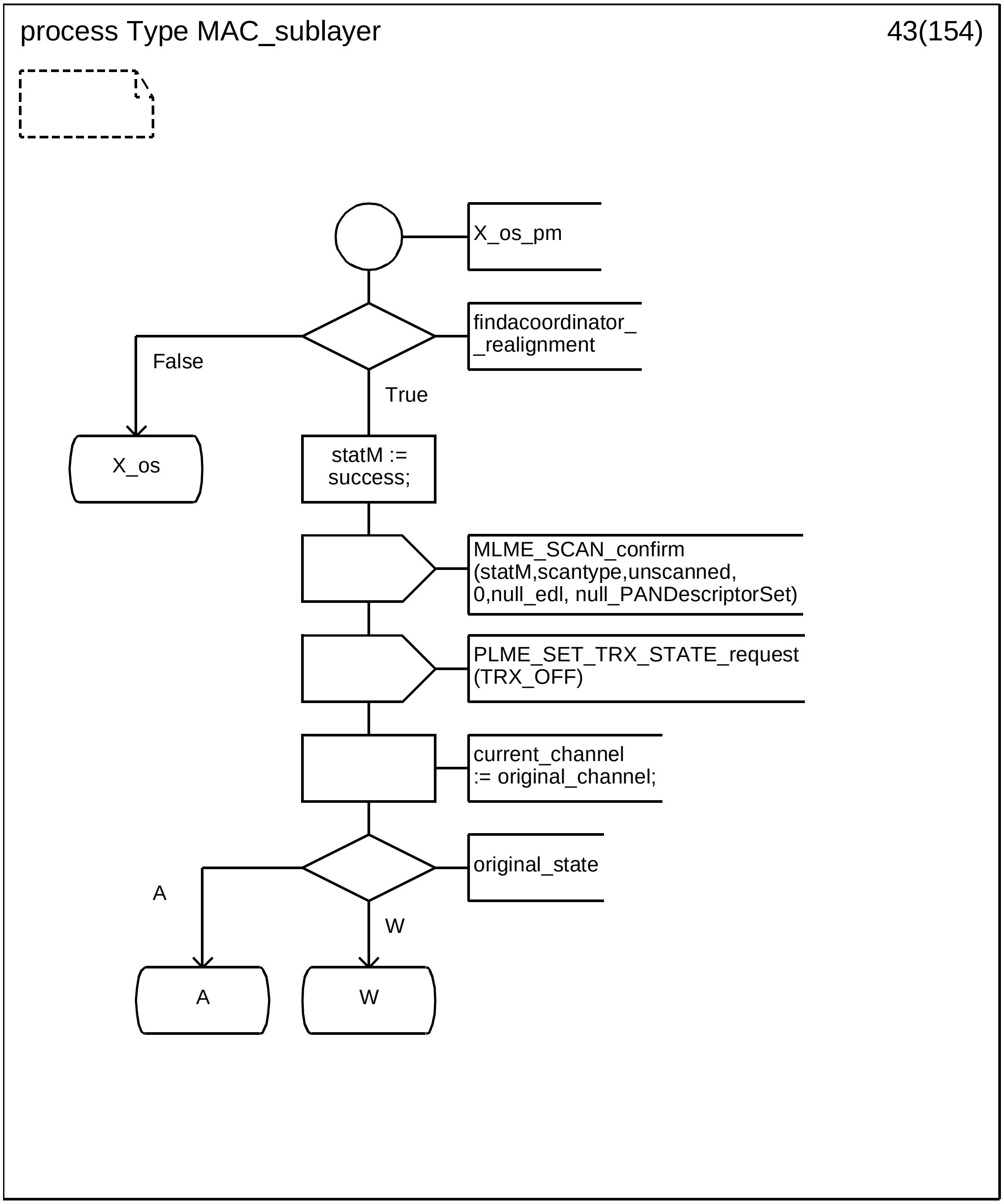
.



D.3.1.42 Process type MAC_sublayer (42)

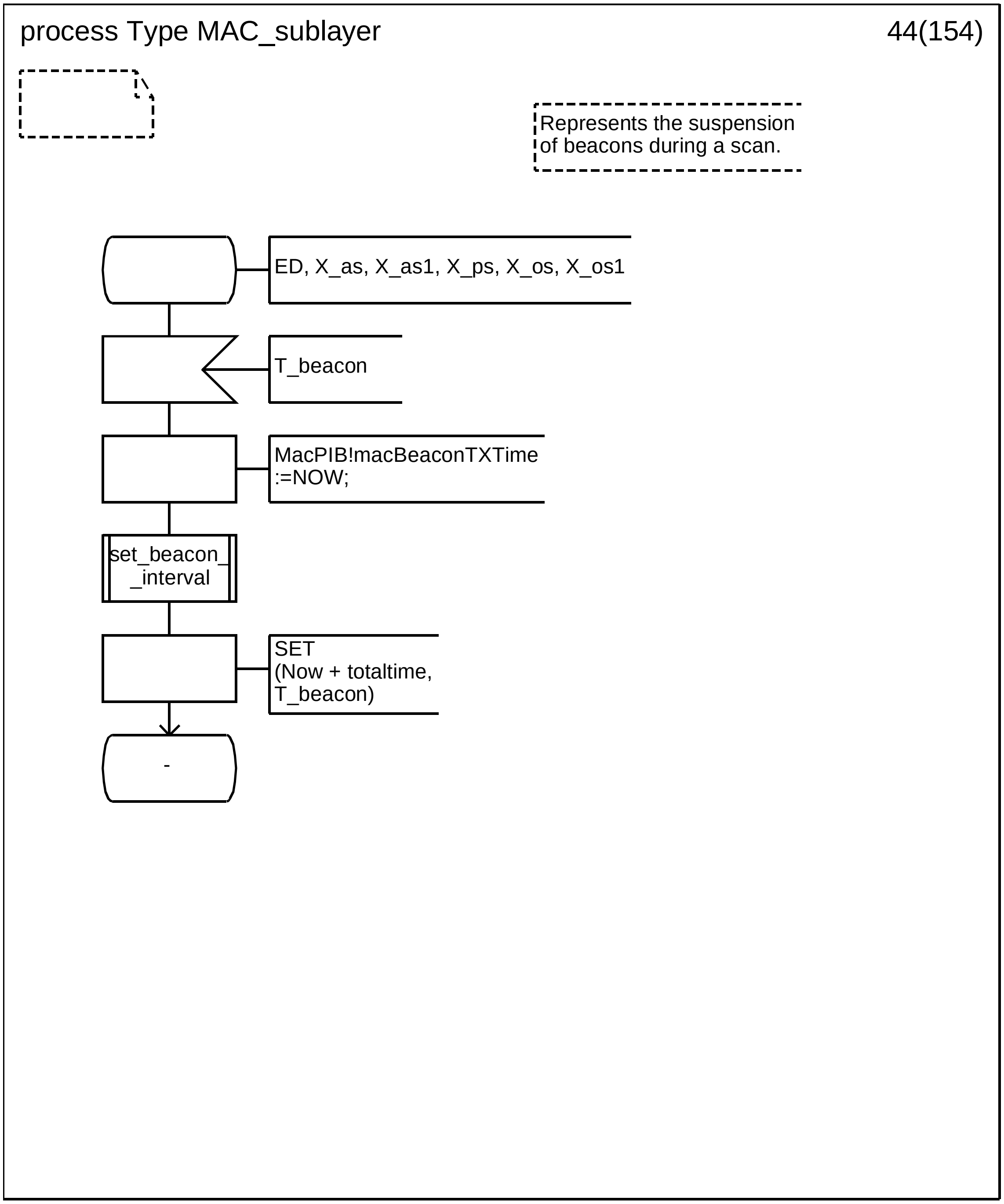


D.3.1.43 Process type MAC_sublayer (43)



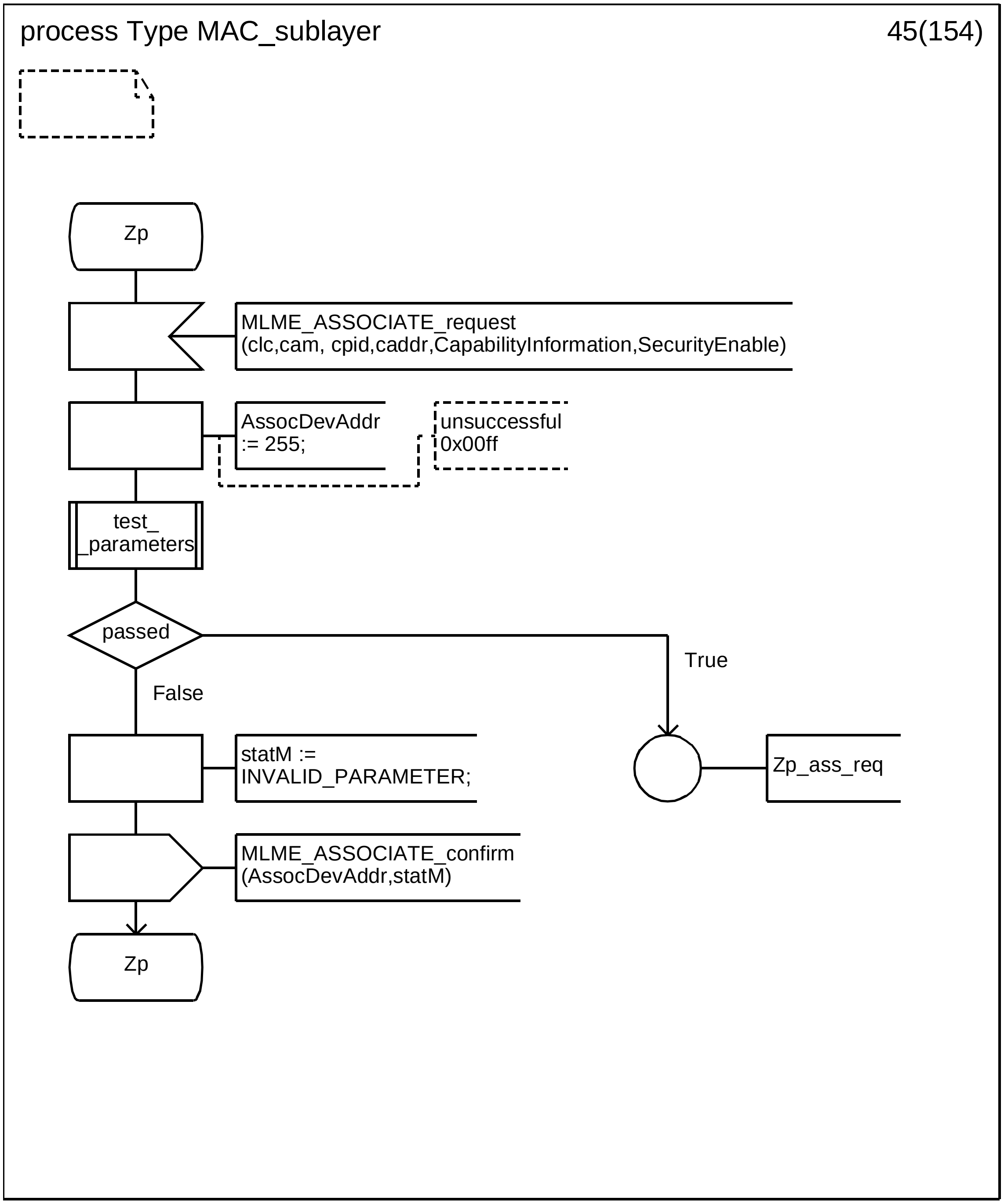
D.3.1.44 Process type MAC_sublayer (44)

.


process Type MAC_sublayer
44(154)
Represents the suspension of beacons during a scan.
ED, X_as, X_as1, X_ps, X_os, X_os1
T_beacon
MacPIB!macBeaconTXTime :=NOW;
set_beacon_ _interval
SET (Now + totaltime, T_beacon)
-

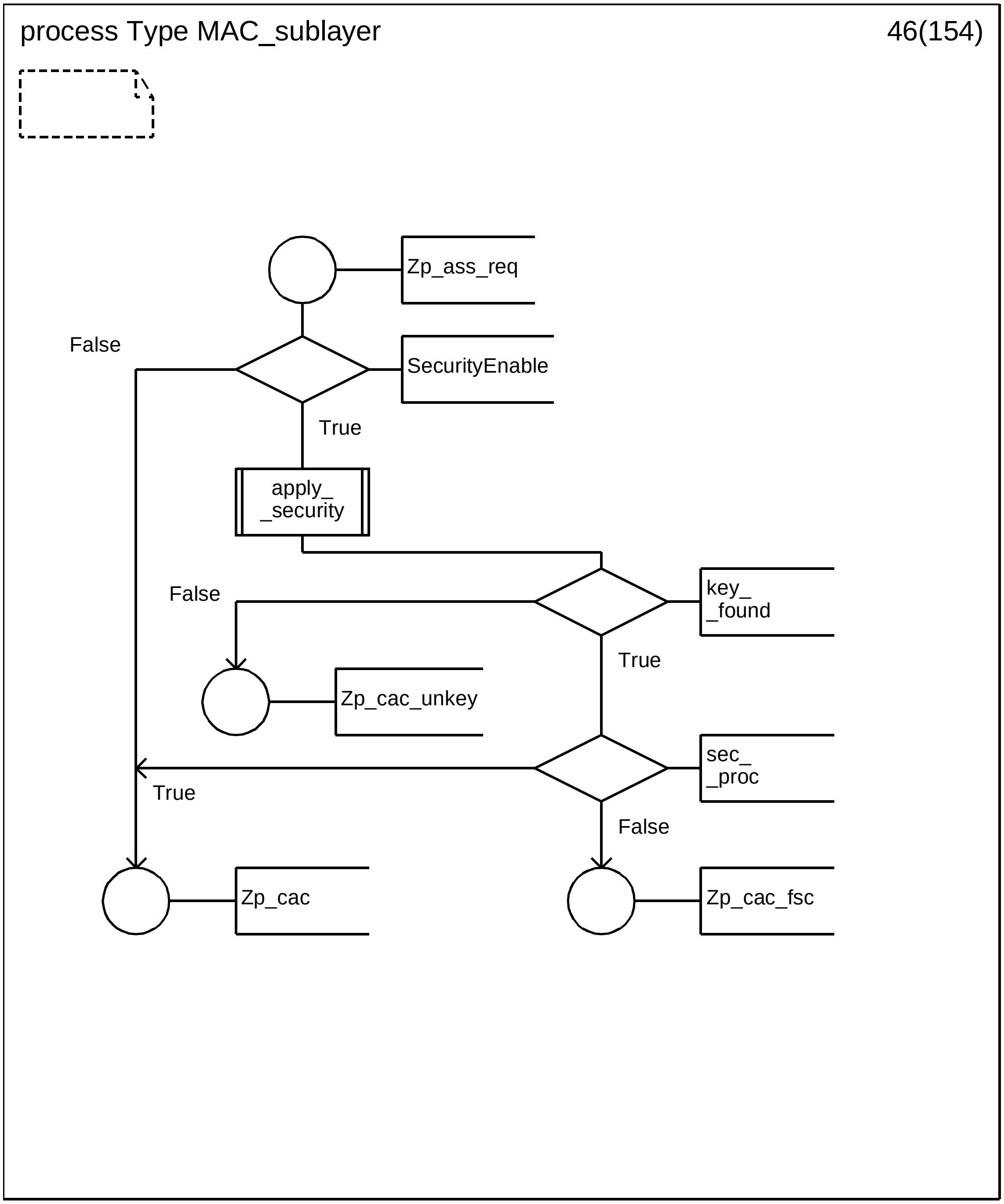
D.3.1.45 Process type MAC_sublayer (45)

.



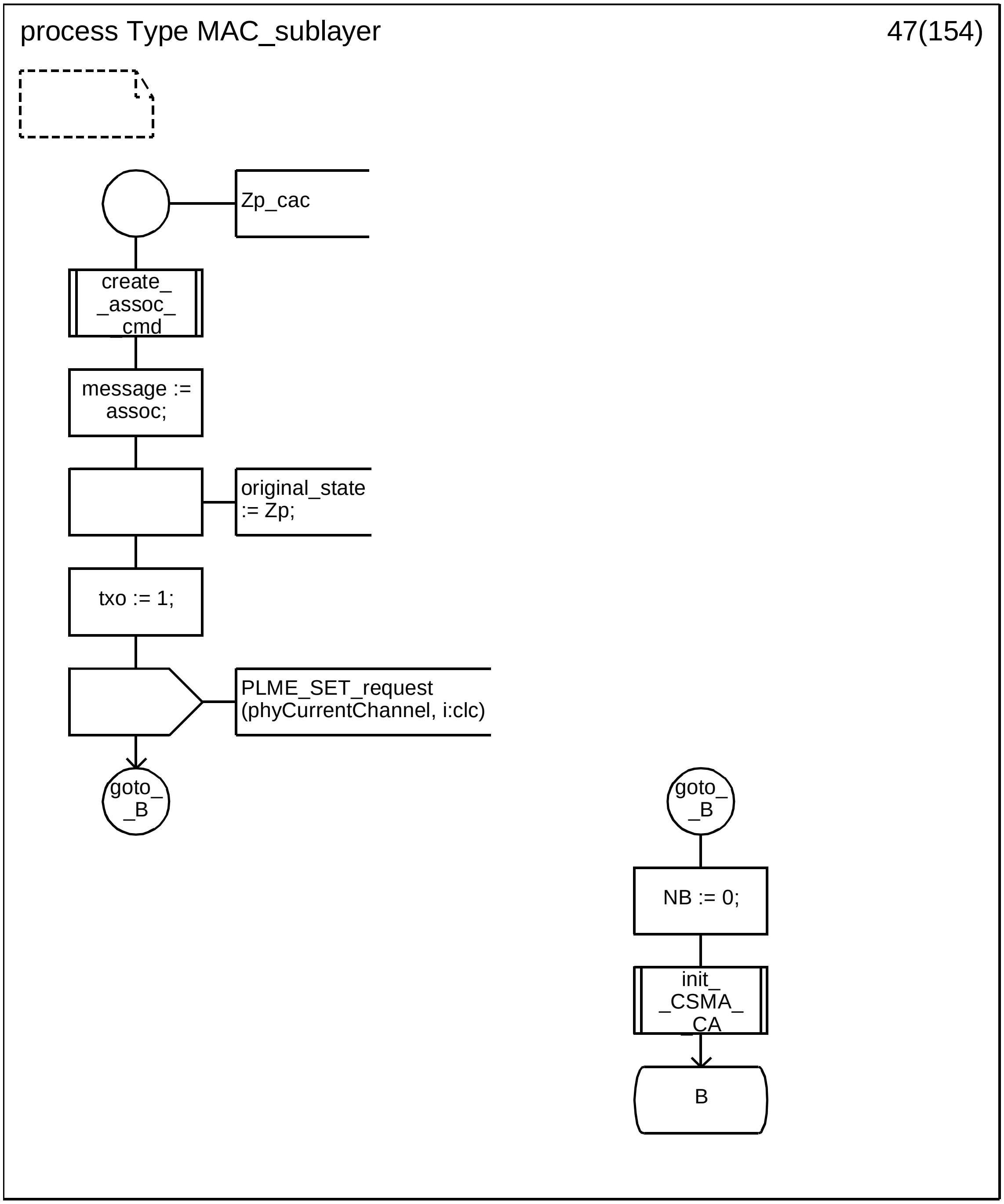
D.3.1.46 Process type MAC_sublayer (46)

.



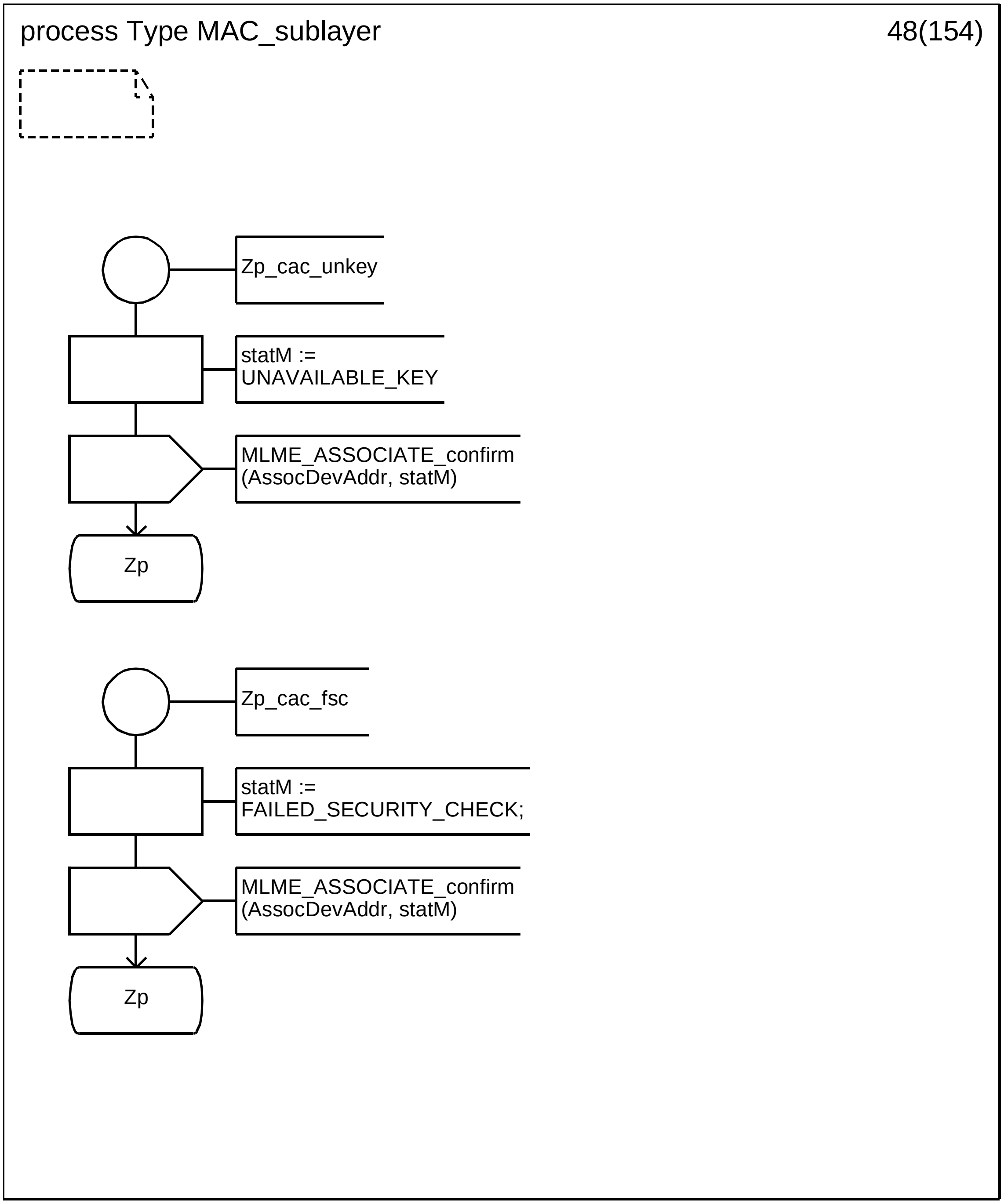
D.3.1.47 Process type MAC_sublayer (47)

.



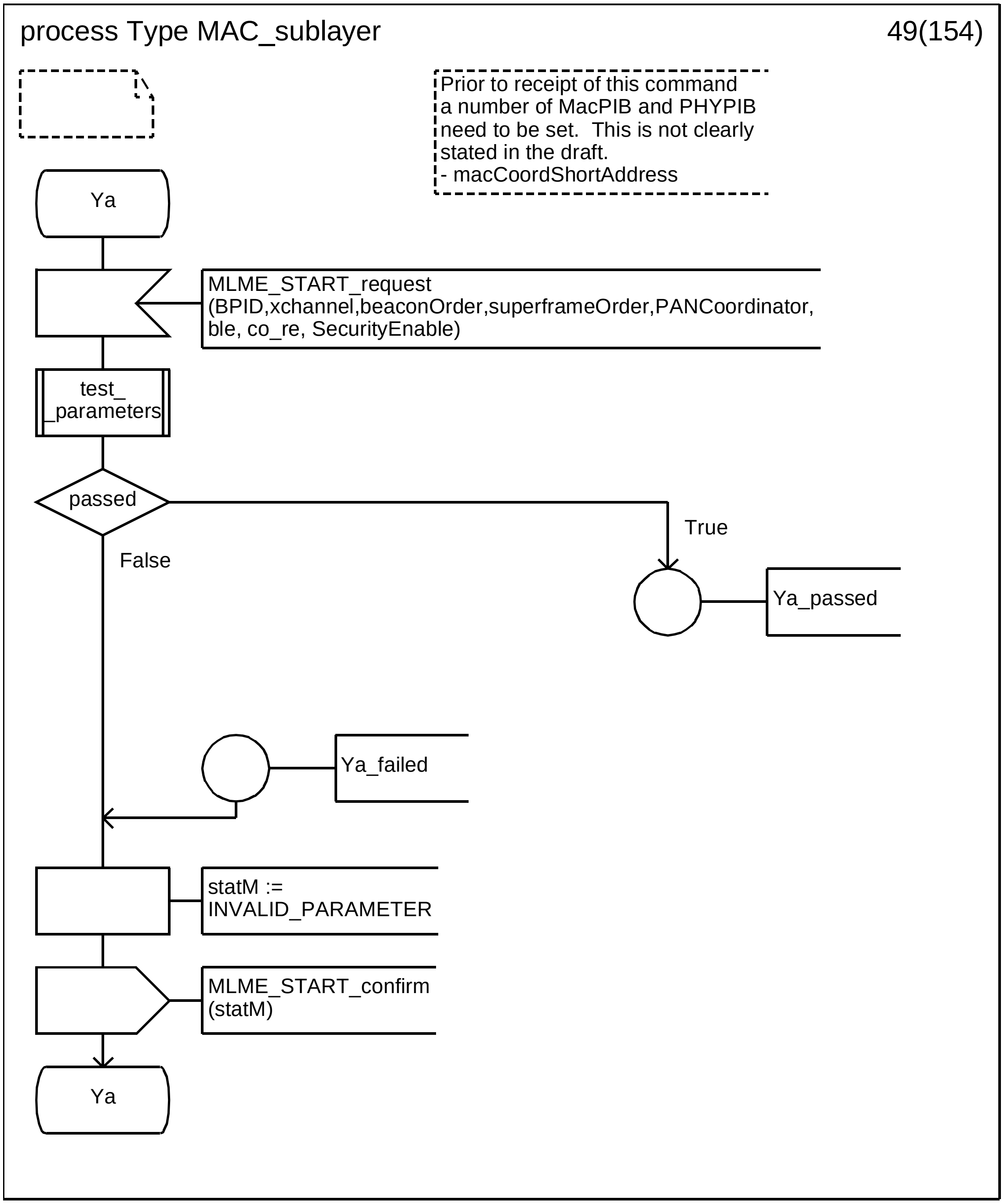
D.3.1.48 Process type MAC_sublayer (48)

.



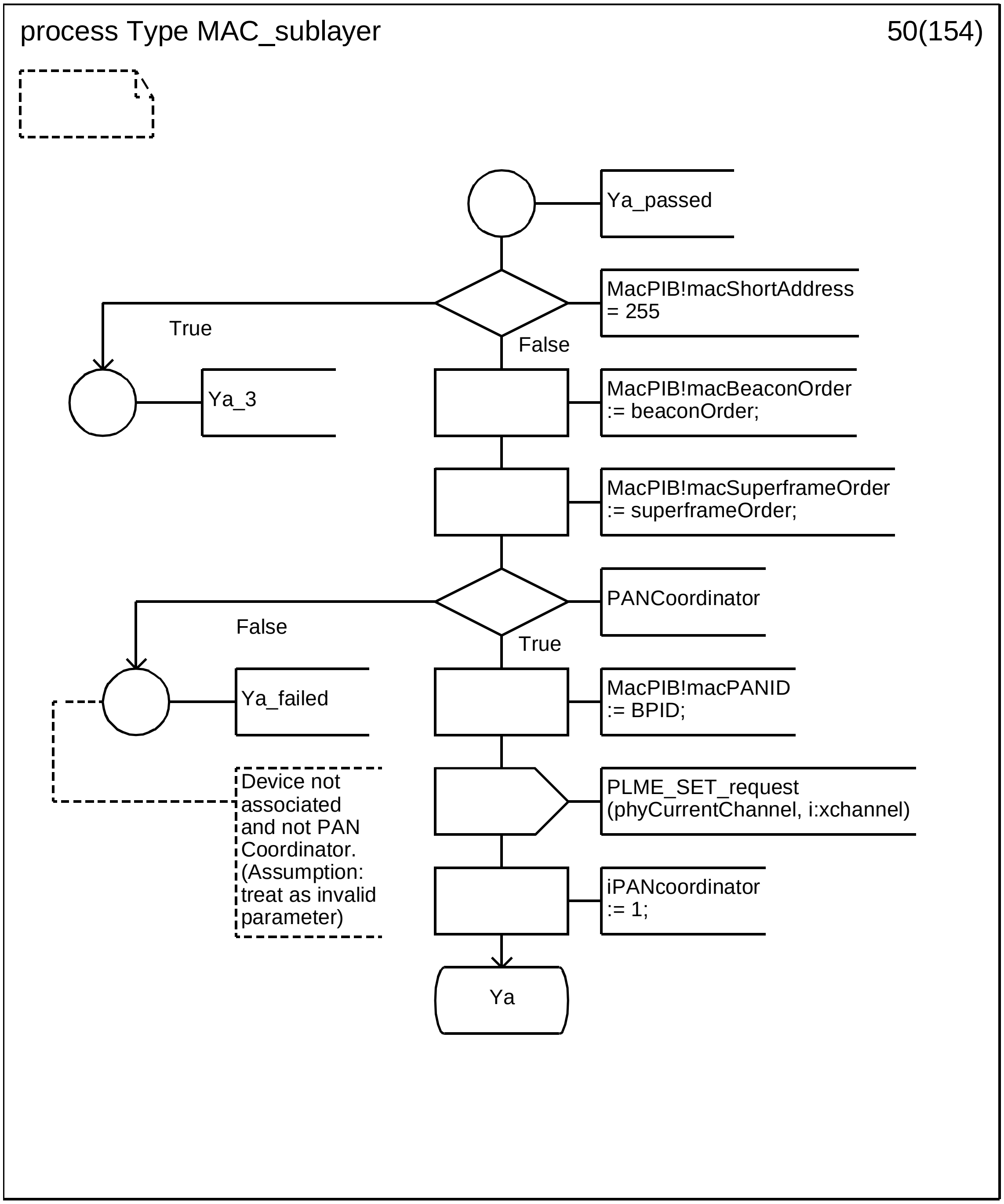
D.3.1.49 Process type MAC_sublayer (49)

.


process Type MAC_sublayer
49(154)
Prior to receipt of this command
a number of MacPIB and PHYPIB
need to be set. This is not clearly
stated in the draft.
- macCoordShortAddress
Ya
MLME_START_request
(BPID,xchannel,beaconOrder,superframeOrder,PANCoordinator,
ble, co_re, SecurityEnable)
test_
_parameters
passed
True
False
Ya_passed
Ya_failed
statM :=
INVALID_PARAMETER
MLME_START_confirm
(statM)
Ya

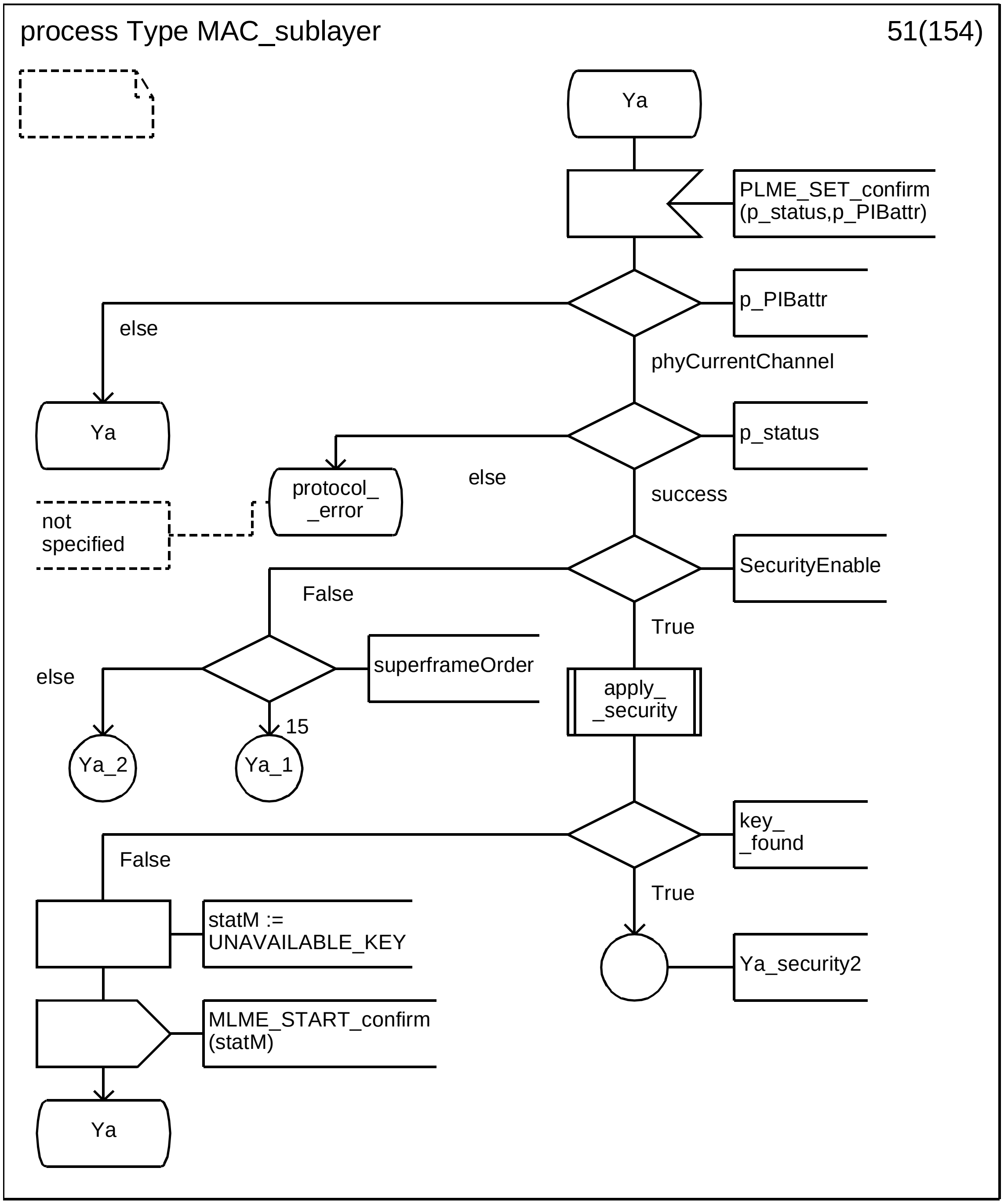
D.3.1.50 Process type MAC_sublayer (50)

.


process Type MAC_sublayer
50(154)
Ya_passed
MacPIB!macShortAddress = 255
True
False
Ya_3
MacPIB!macBeaconOrder := beaconOrder;
MacPIB!macSuperframeOrder := superframeOrder;
PANCoordinator
False
True
Ya_failed
MacPIB!macPANID := BPID;
Device not associated and not PAN Coordinator. (Assumption: treat as invalid parameter)
PLME_SET_request (phyCurrentChannel, i:xchannel)
iPANcoordinator := 1;
Ya

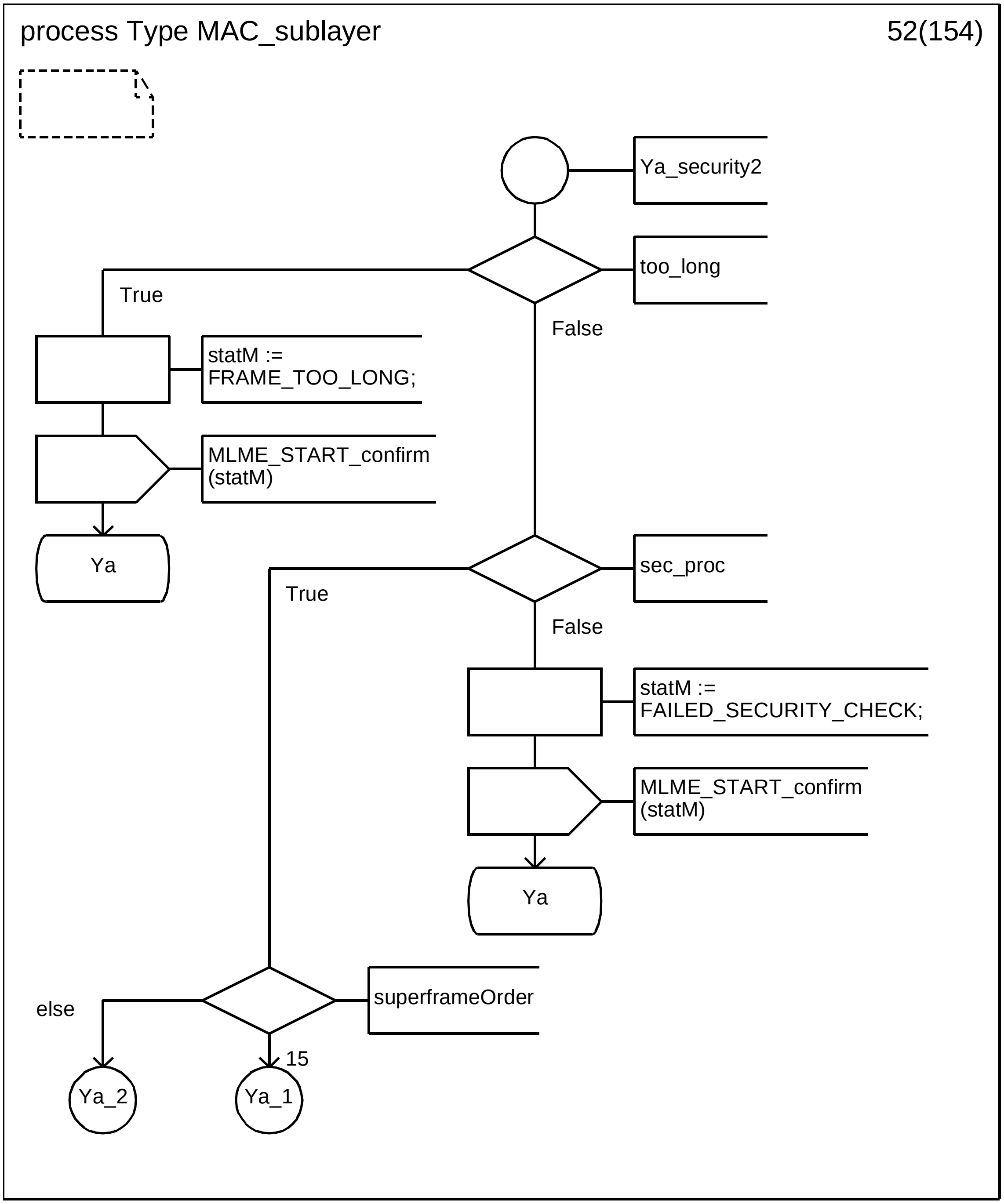
D.3.1.51 Process type MAC_sublayer (51)

.



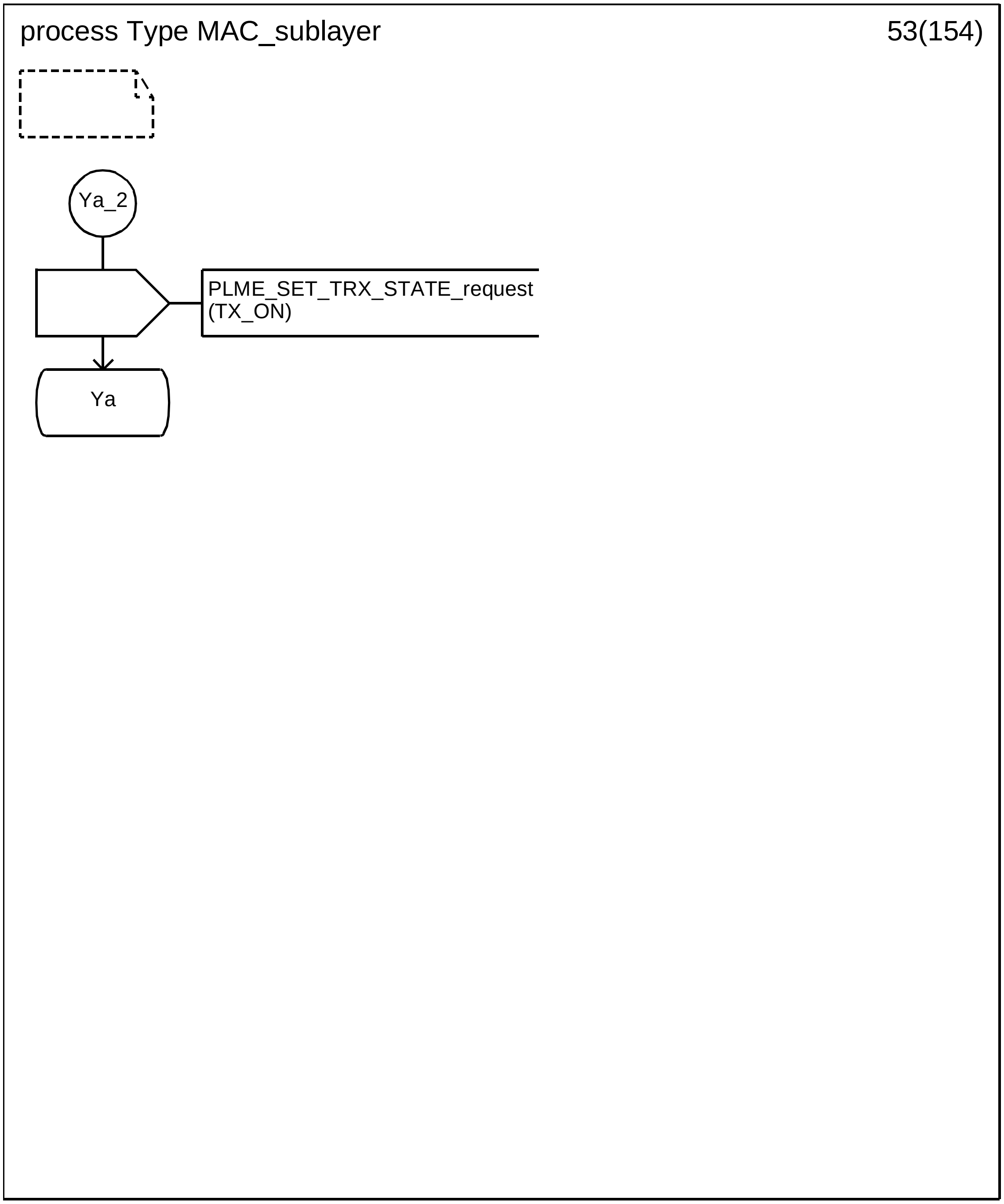
D.3.1.52 Process type MAC_sublayer (52)

.

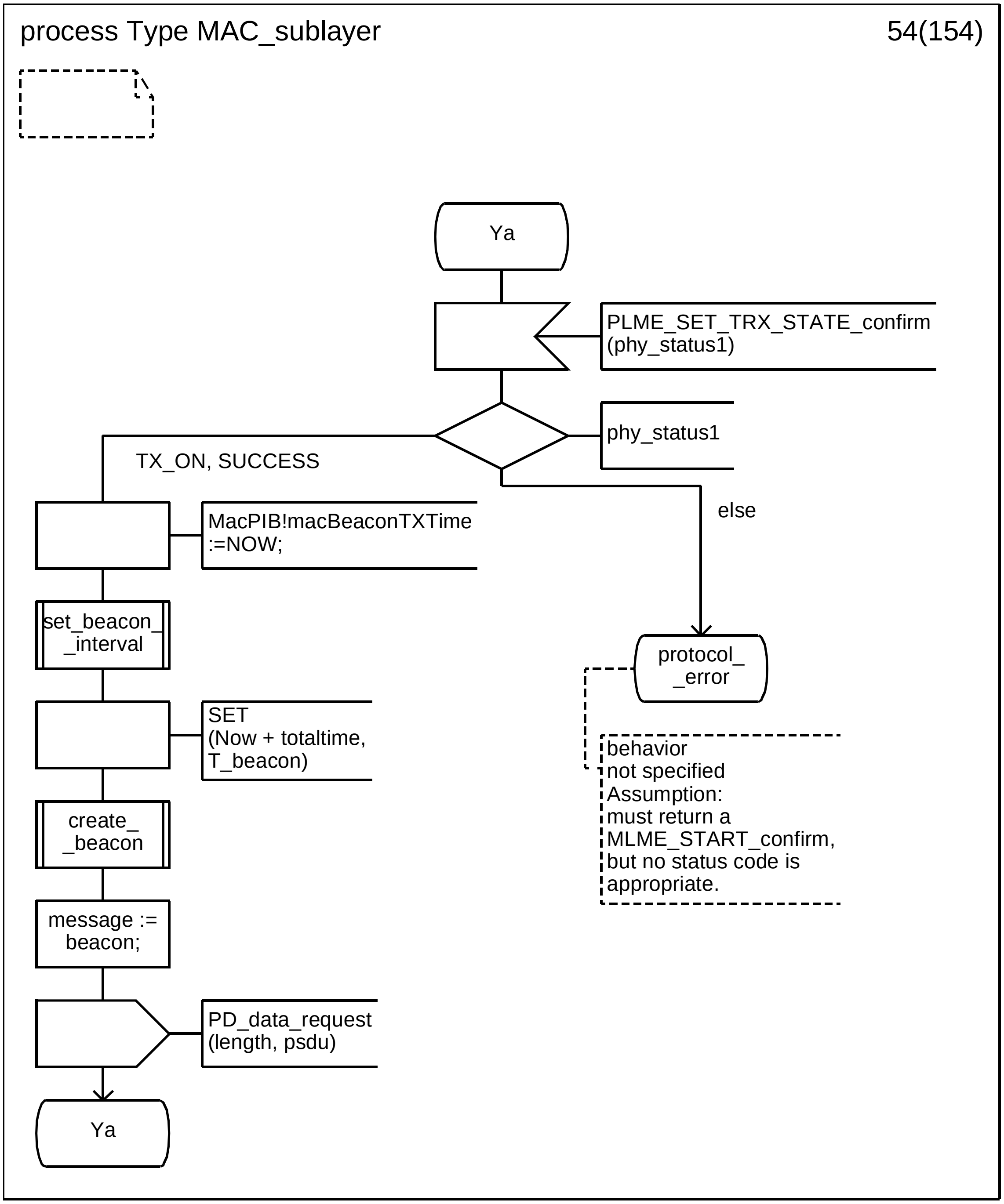


D.3.1.53 Process type MAC_sublayer (53)

.

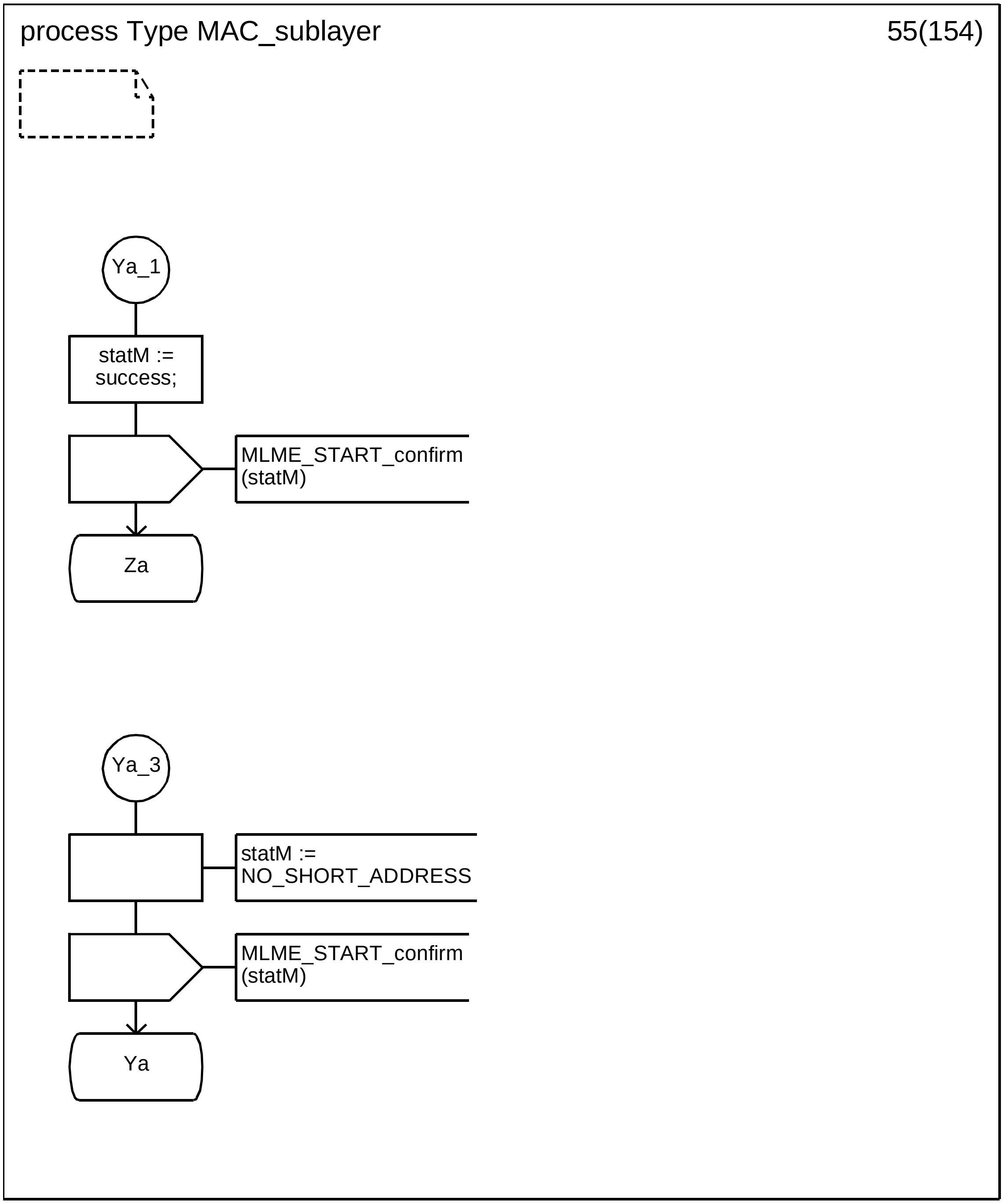


D.3.1.54 Process type MAC_sublayer (54)



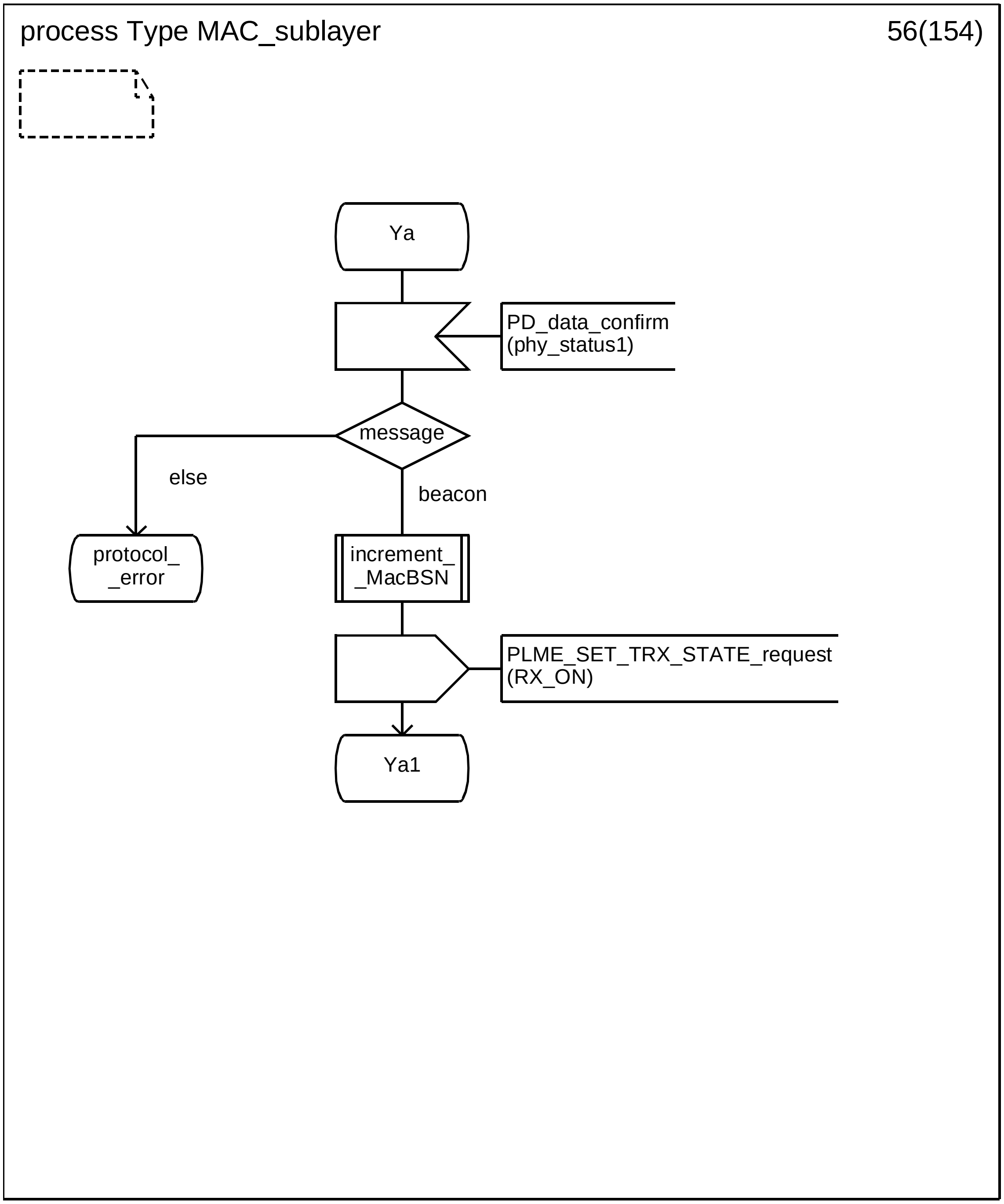
D.3.1.55 Process type MAC_sublayer (55)

.

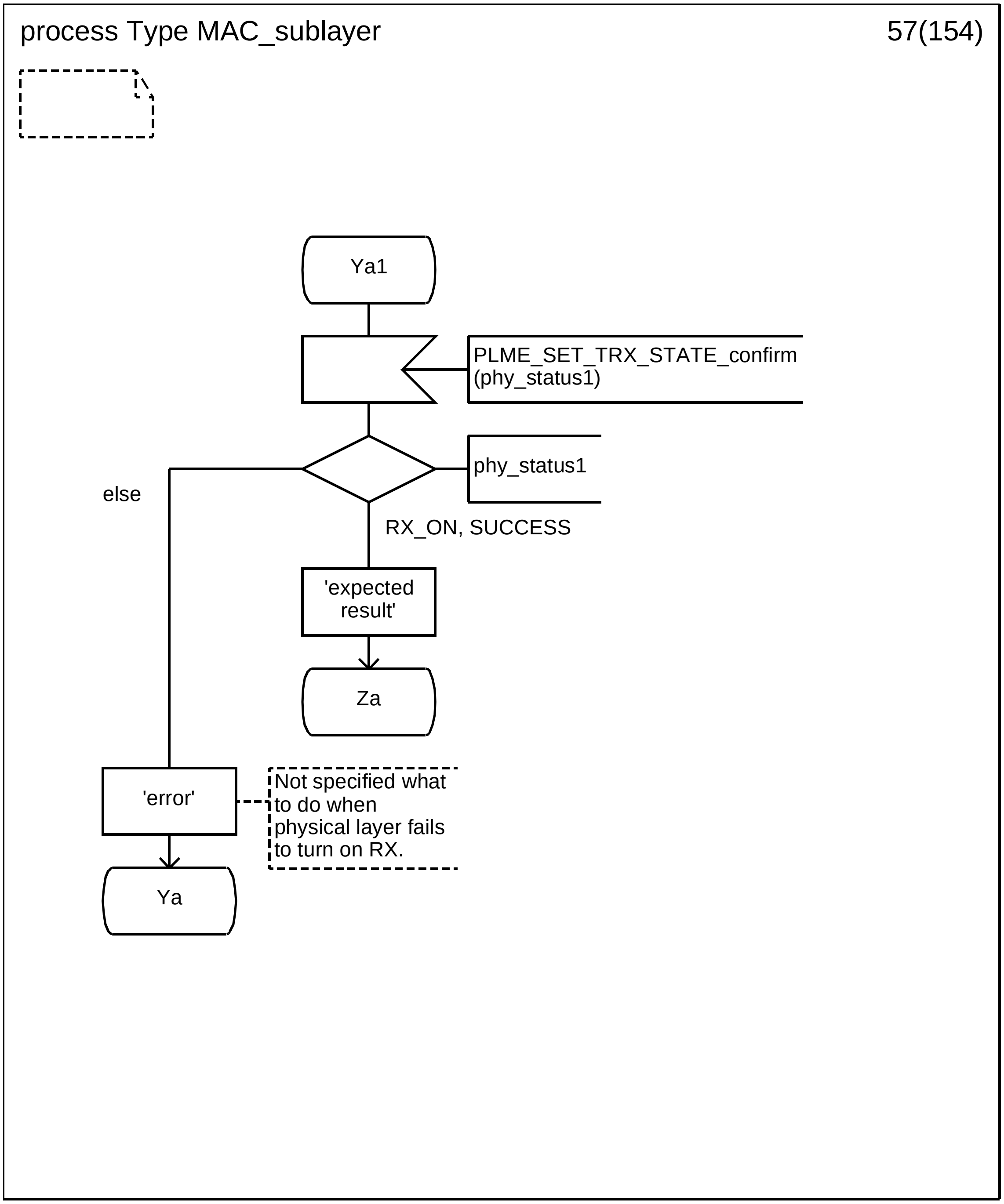


D.3.1.56 Process type MAC_sublayer (56)

.

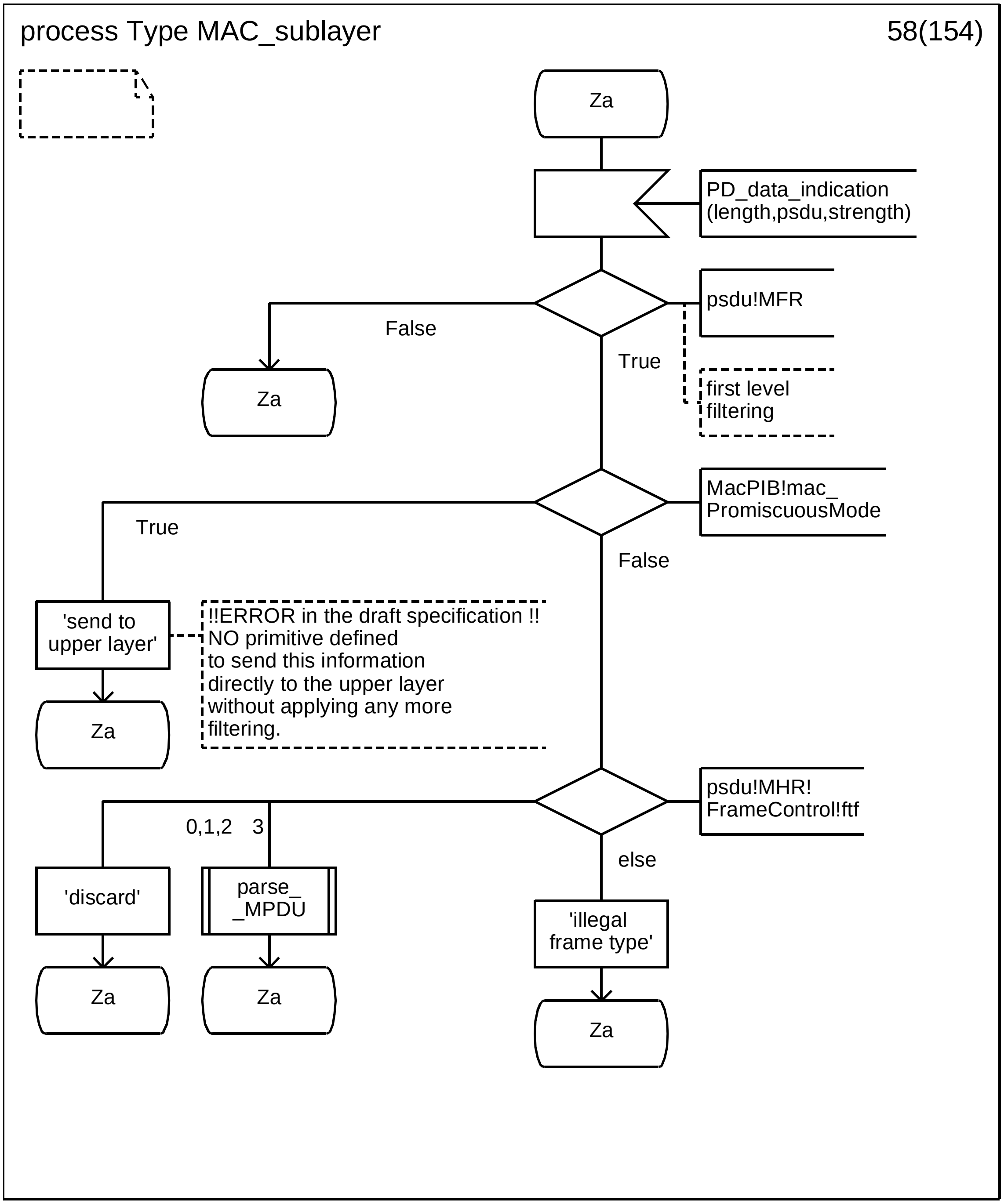


D.3.1.57 Process type MAC_sublayer (57)


process Type MAC_sublayer
57(154)
Ya1
PLME_SET_TRX_STATE_confirm
(phy_status1)
phy_status1
else
RX_ON, SUCCESS
'expected result'
Za
'error'
Not specified what to do when physical layer fails to turn on RX.
Ya

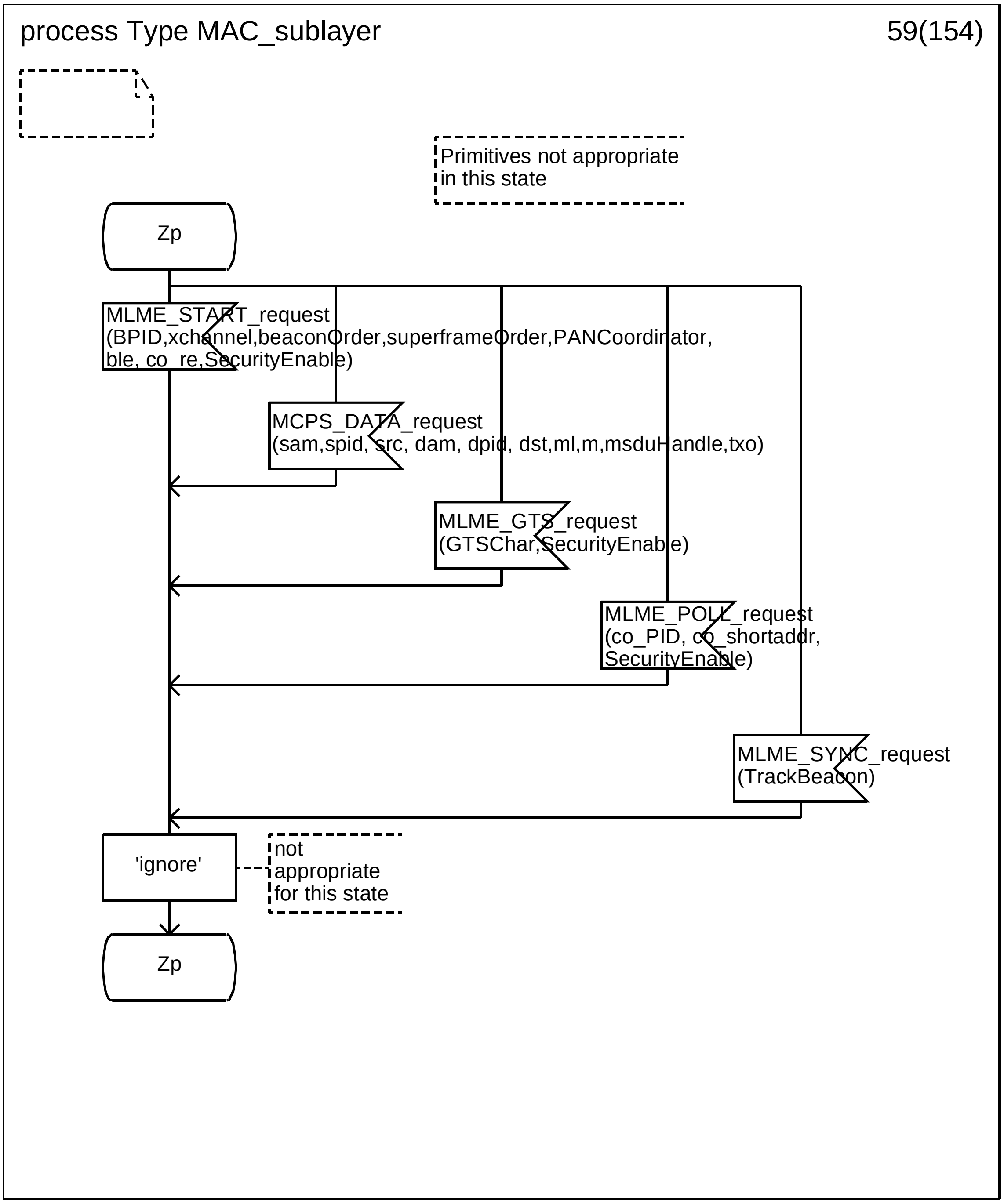
D.3.1.58 Process type MAC_sublayer (58)

.



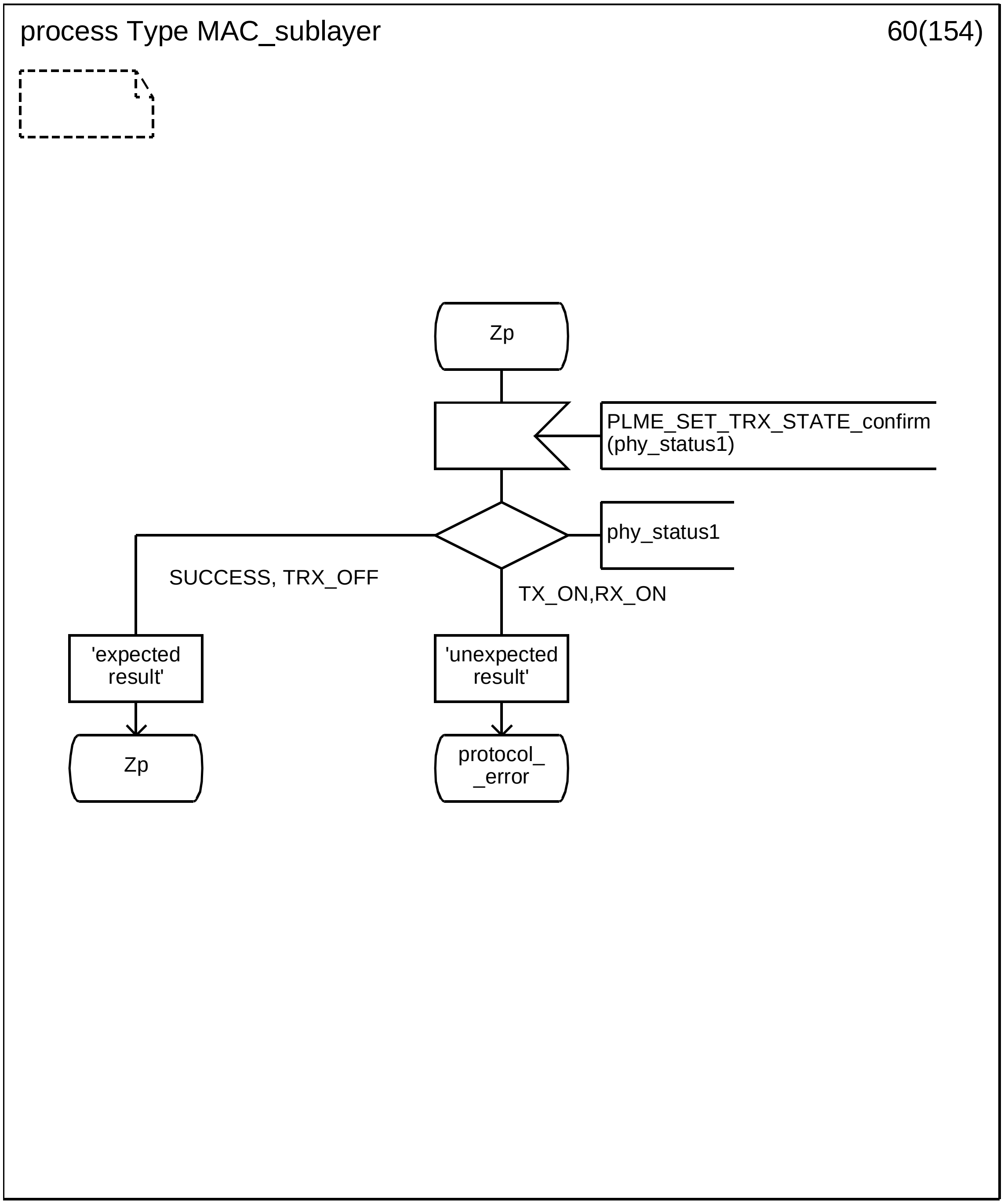
D.3.1.59 Process type MAC_sublayer (59)

.


process Type MAC_sublayer
59(154)
Primitives not appropriate in this state
Zp
MLME_START_request (BPID,xchannel,beaconOrder,superframeOrder,PANCoordinator, ble, co_re,SecurityEnable)
MCPS_DATA_request (sam,spid, src, dam, dpid, dst,ml,m,msduHandle,txo)
MLME_GTS_request (GTSChar,SecurityEnable)
MLME_POLL_request (co_PID, co_shortaddr, SecurityEnable)
MLME_SYNC_request (TrackBeacon)
'ignore'
not appropriate for this state
Zp

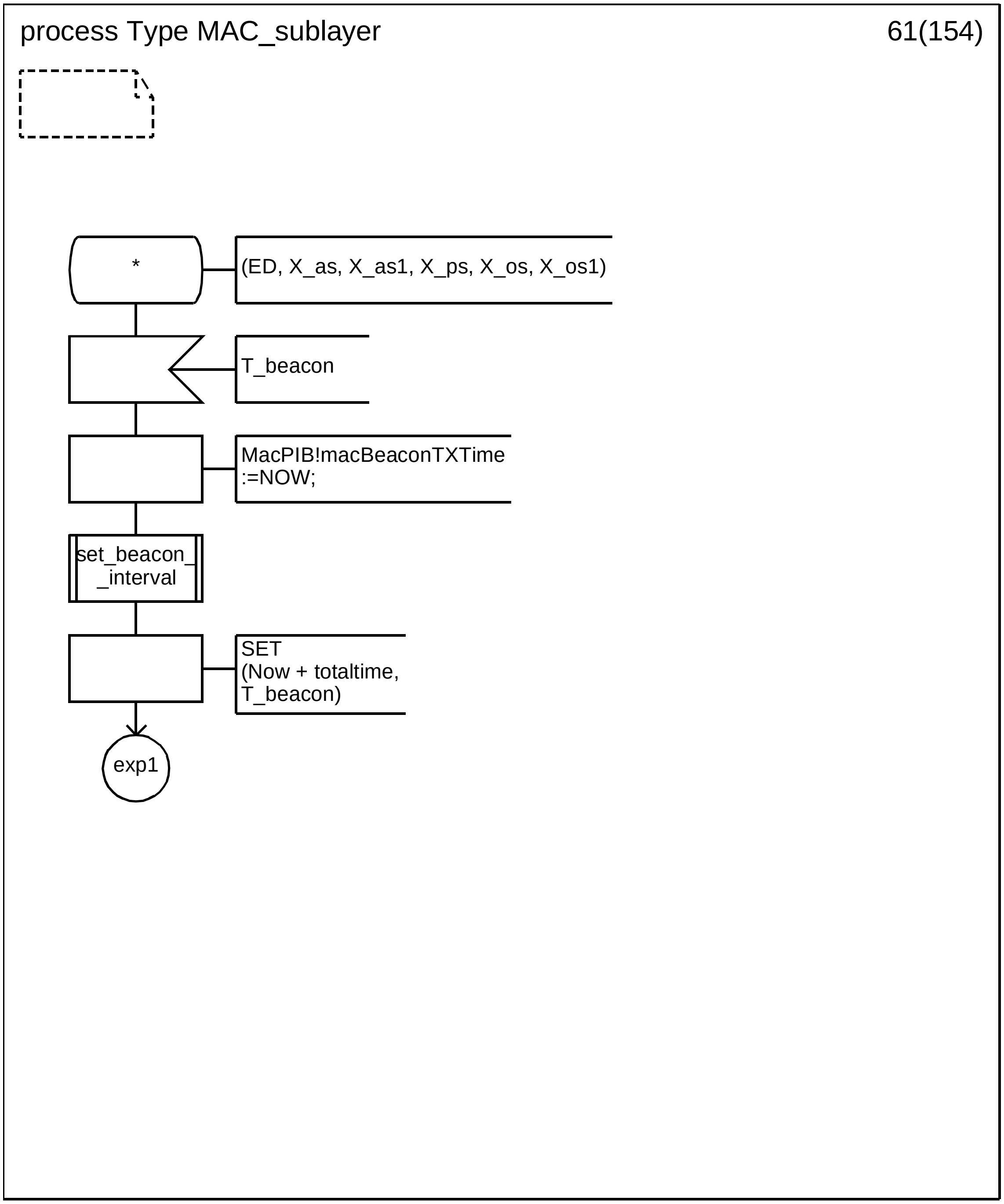
D.3.1.60 Process type MAC_sublayer (60)

.



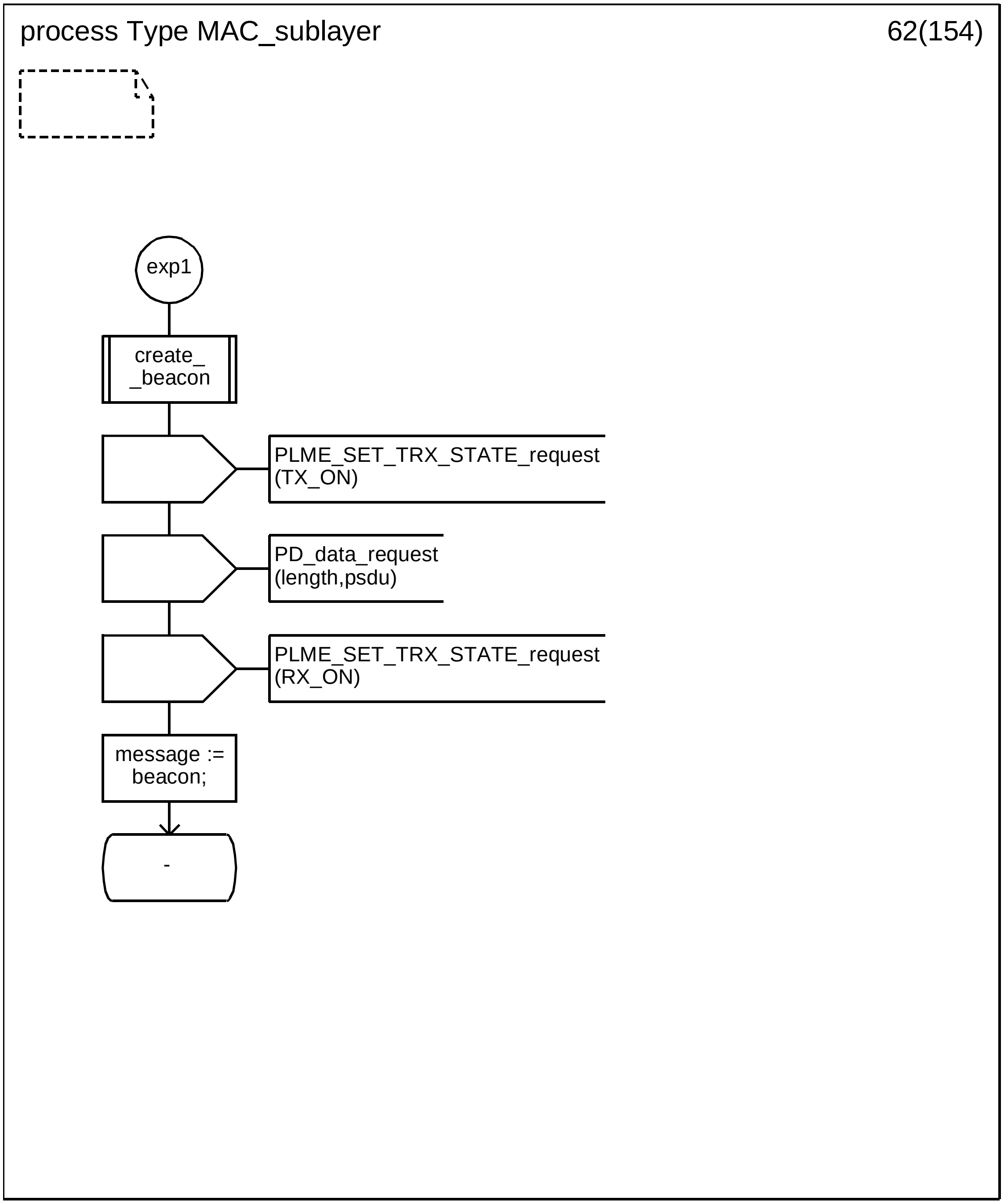
D.3.1.61 Process type MAC_sublayer (61)

.



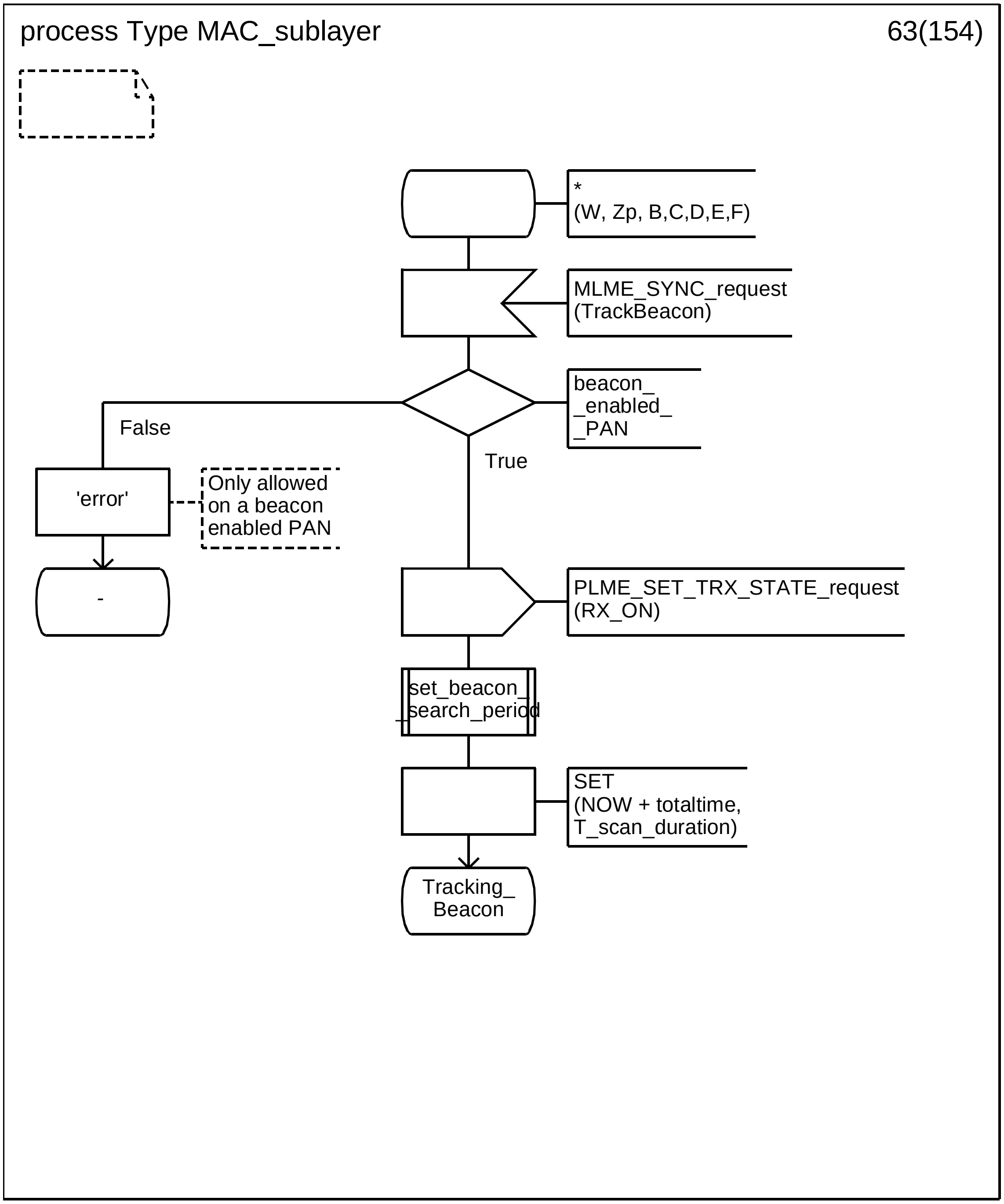
D.3.1.62 Process type MAC_sublayer (62)

.



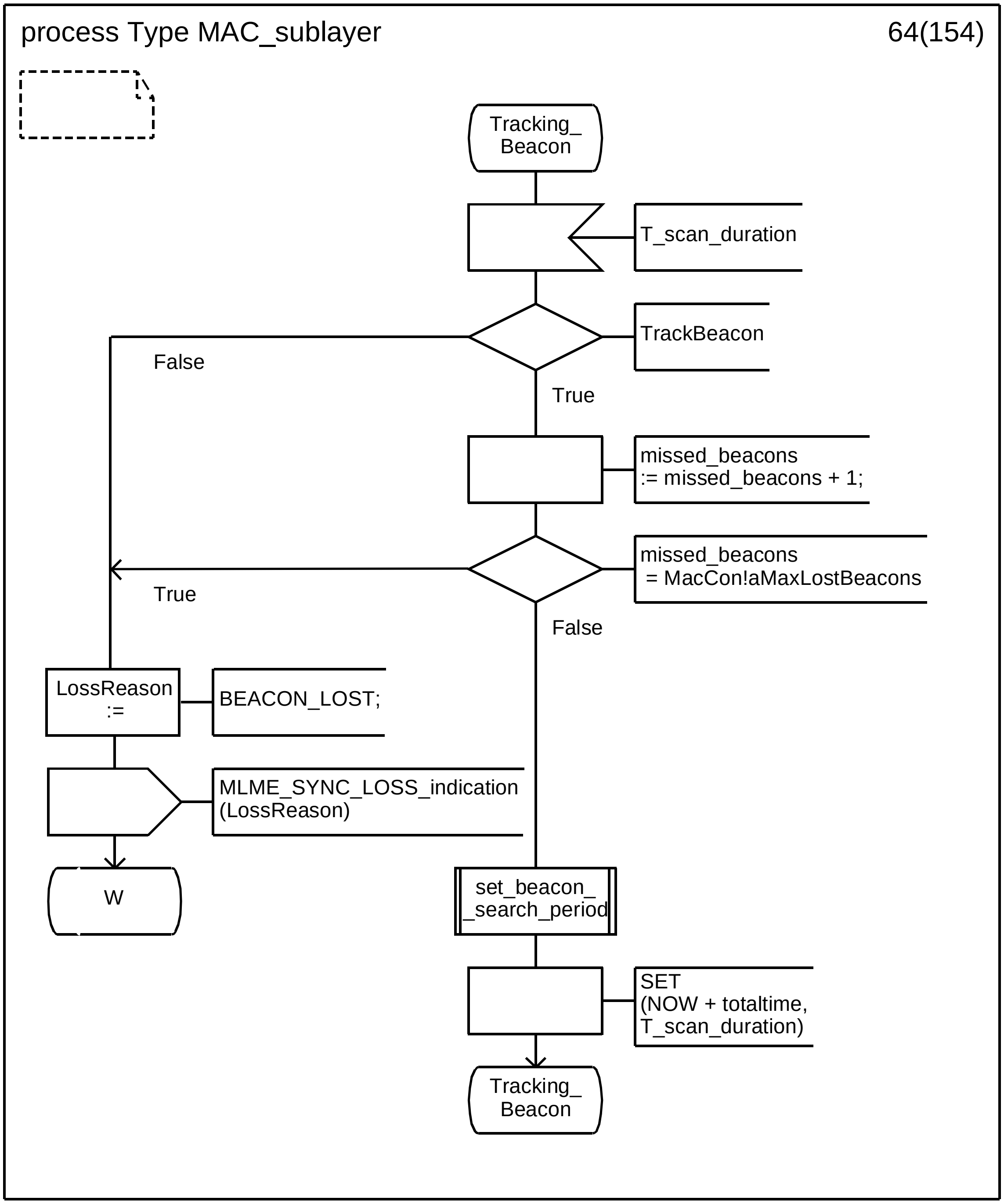
D.3.1.63 Process type MAC_sublayer (63)

.



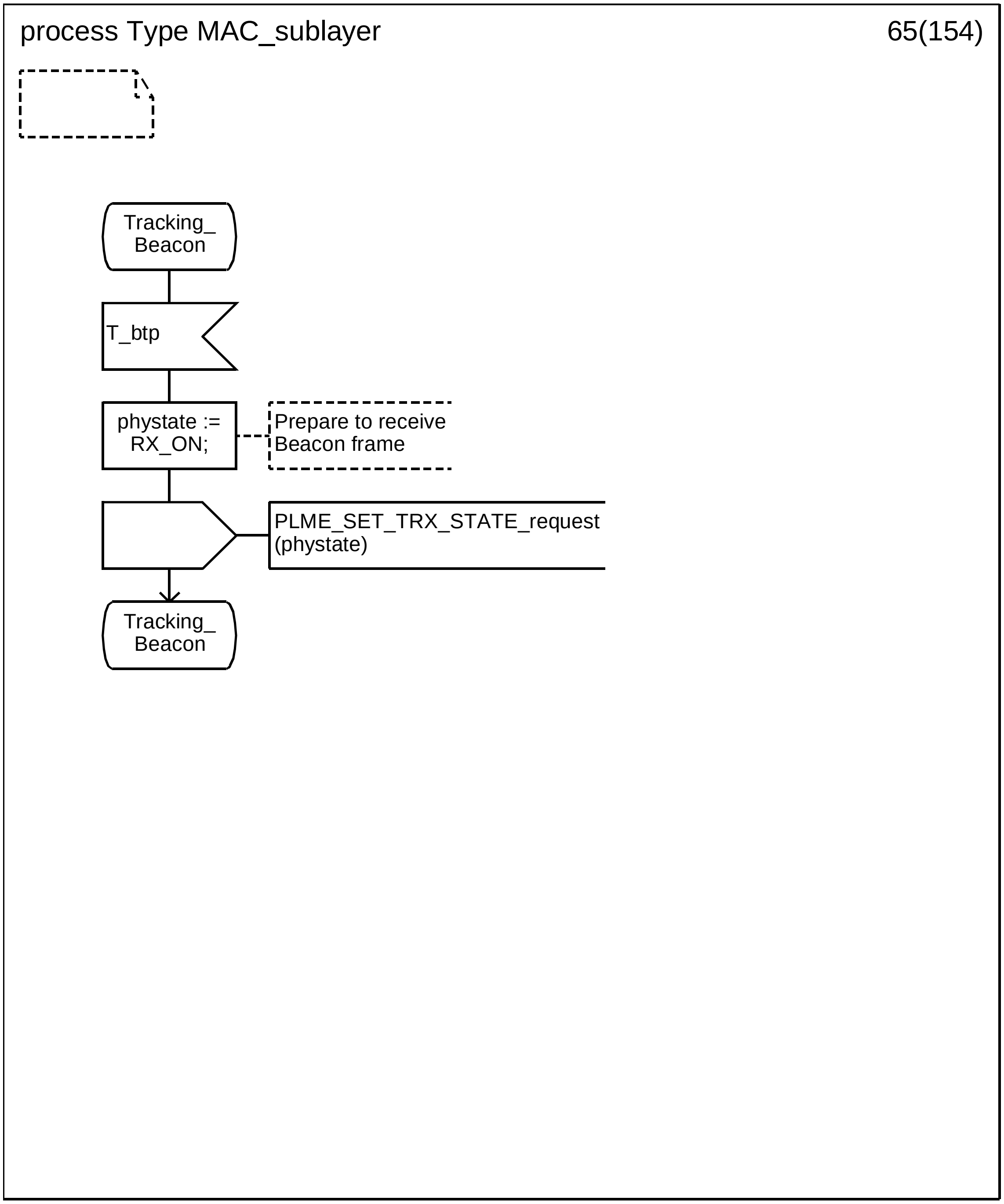
D.3.1.64 Process type MAC_sublayer (64)

.



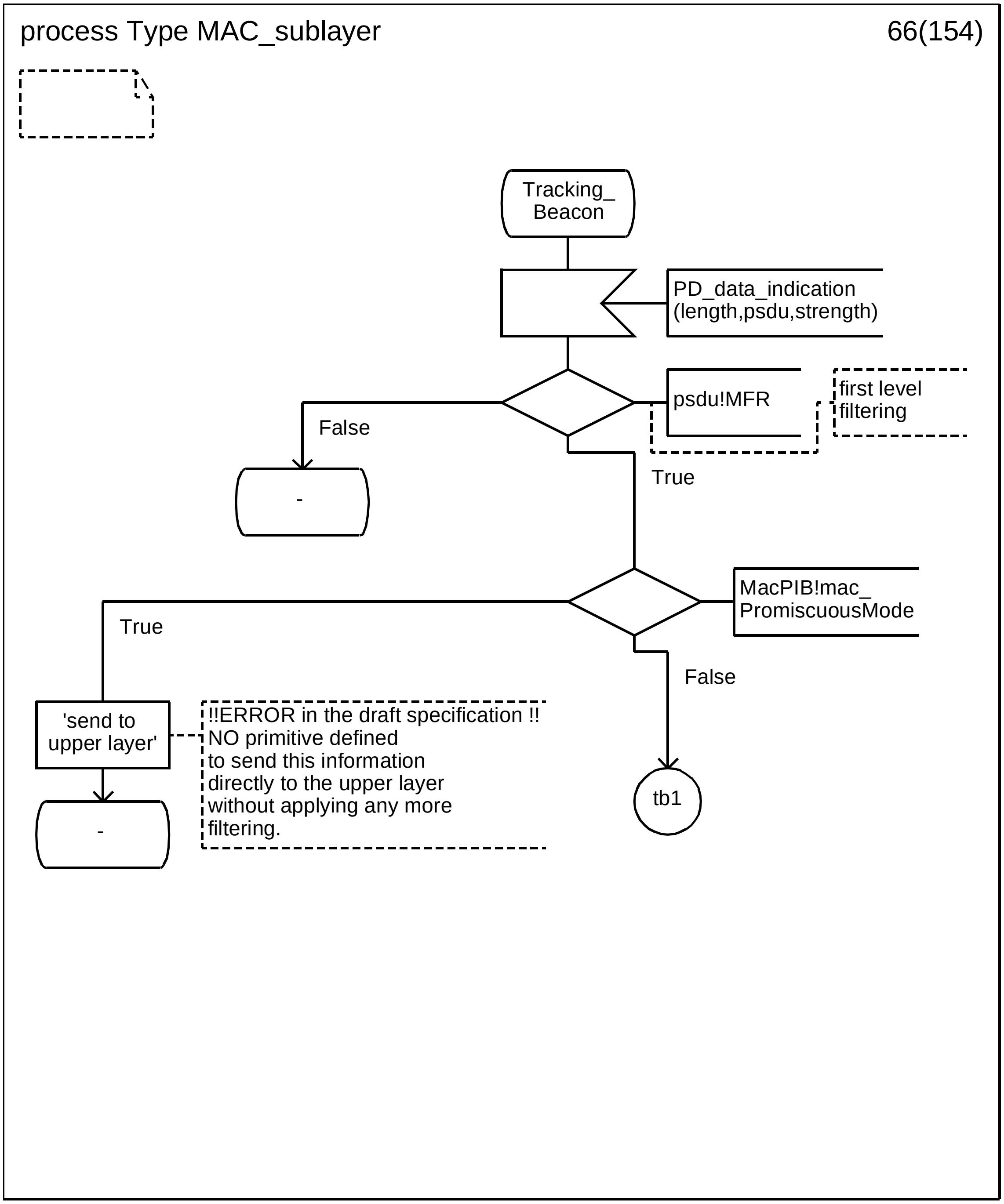
D.3.1.65 Process type MAC_sublayer (65)

.


process Type MAC_sublayer
65(154)
Tracking_ Beacon
T_btp
phystate := RX_ON;
Prepare to receive Beacon frame
PLME_SET_TRX_STATE_request (phystate)
Tracking_ Beacon

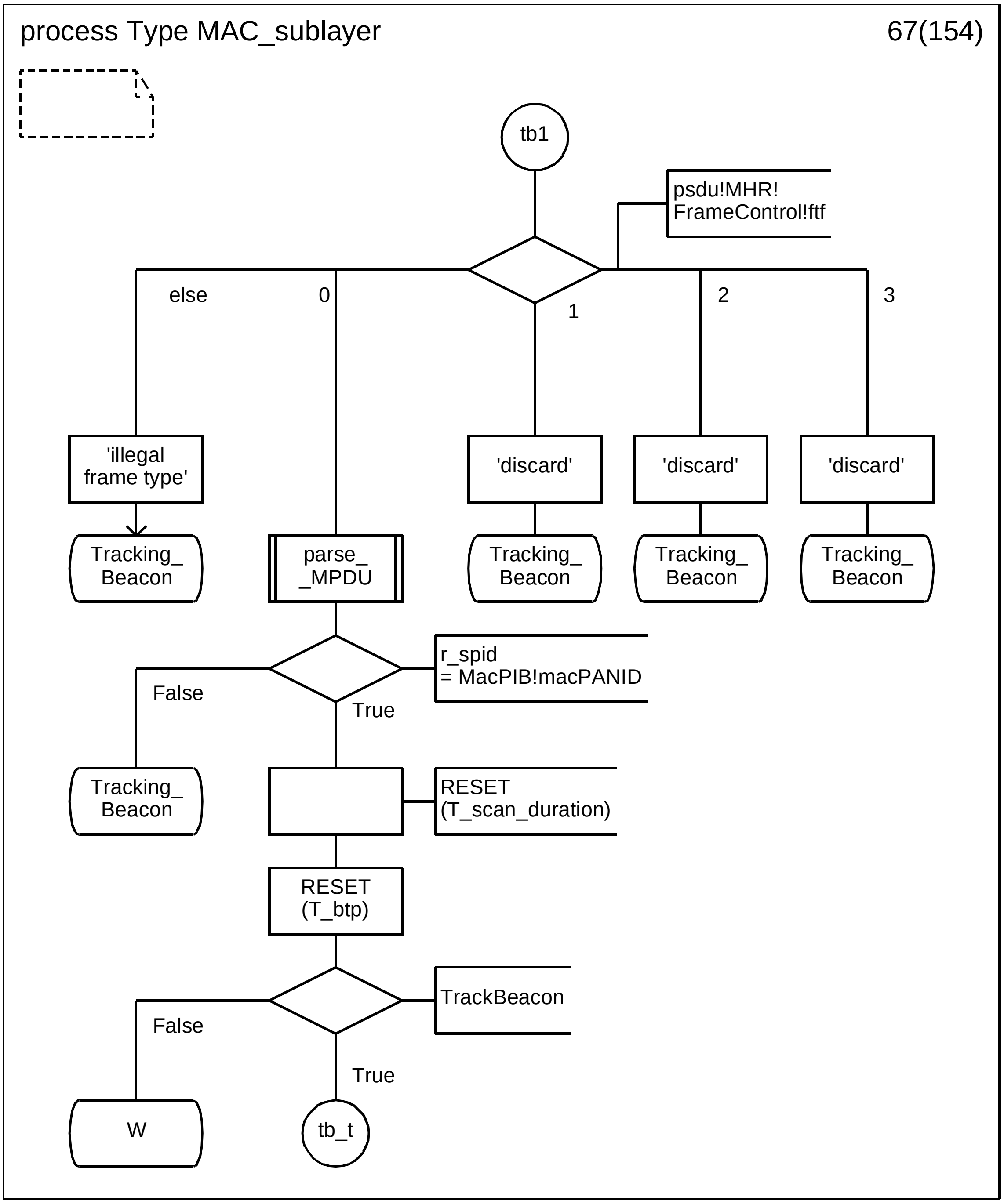
D.3.1.66 Process type MAC_sublayer (66)

.



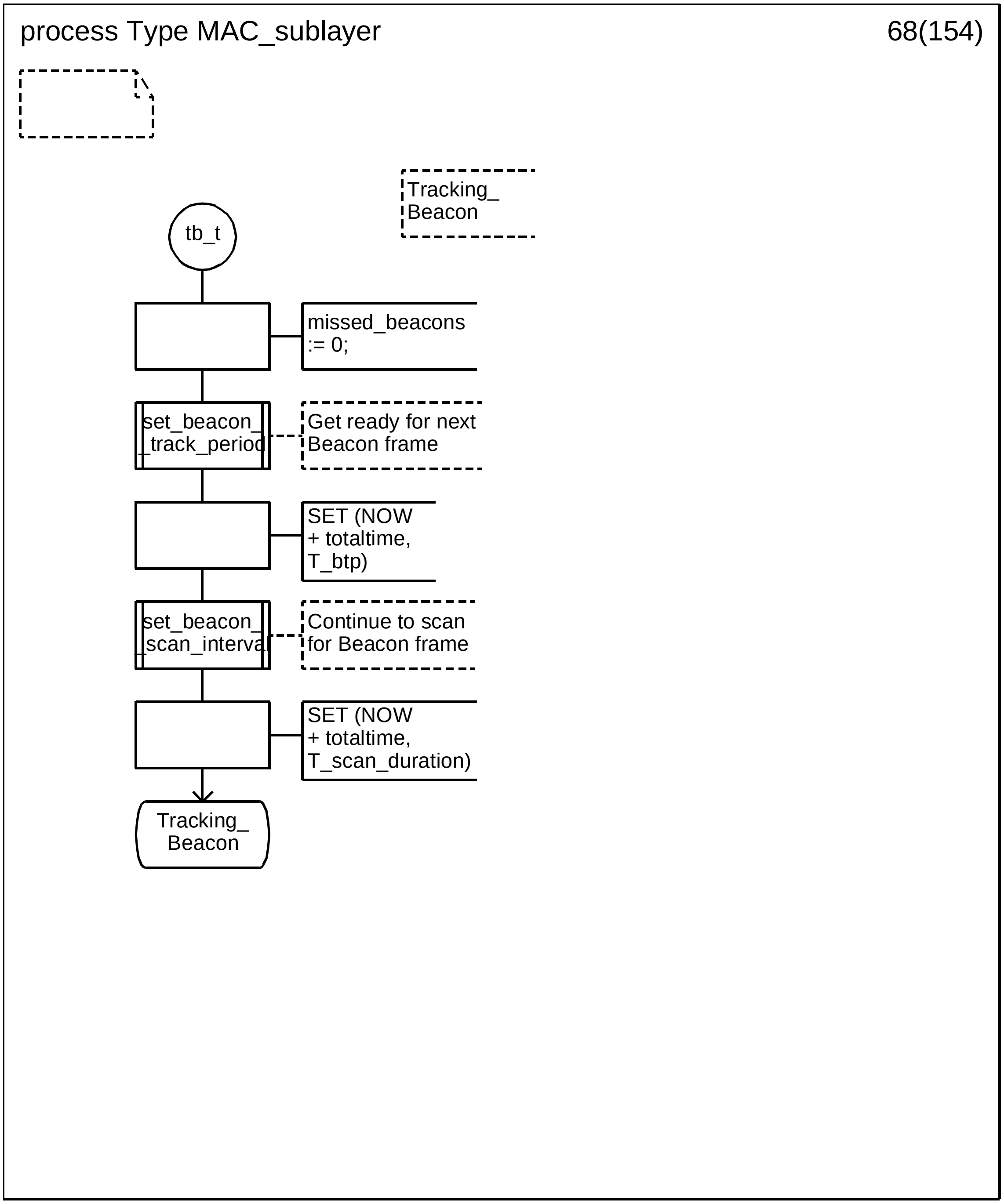
D.3.1.67 Process type MAC_sublayer (67)

.



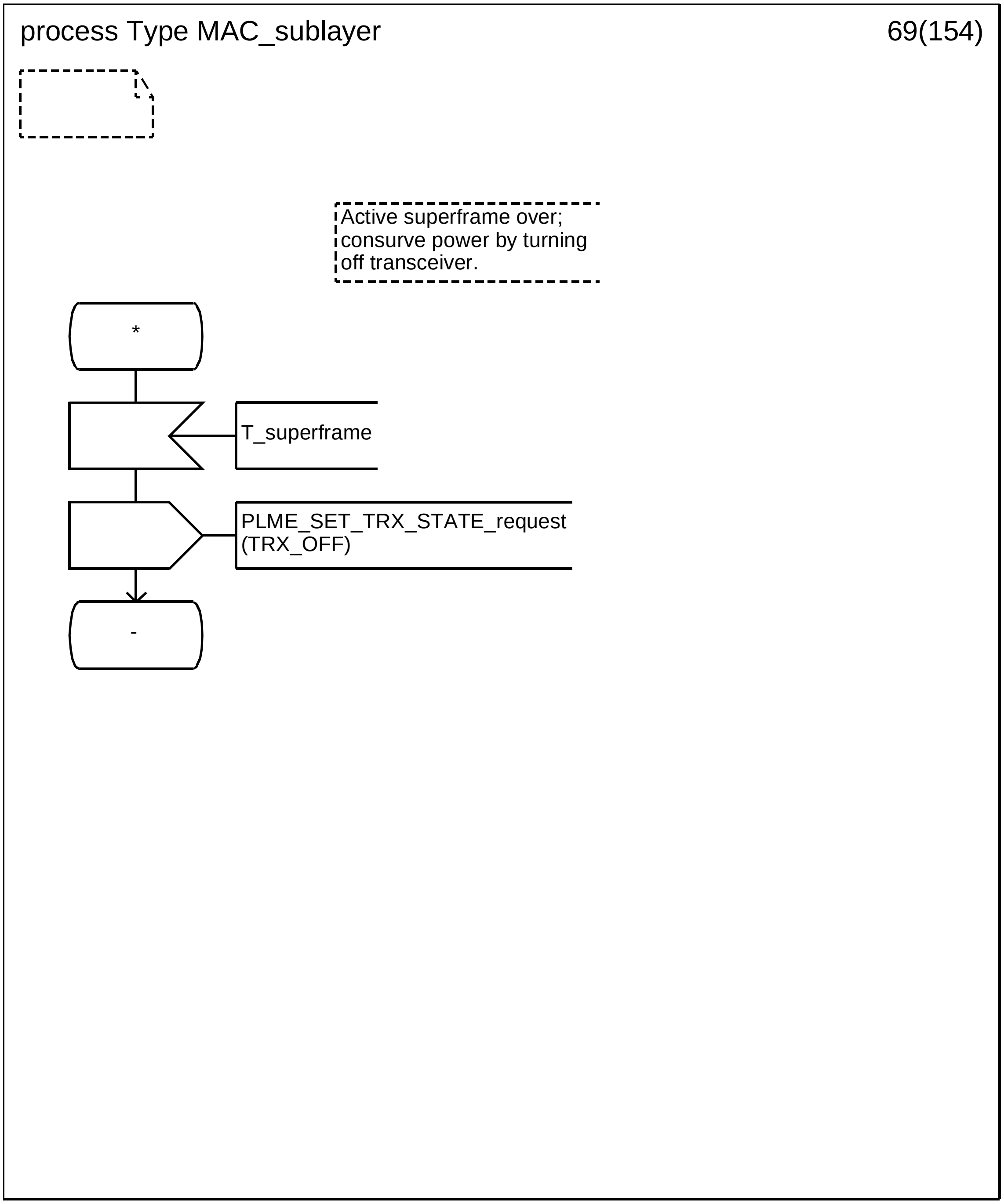
D.3.1.68 Process type MAC_sublayer (68)

.



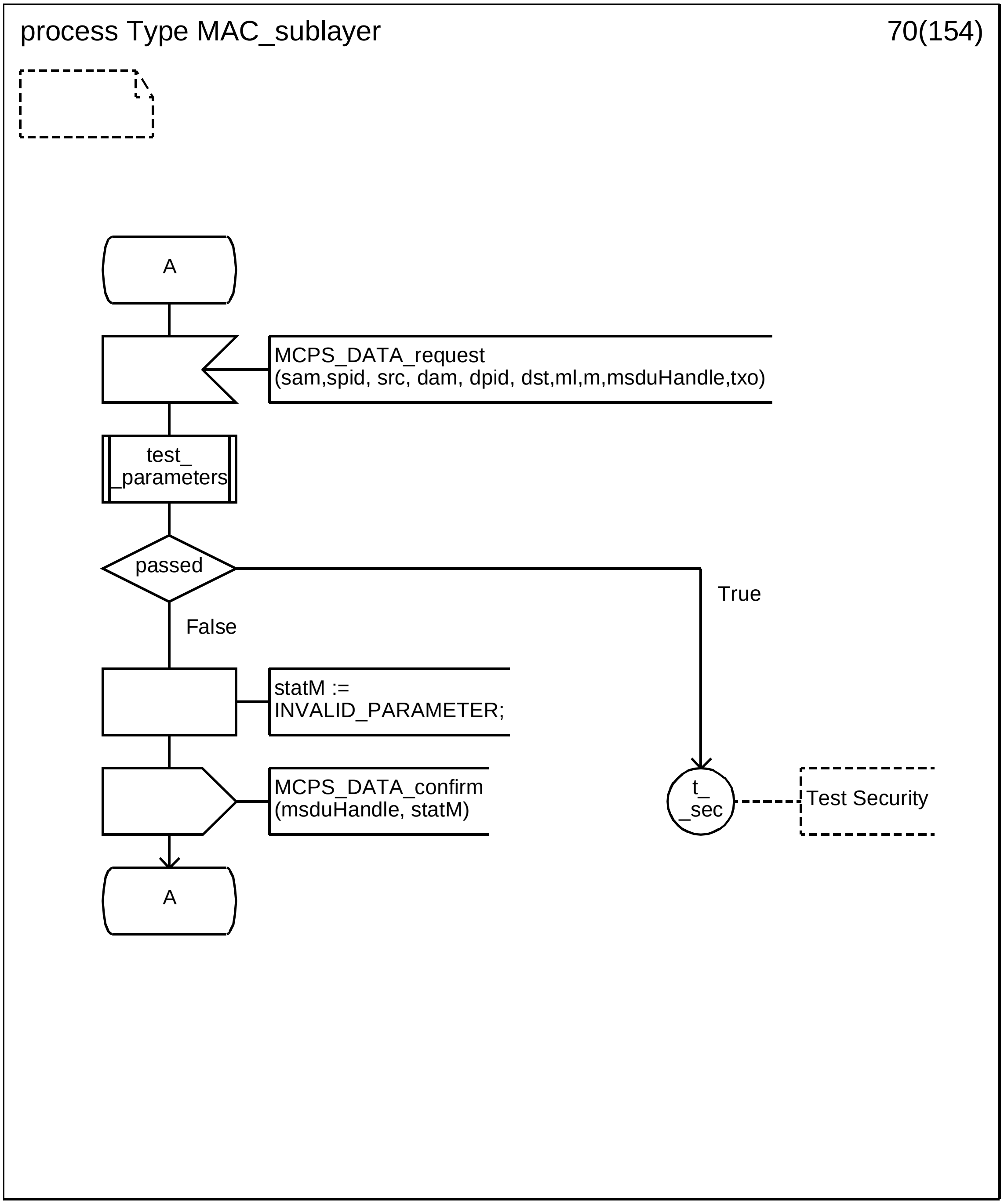
D.3.1.69 Process type MAC_sublayer (69)

.


process Type MAC_sublayer
69(154)
Active superframe over; consurve power by turning off transceiver.
*
T_superframe
PLME_SET_TRX_STATE_request (TRX_OFF)
-

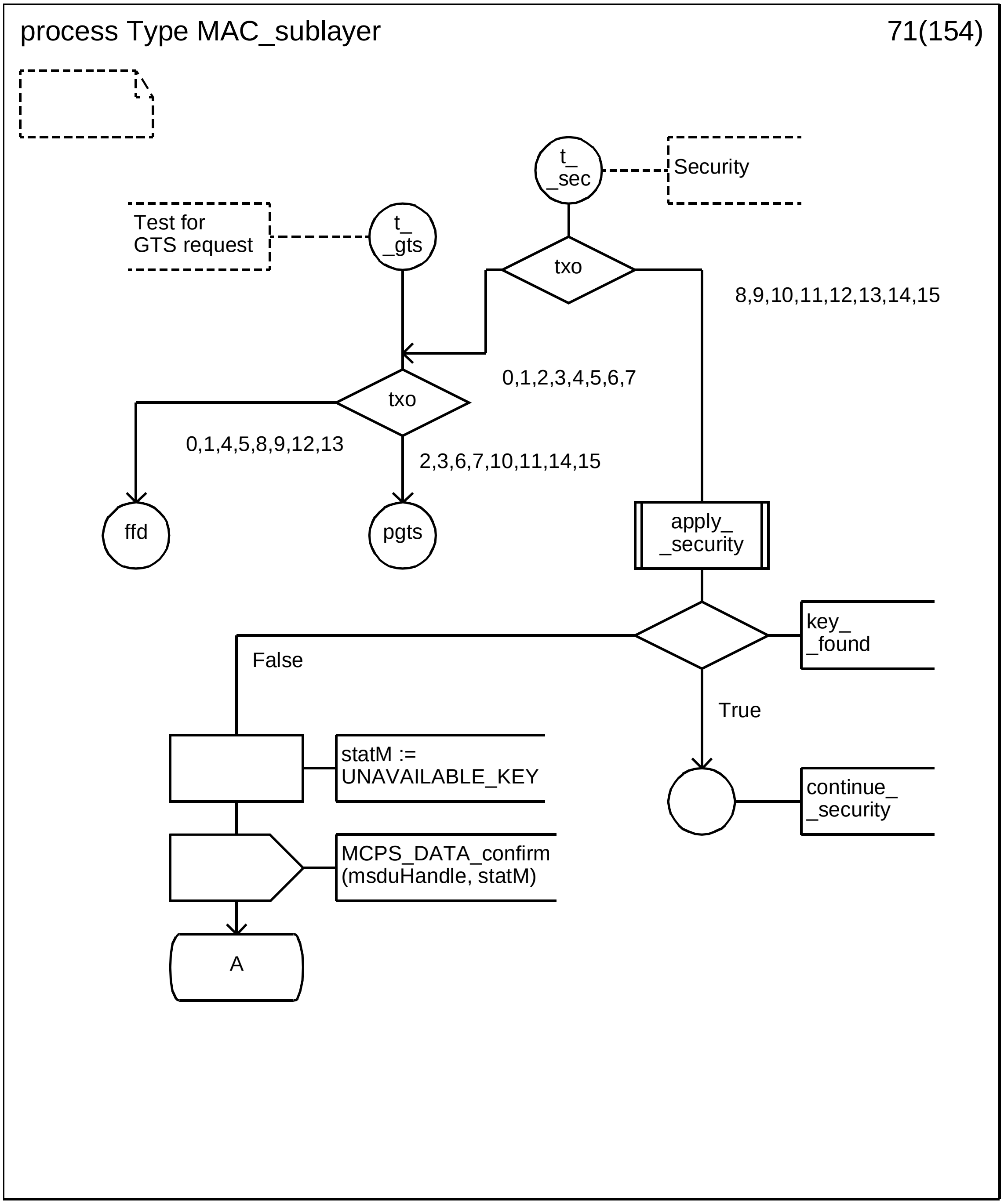
D.3.1.70 Process type MAC_sublayer (70)

.



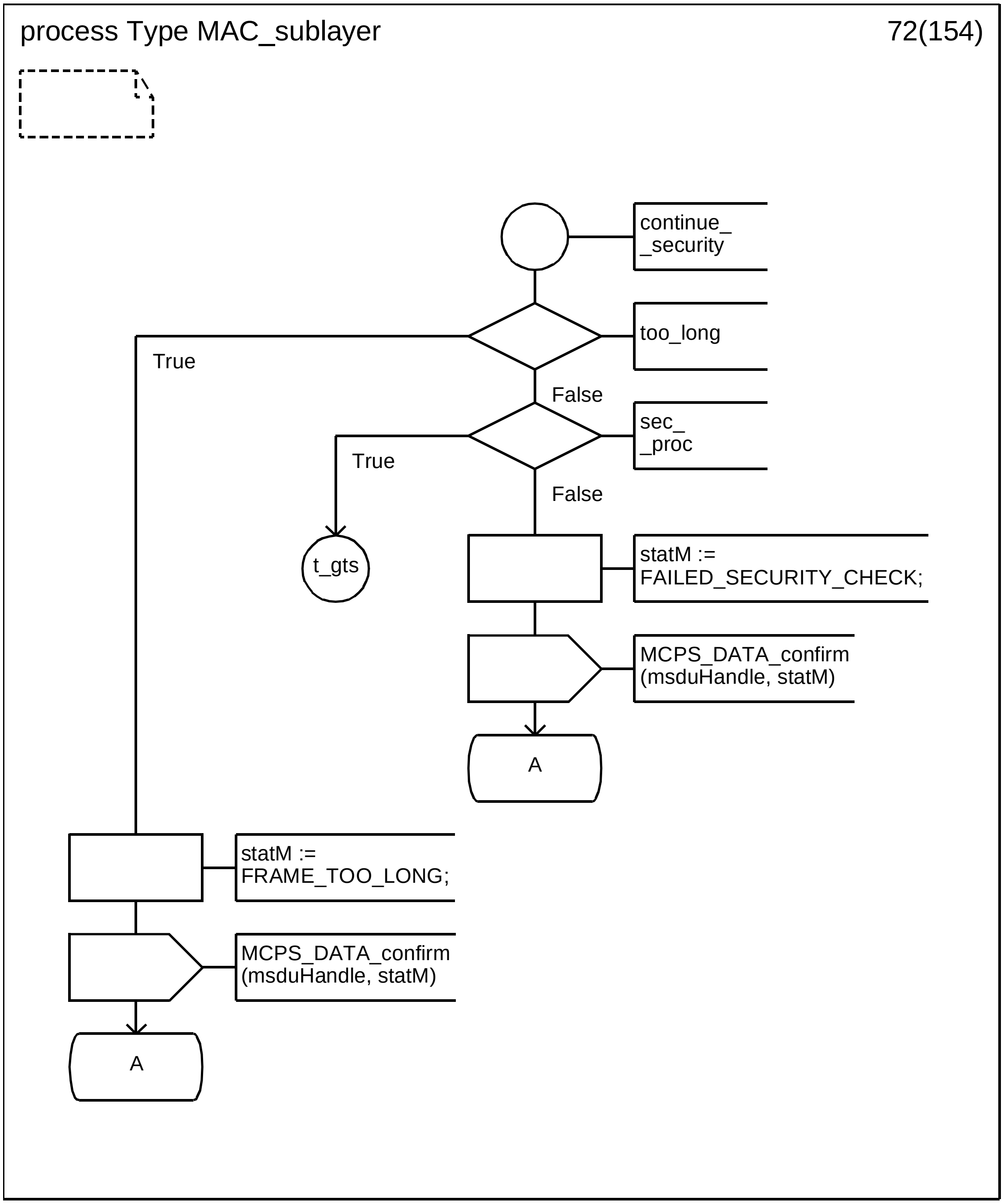
D.3.1.71 Process type MAC_sublayer (71)

.



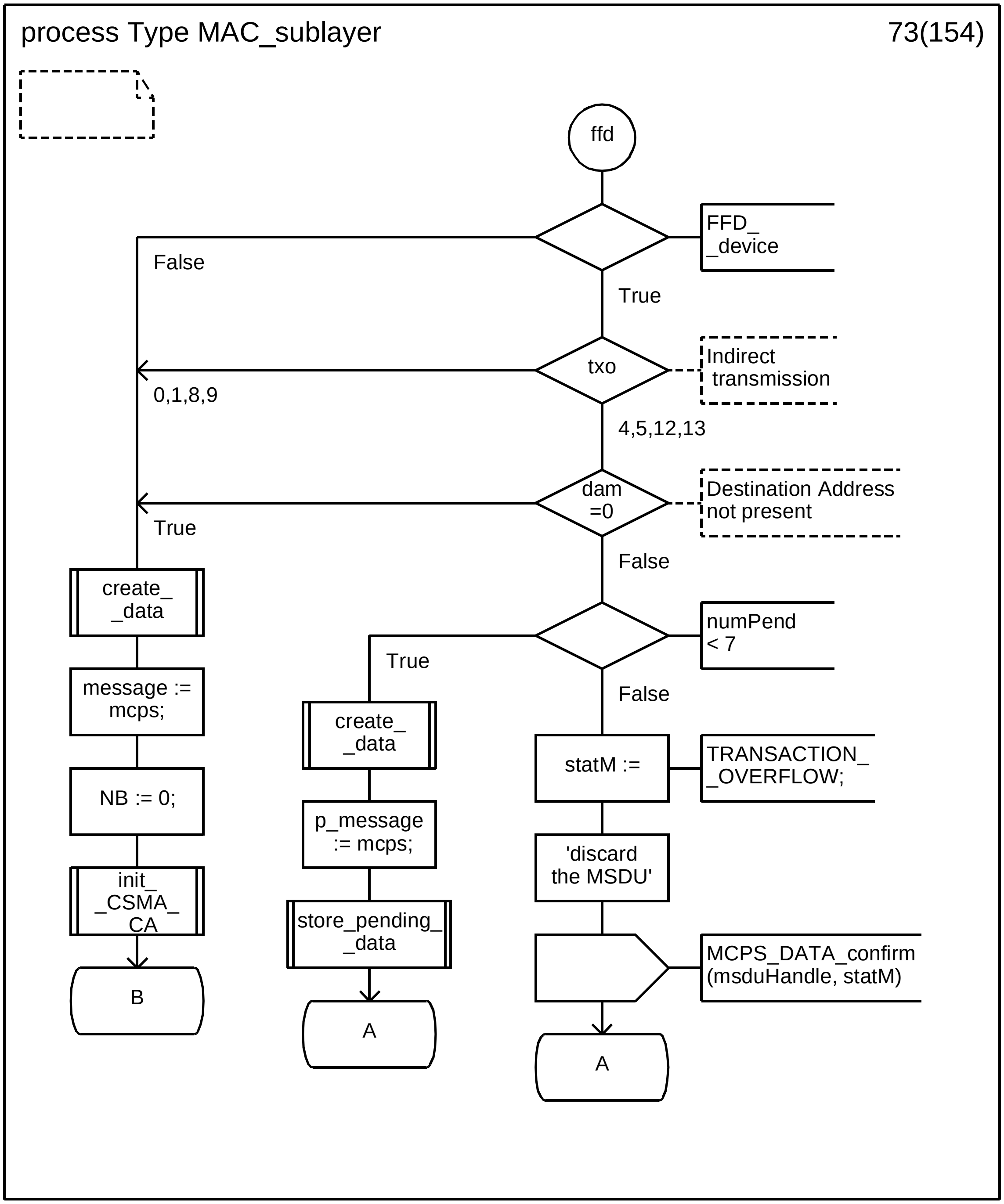
D.3.1.72 Process type MAC_sublayer (72)

.



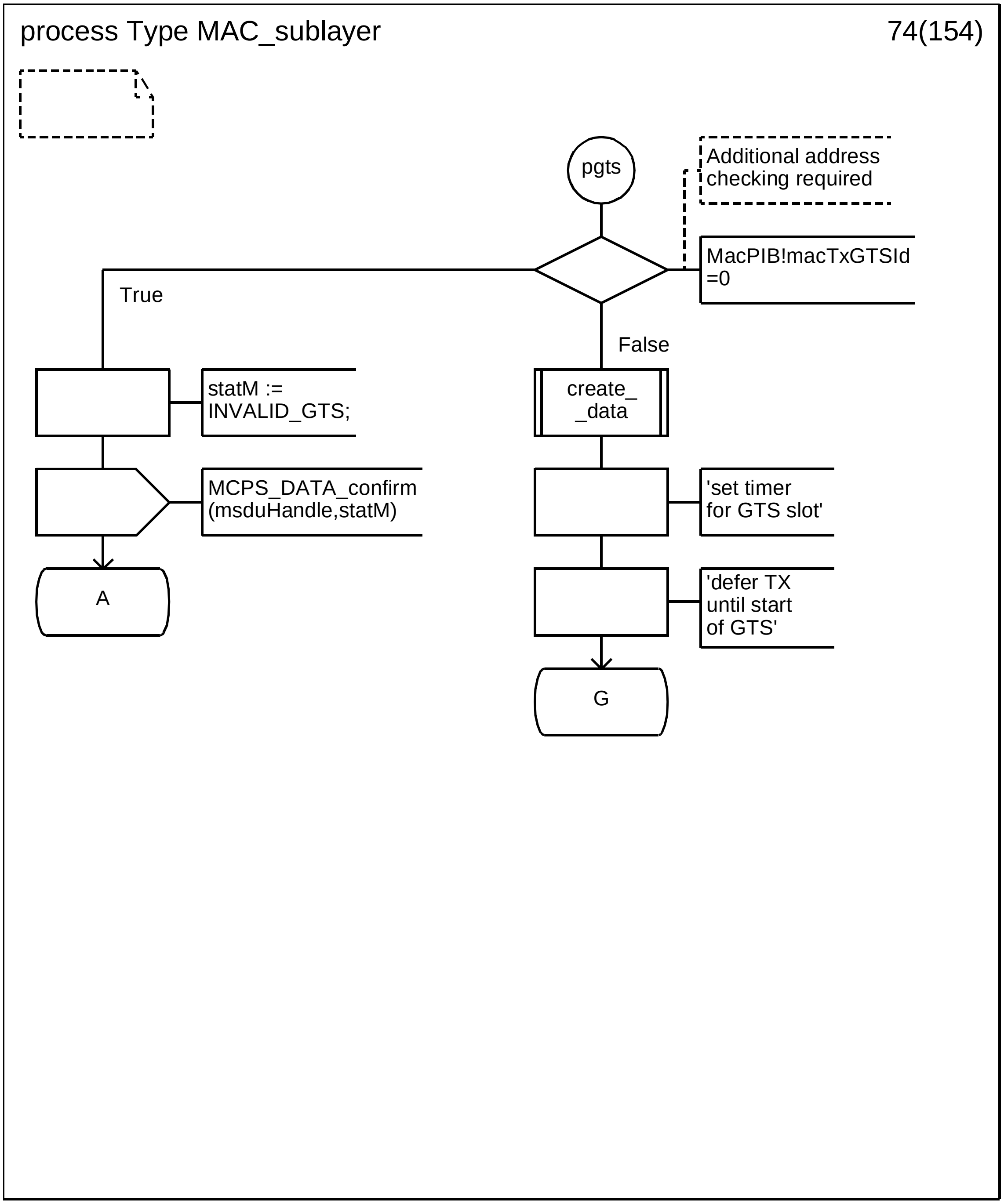
D.3.1.73 Process type MAC_sublayer (73)

.



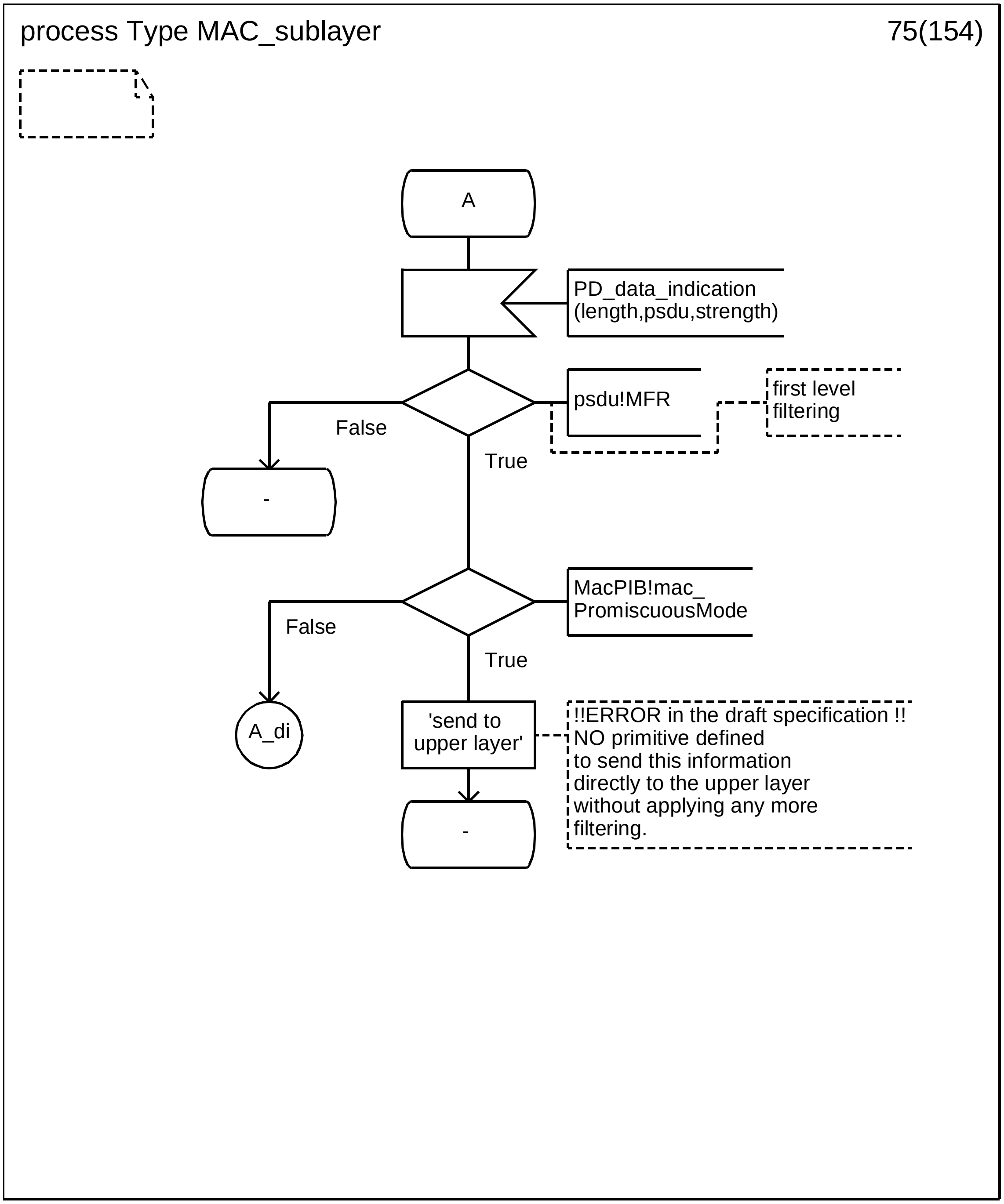
D.3.1.74 Process type MAC_sublayer (74)

.


process Type MAC_sublayer
74(154)
pgts
Additional address checking required
MacPIB!macTxGTSId =0
True
False
statM := INVALID_GTS;
MCPS_DATA_confirm (msduHandle,statM)
A
create_ _data
'set timer for GTS slot'
'defer TX until start of GTS'
G

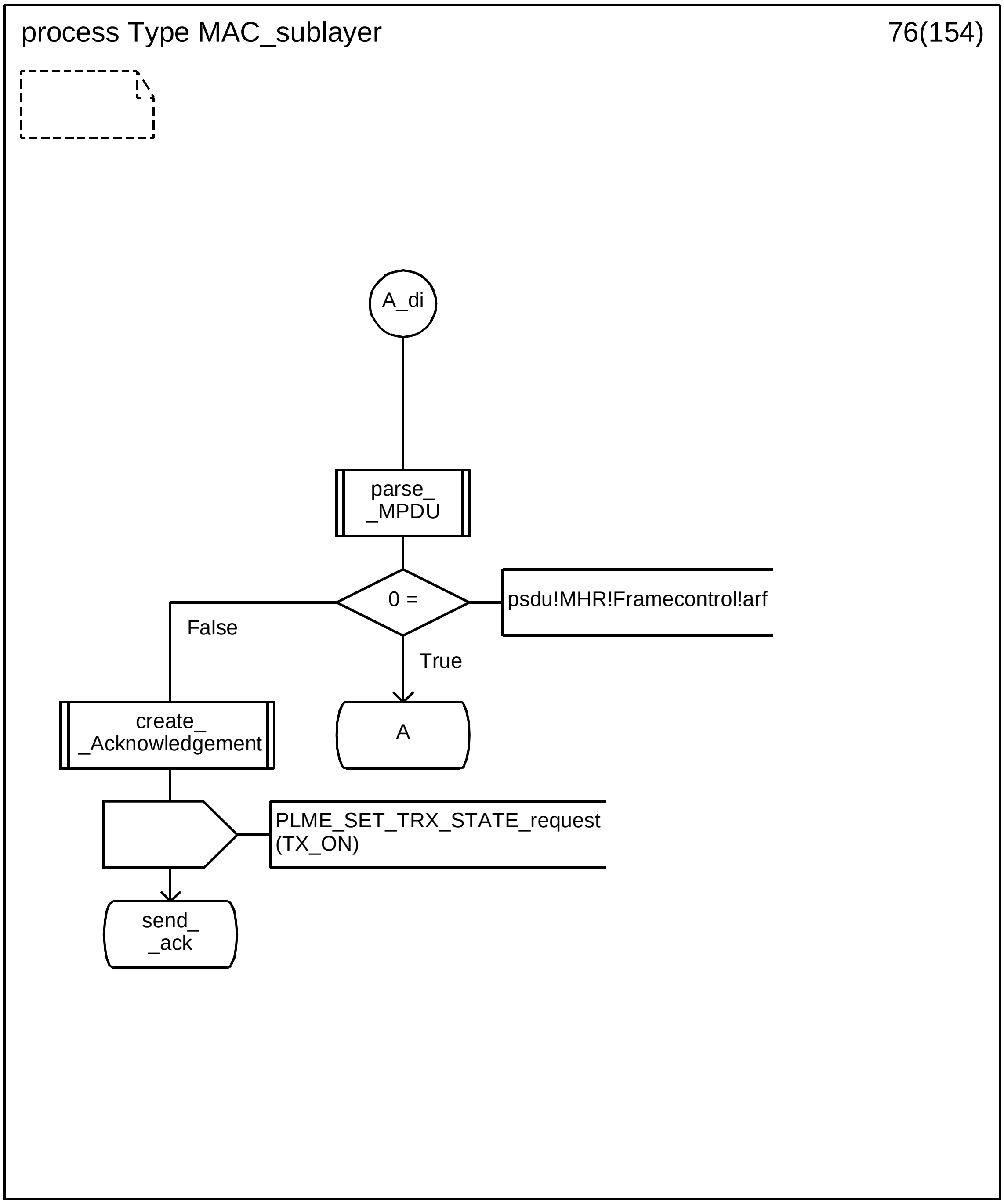
D.3.1.75 Process type MAC_sublayer (75)

.



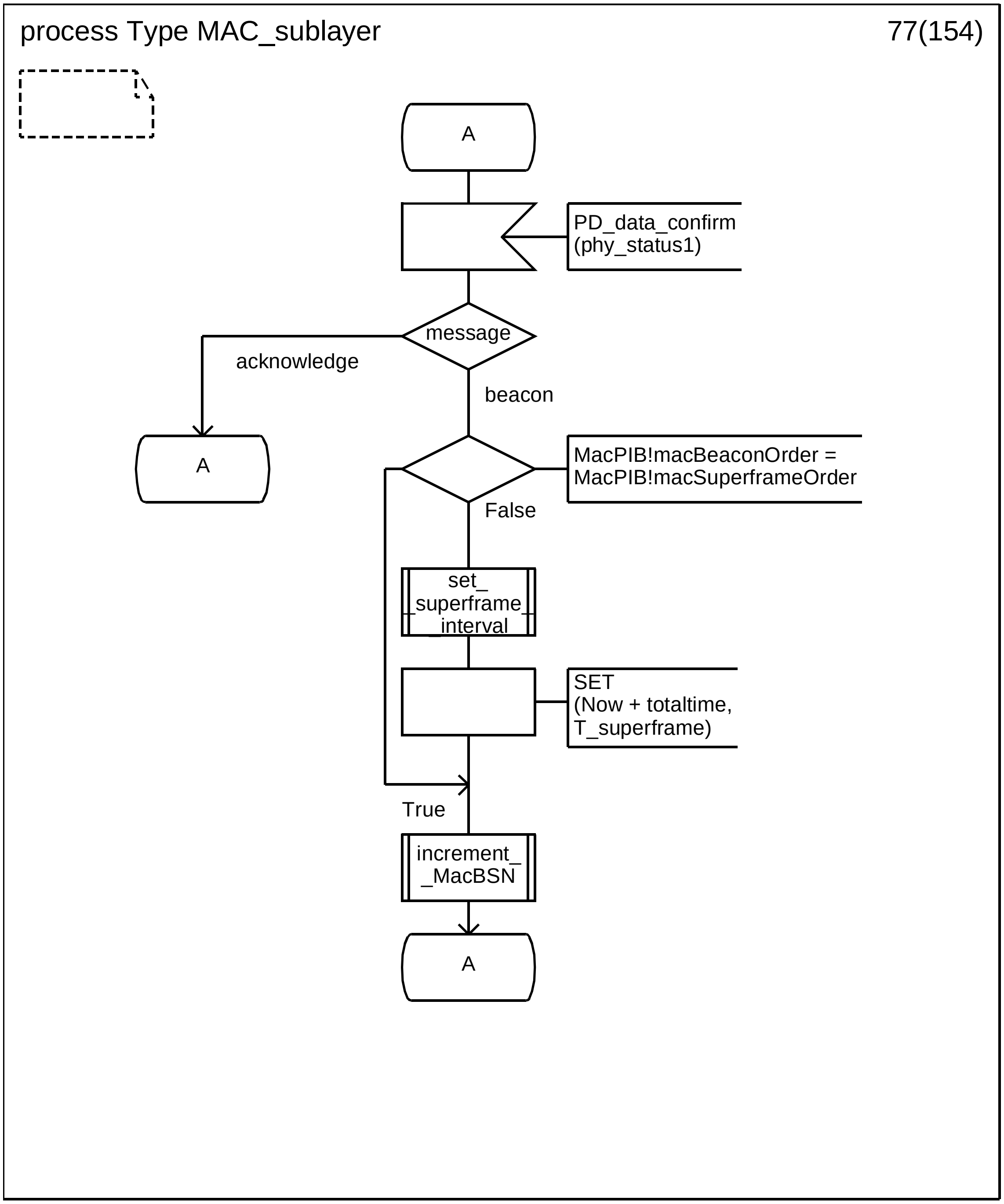
D.3.1.76 Process type MAC_sublayer (76)

.


process Type MAC_sublayer
76(154)
A_di
parse_
_MPDU
0 =
psdu!MHR!Framecontrol!arf
False
True
A
create_
_Acknowledgement
PLME_SET_TRX_STATE_request
(TX_ON)
send_
_ack

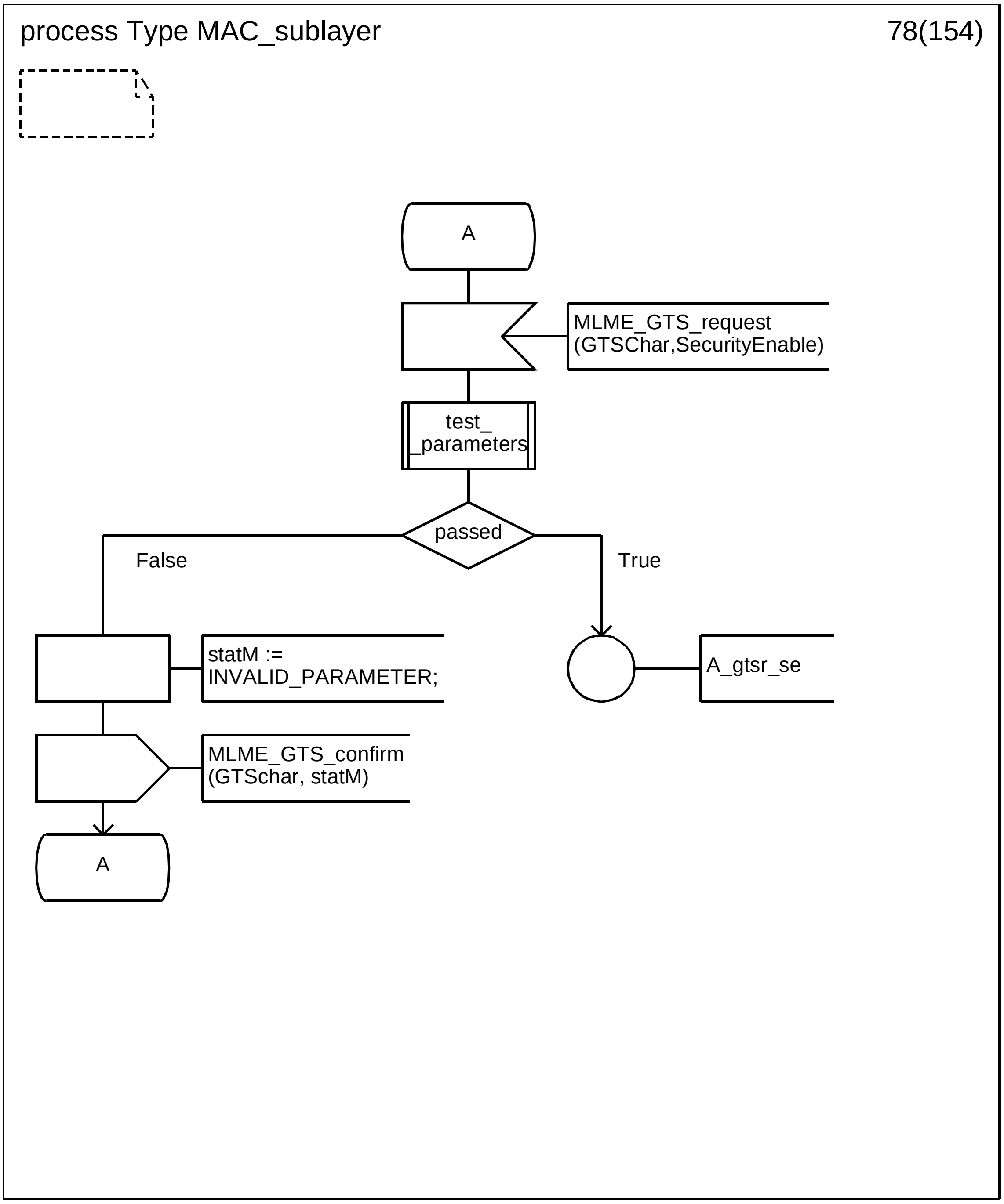
D.3.1.77 Process type MAC_sublayer (77)

.



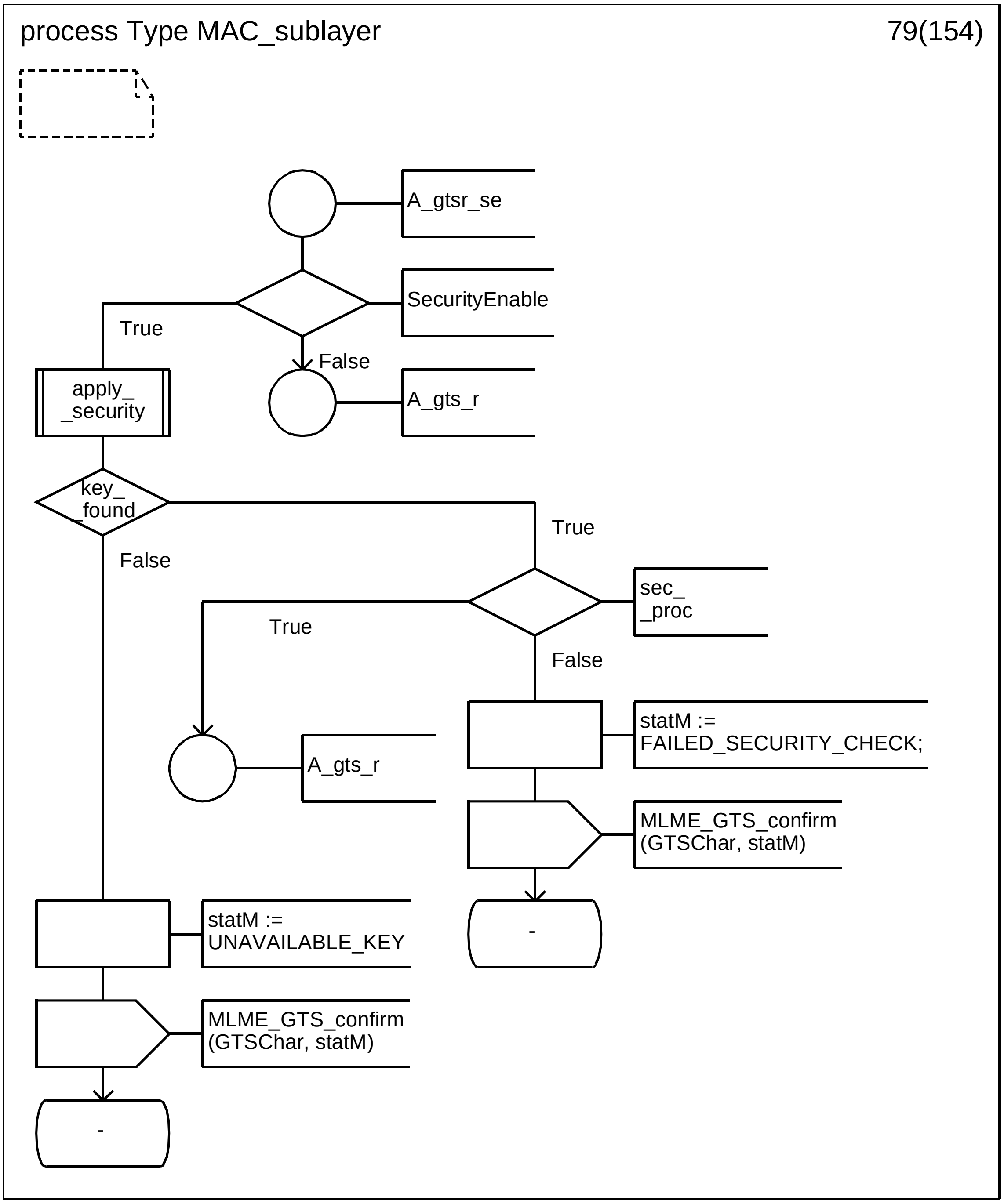
D.3.1.78 Process type MAC_sublayer (78)

.



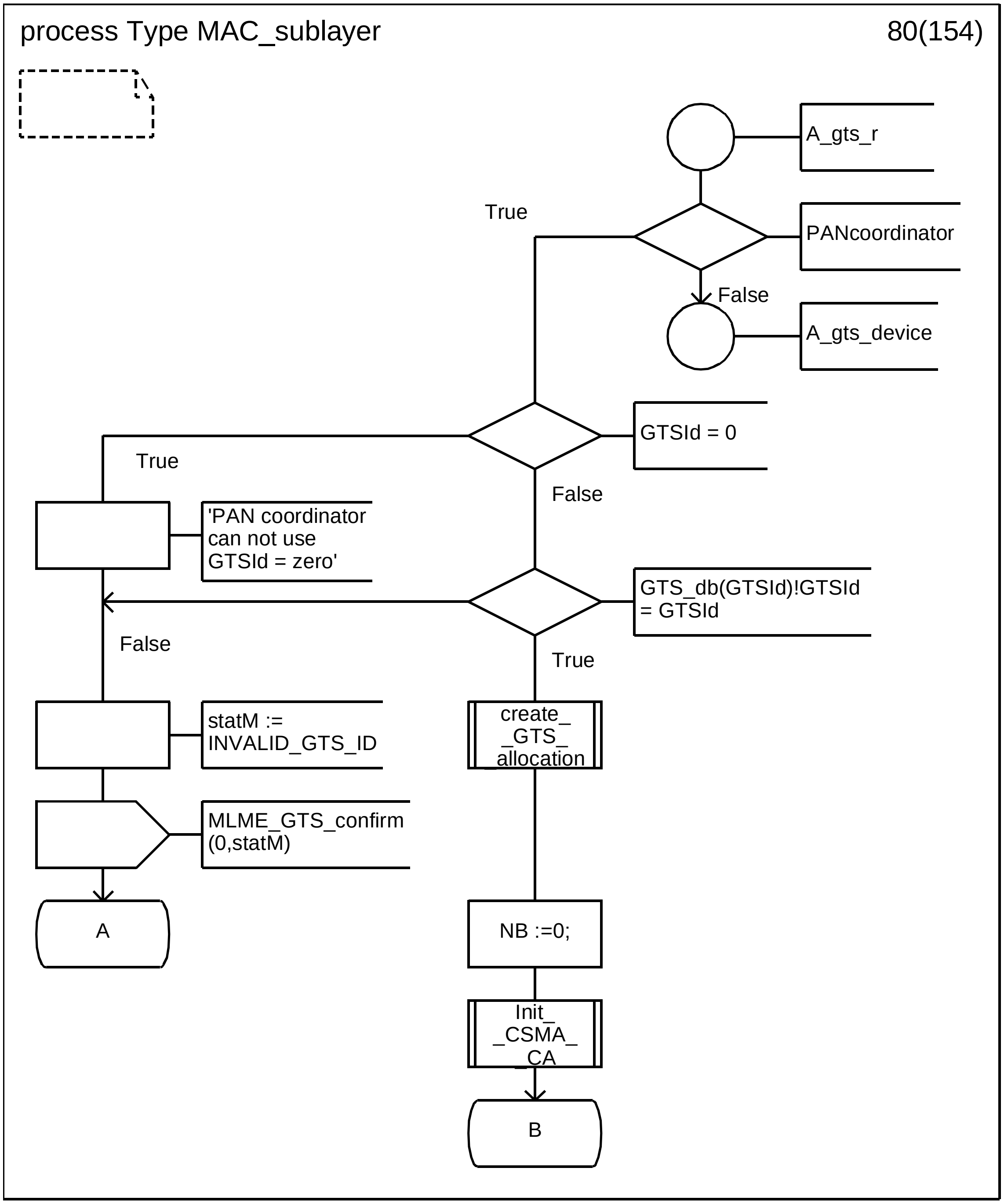
D.3.1.79 Process type MAC_sublayer (79)

.



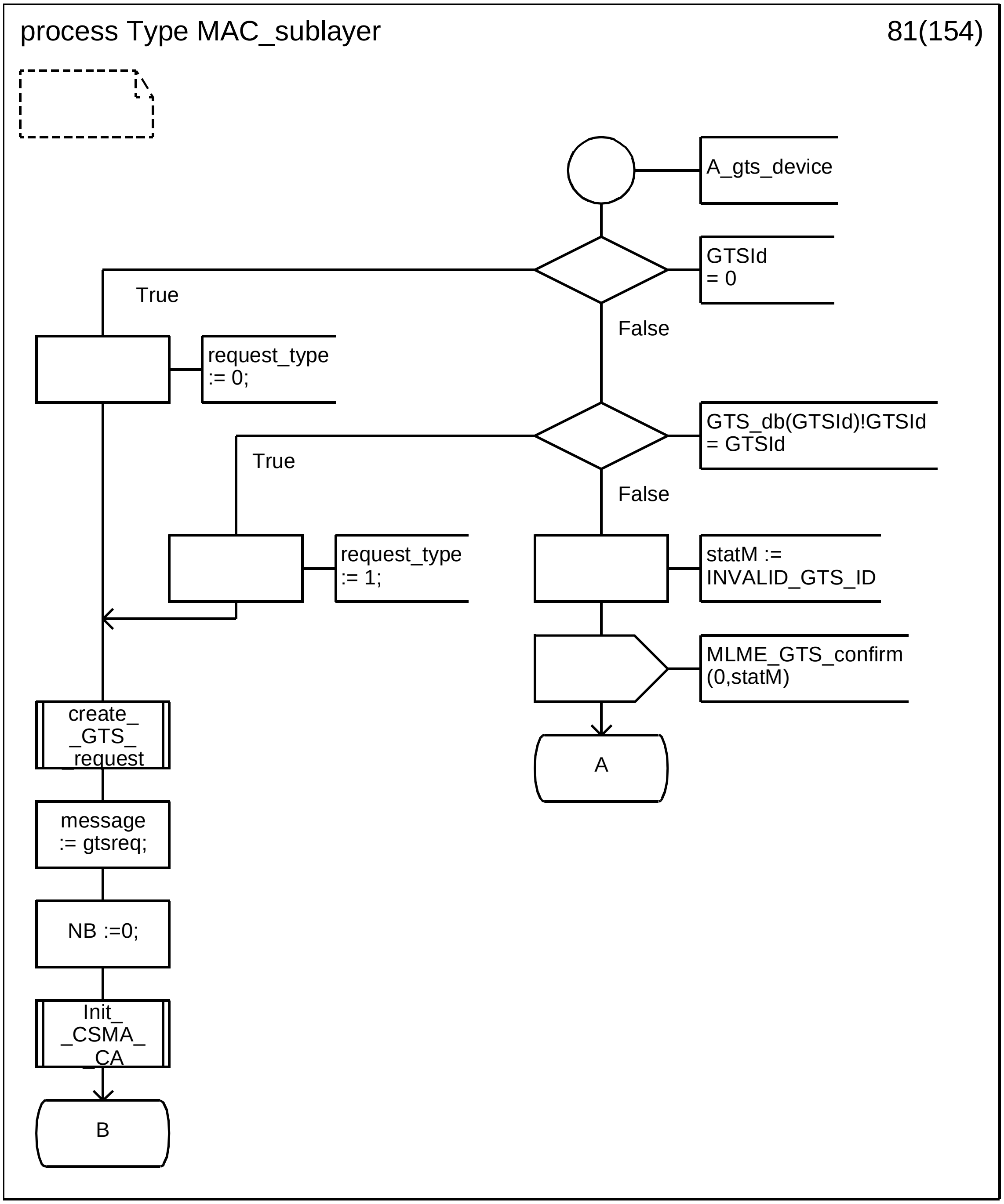
D.3.1.80 Process type MAC_sublayer (80)

.



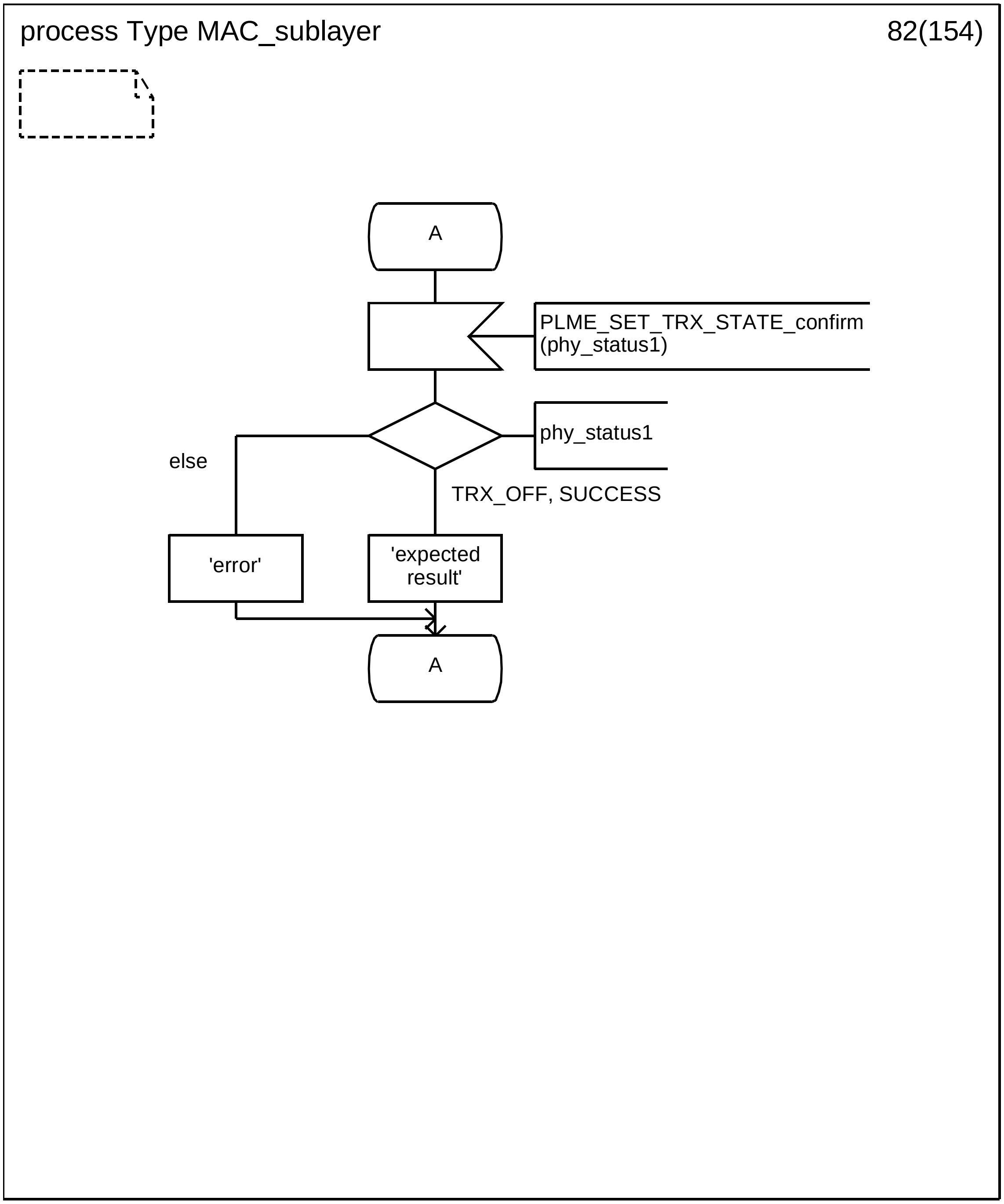
D.3.1.81 Process type MAC_sublayer (81)

.



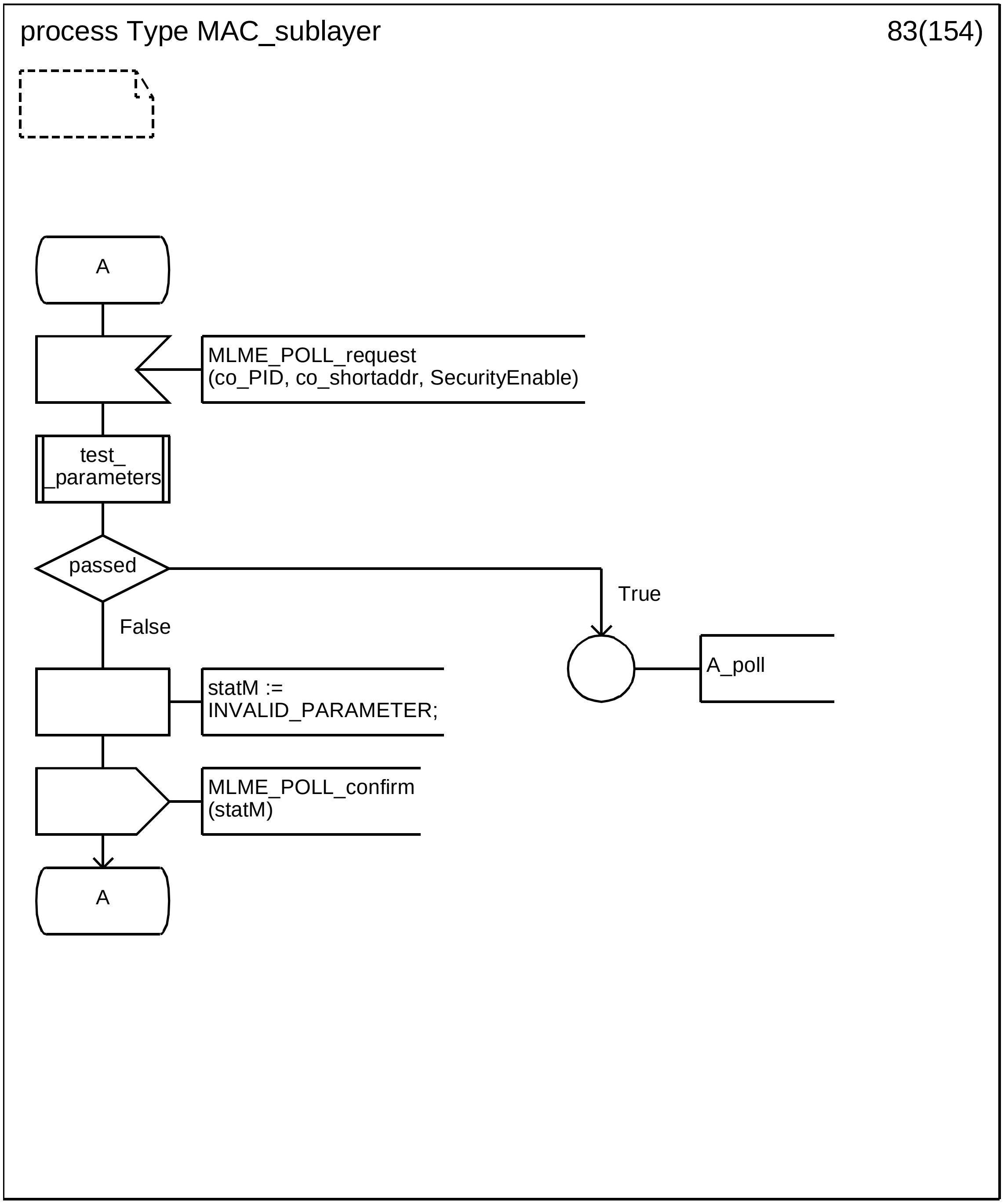
D.3.1.82 Process type MAC_sublayer (82)

.



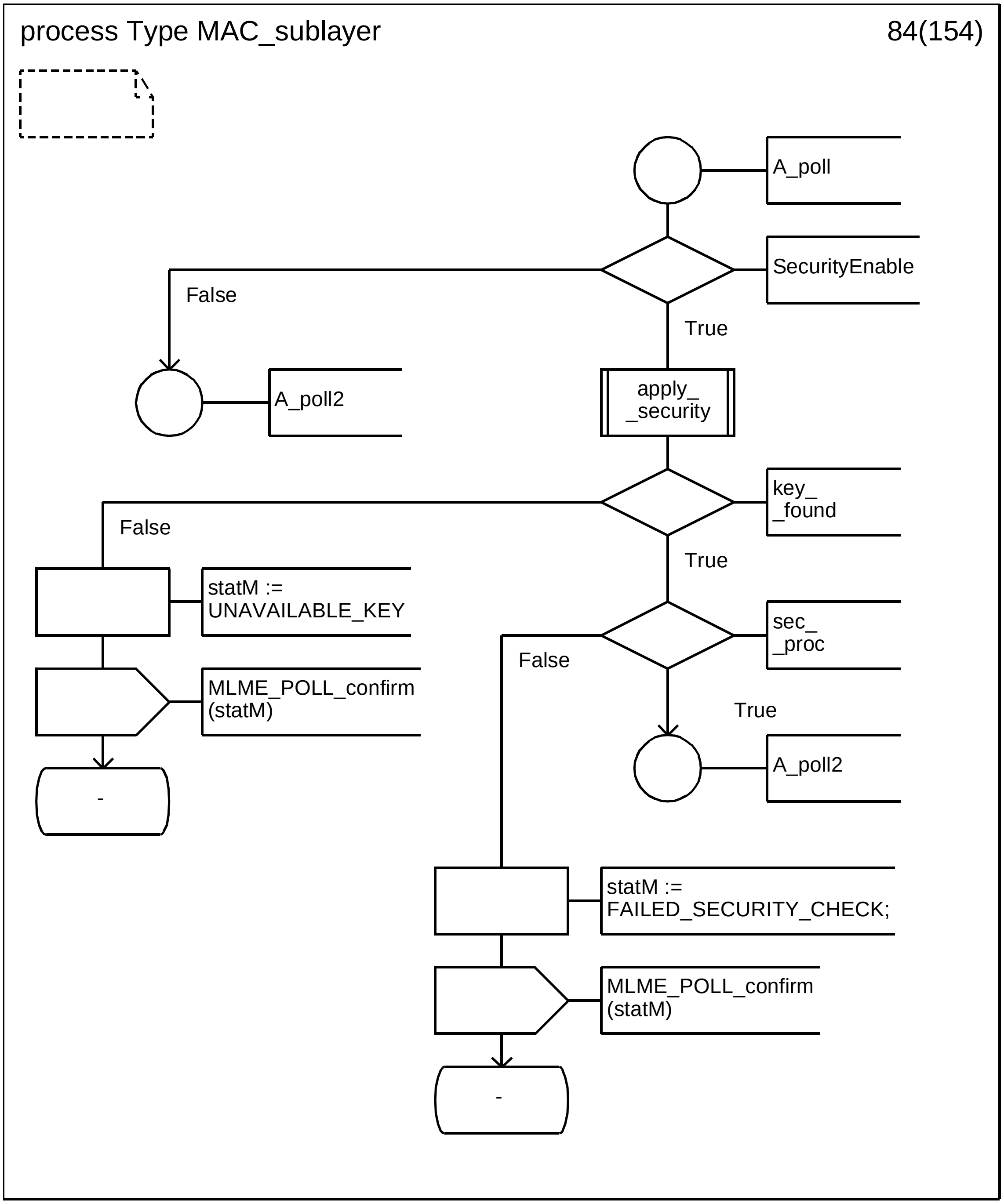
D.3.1.83 Process type MAC_sublayer (83)

.



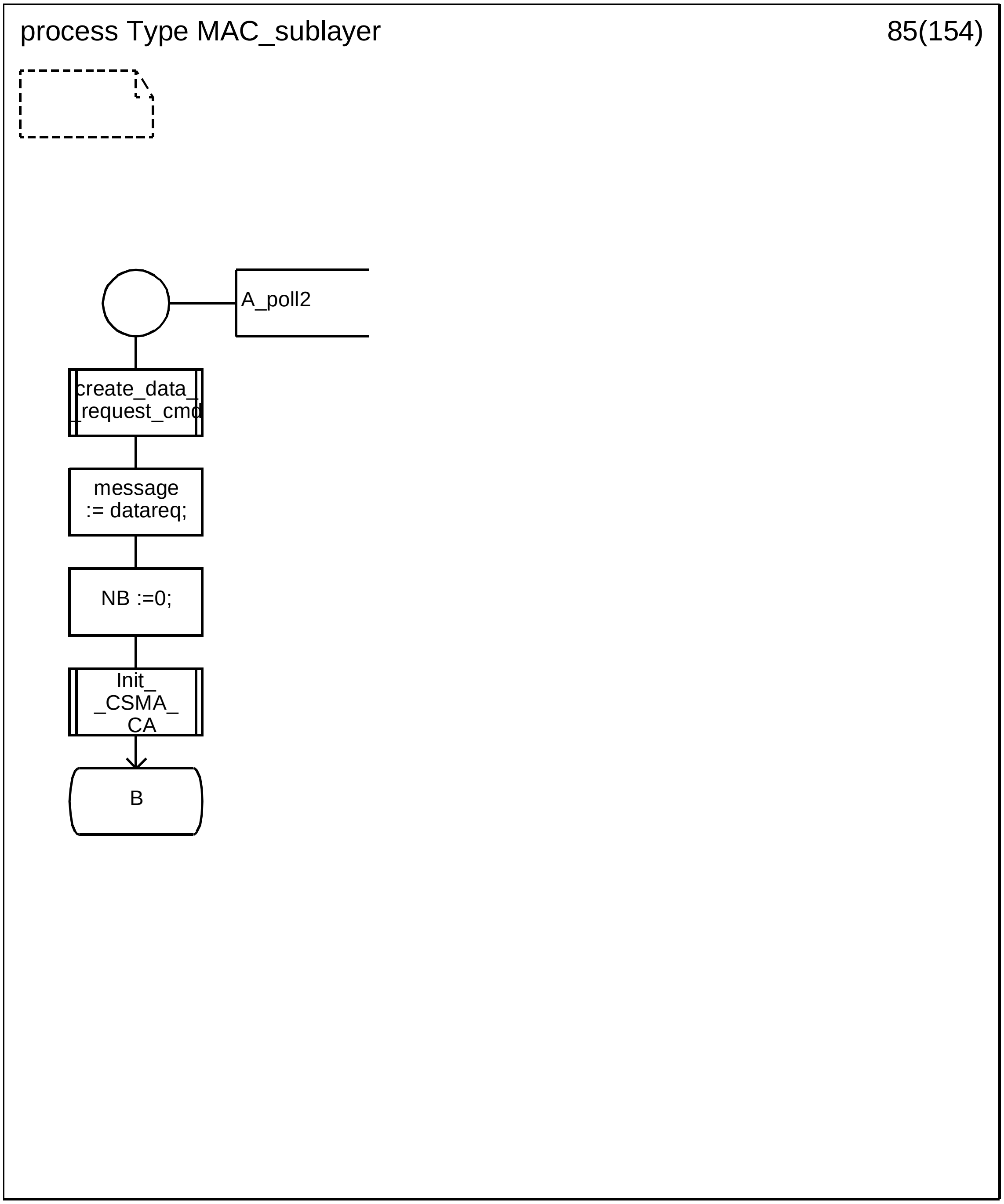
D.3.1.84 Process type MAC_sublayer (84)

.



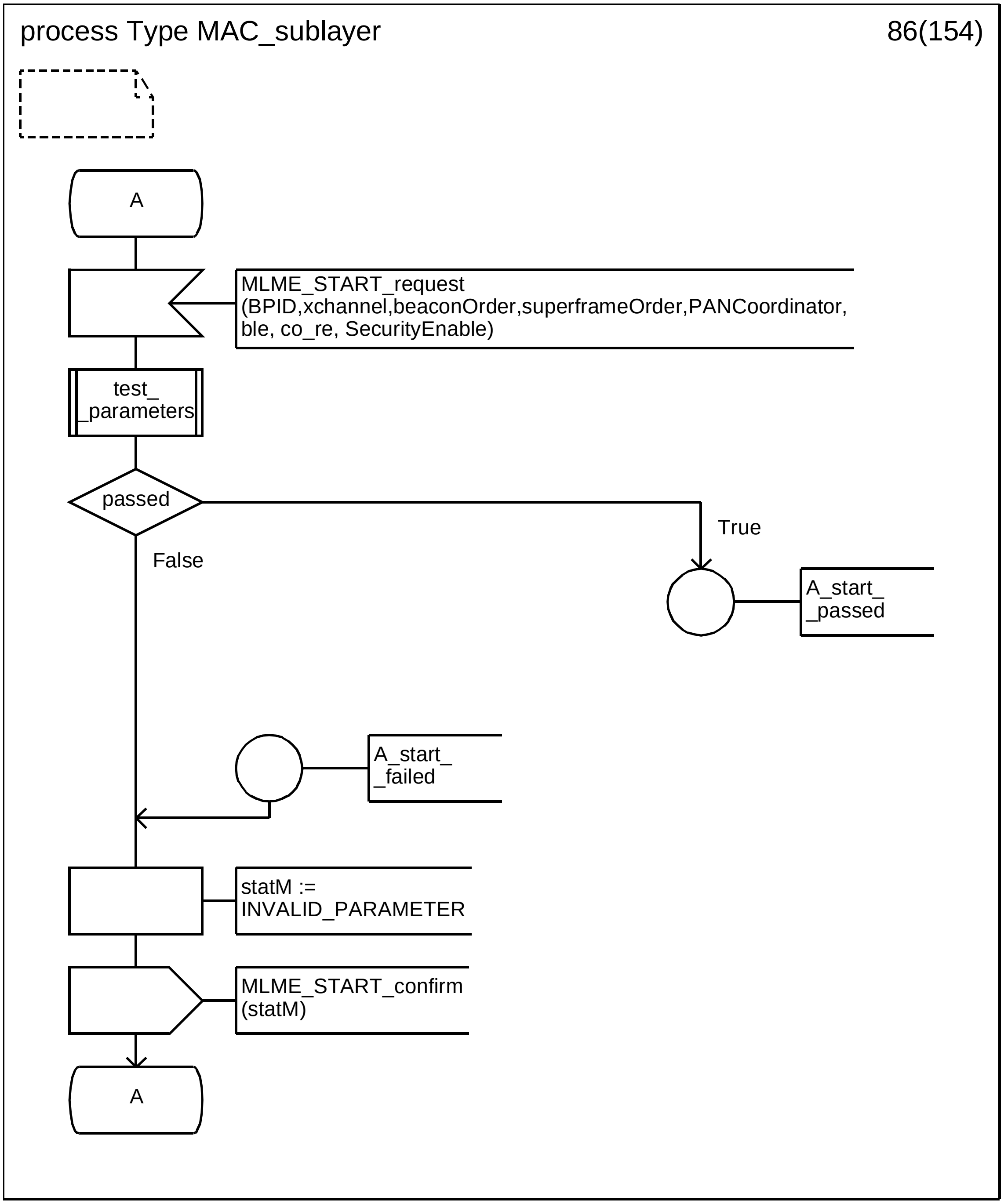
D.3.1.85 Process type MAC_sublayer (85)

.



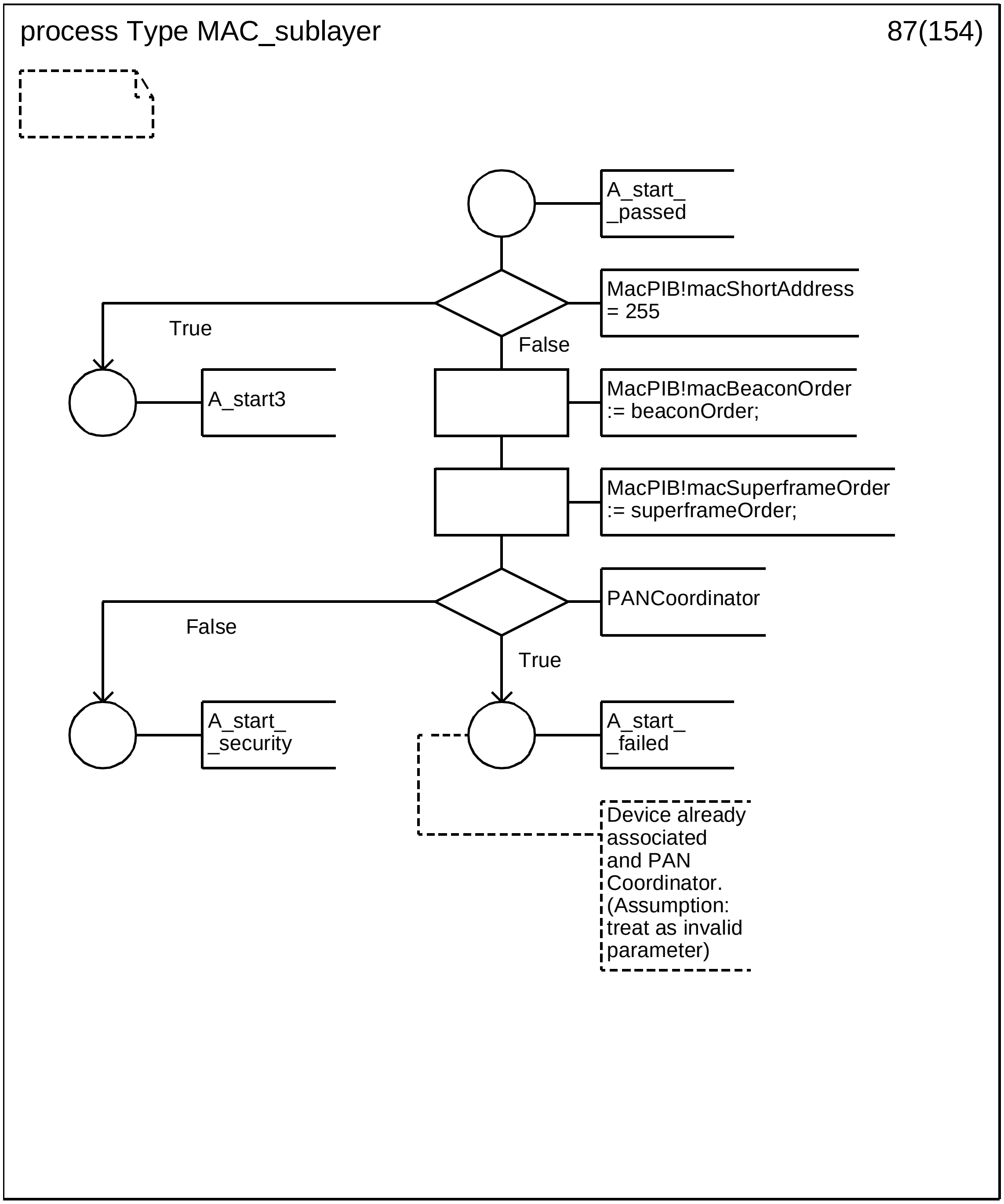
D.3.1.86 Process type MAC_sublayer (86)

.



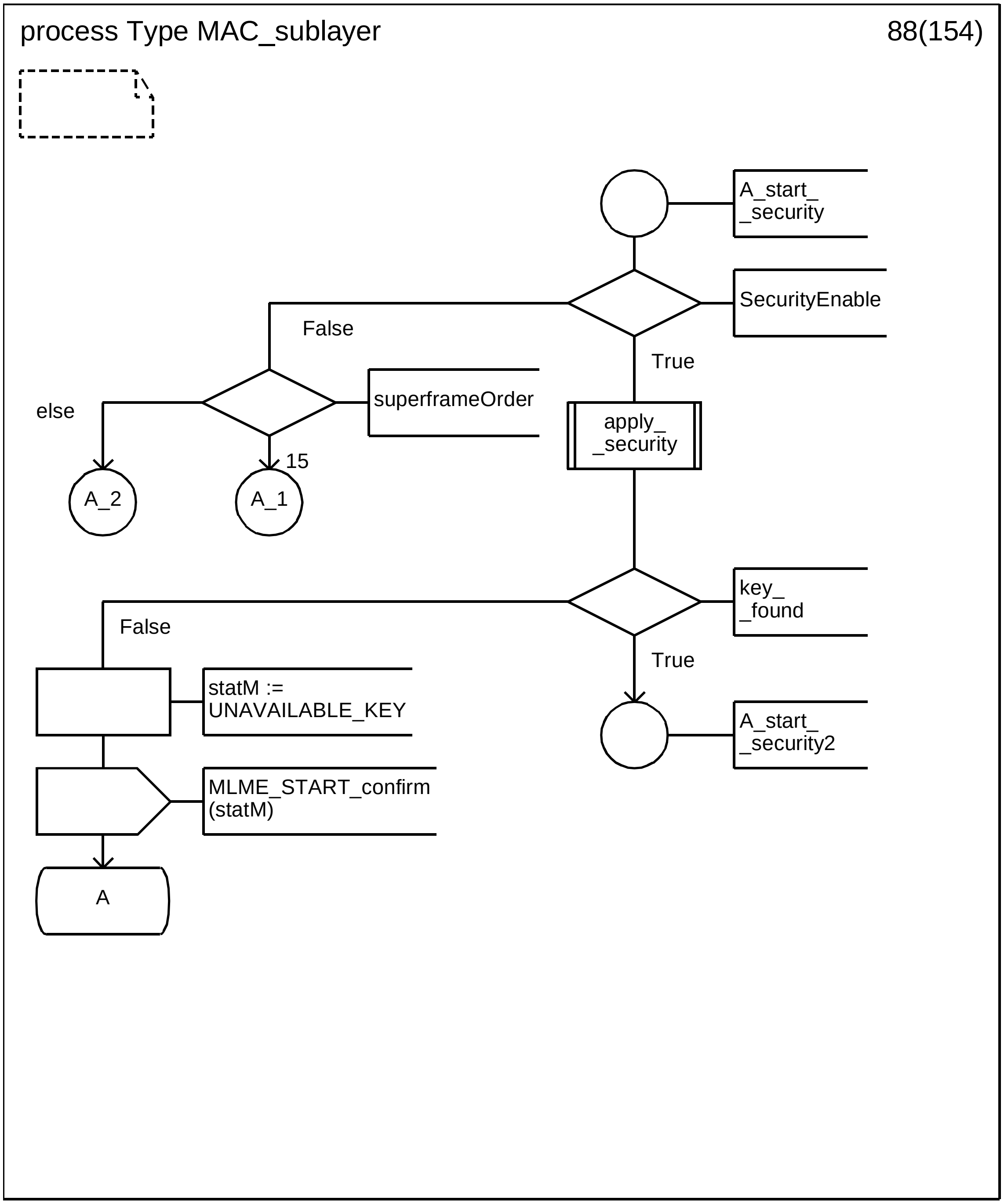
D.3.1.87 Process type MAC_sublayer (87)

.

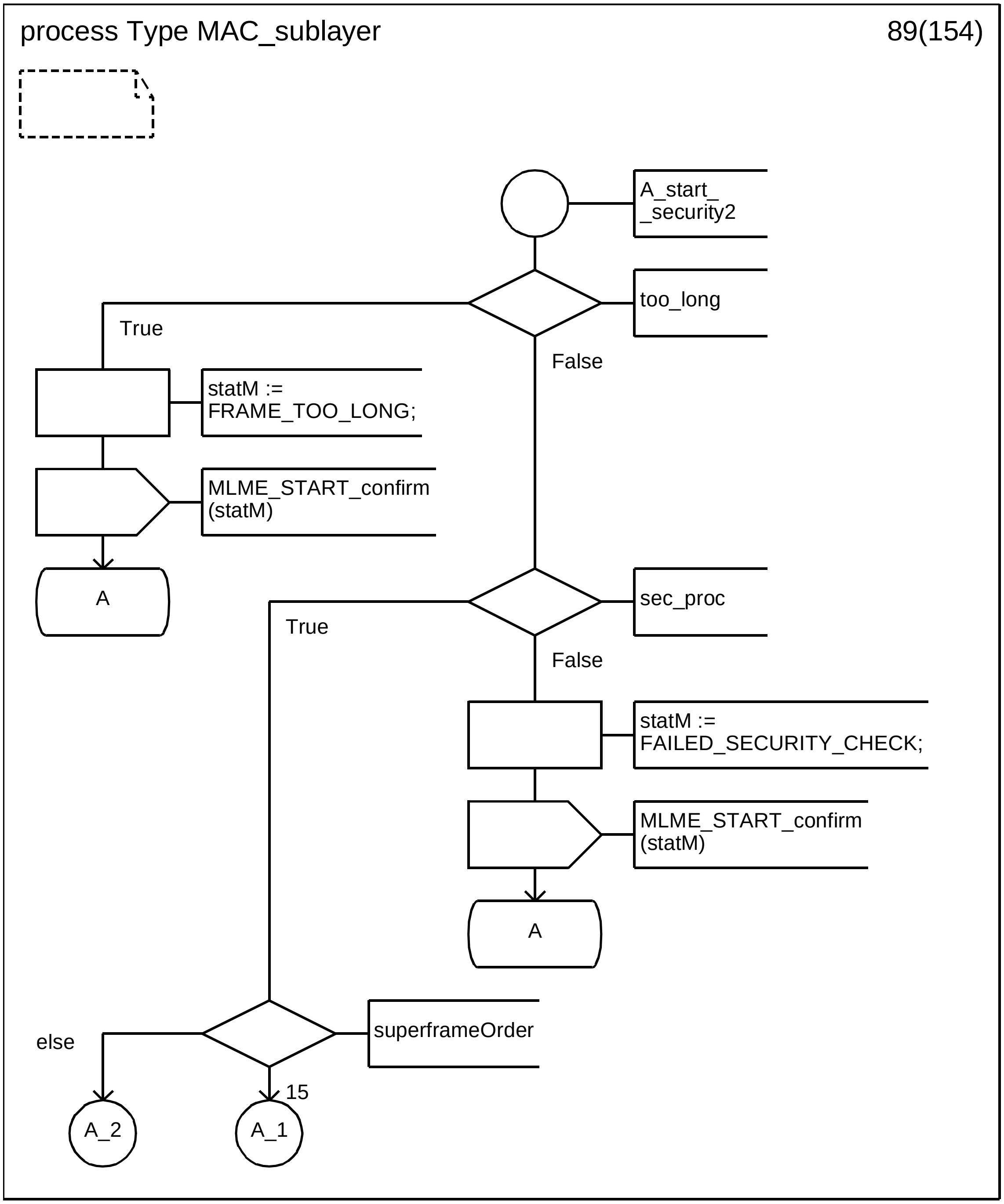


D.3.1.88 Process type MAC_sublayer (88)

.

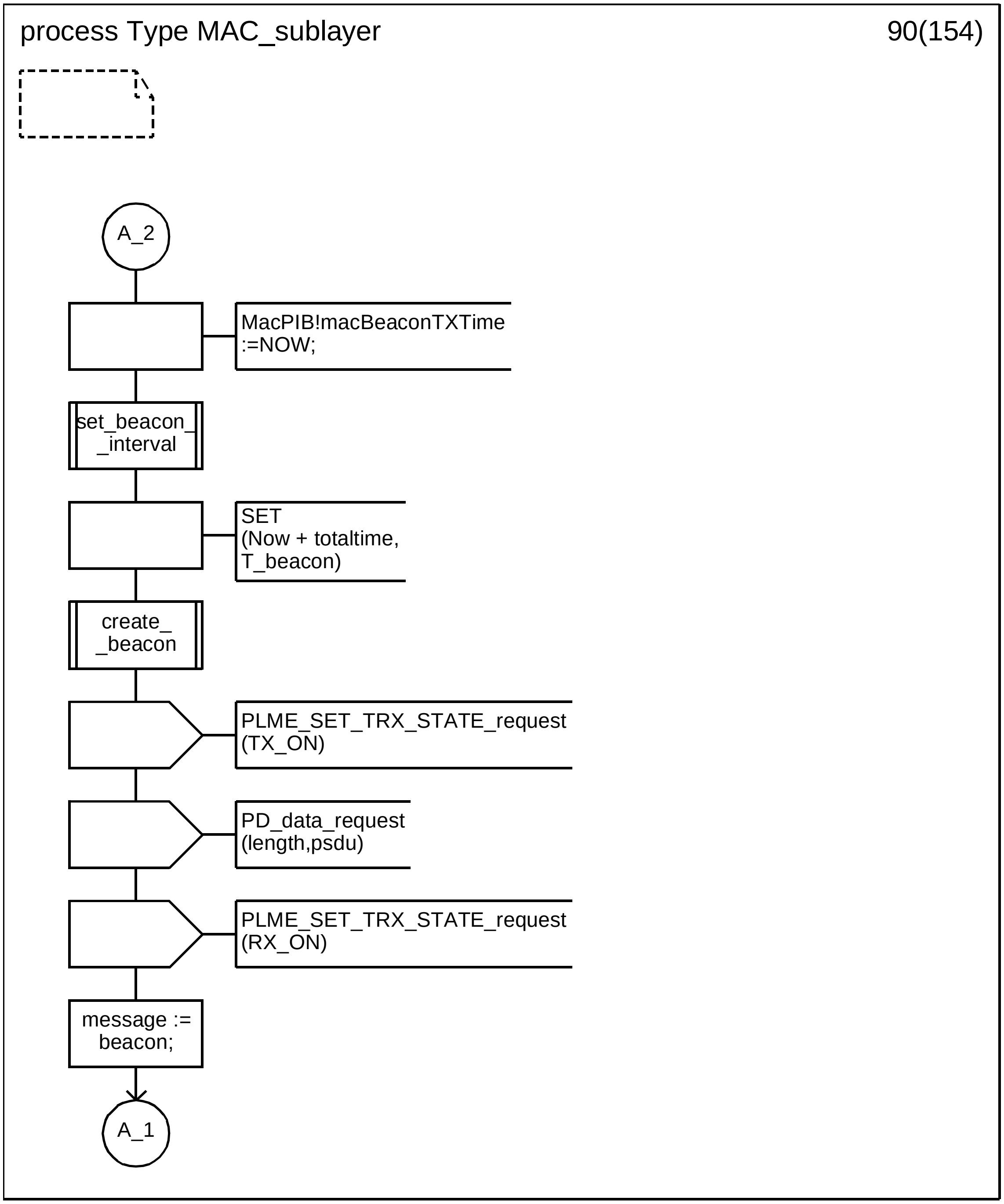


D.3.1.89 Process type MAC_sublayer (89)



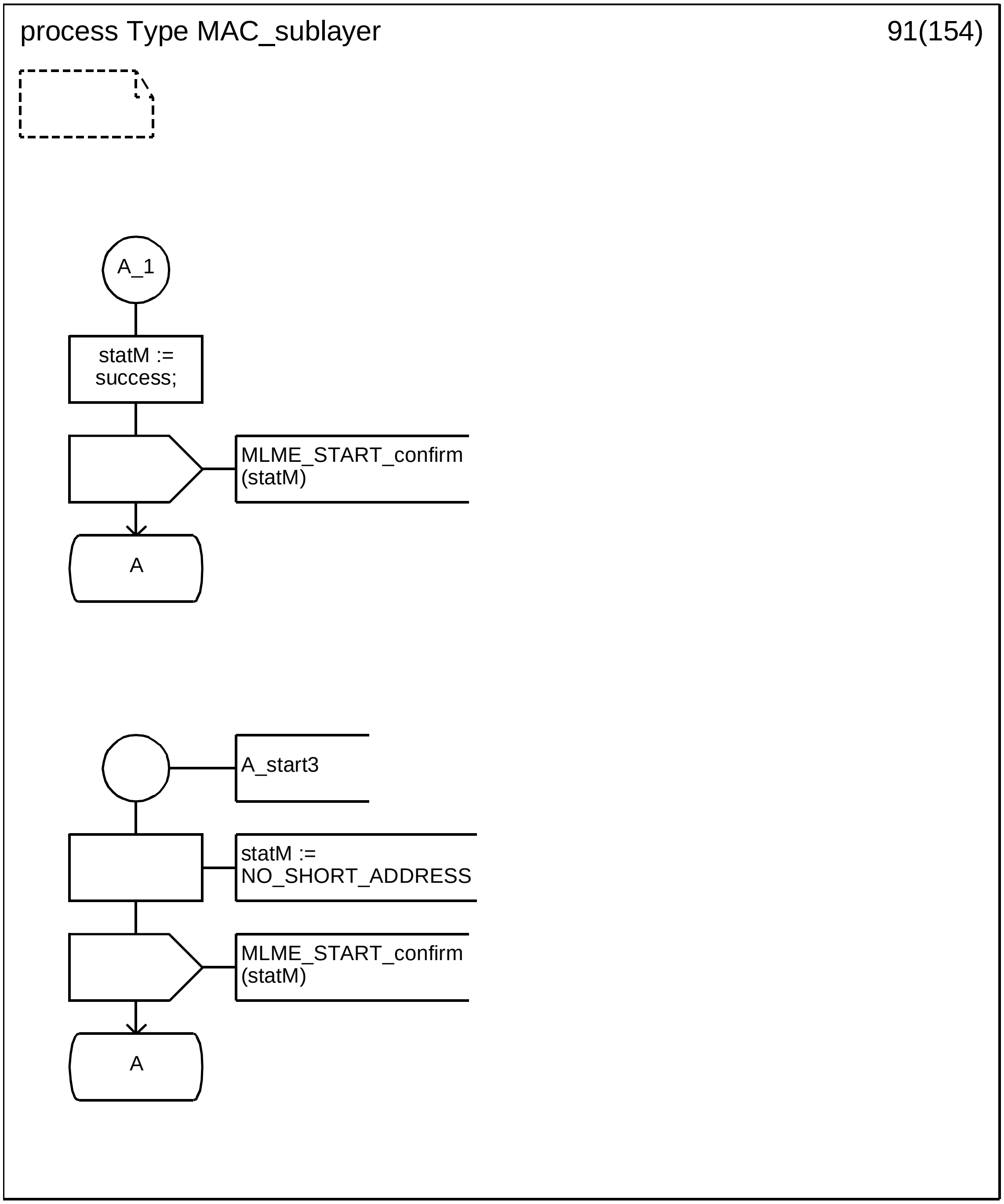
D.3.1.90 Process type MAC_sublayer (90)

.



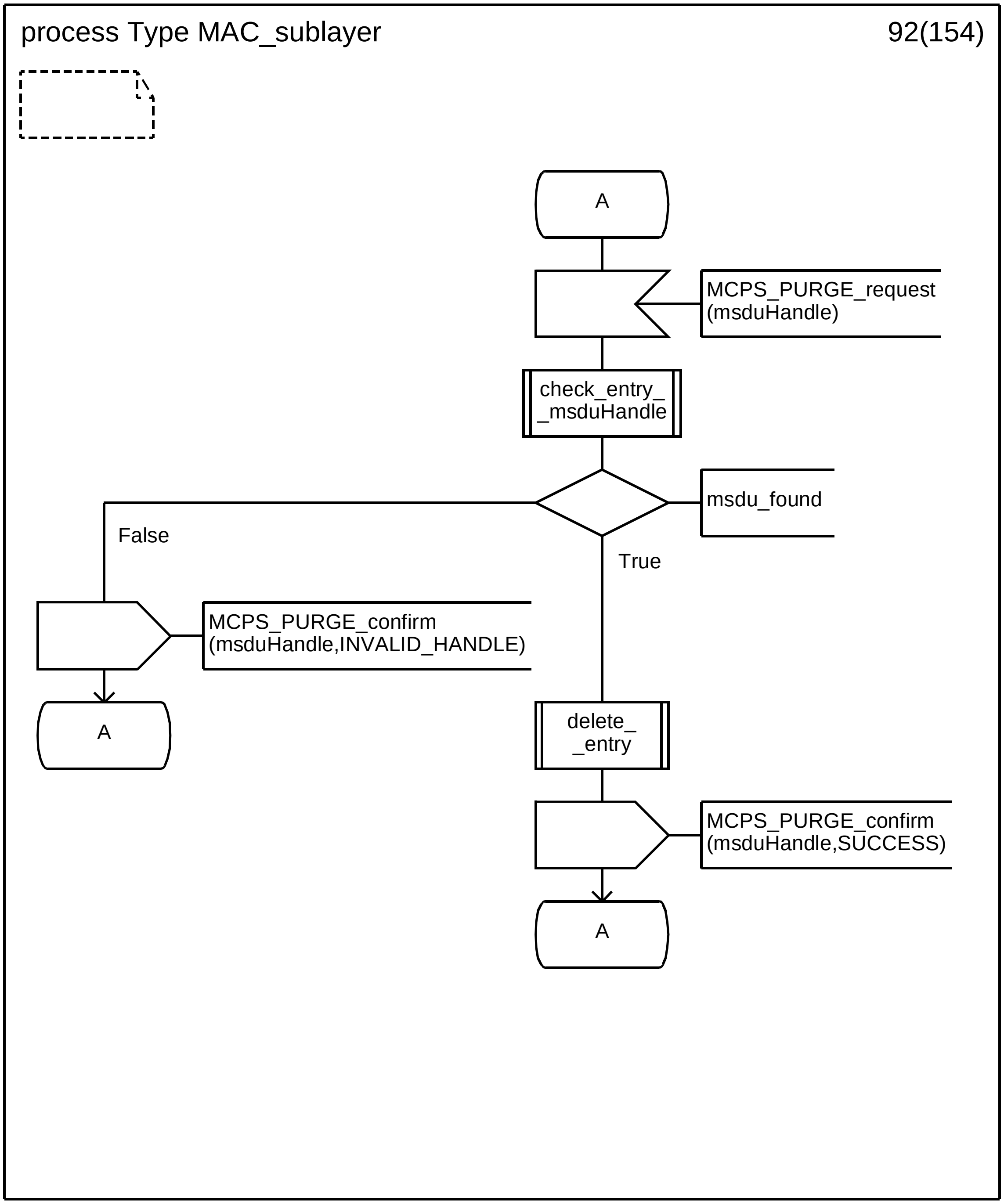
D.3.1.91 Process type MAC_sublayer (91)

.


process Type MAC_sublayer
91(154)
A_1
statM := success;
MLME_START_confirm (statM)
A
A_start3
statM := NO_SHORT_ADDRESS
MLME_START_confirm (statM)
A

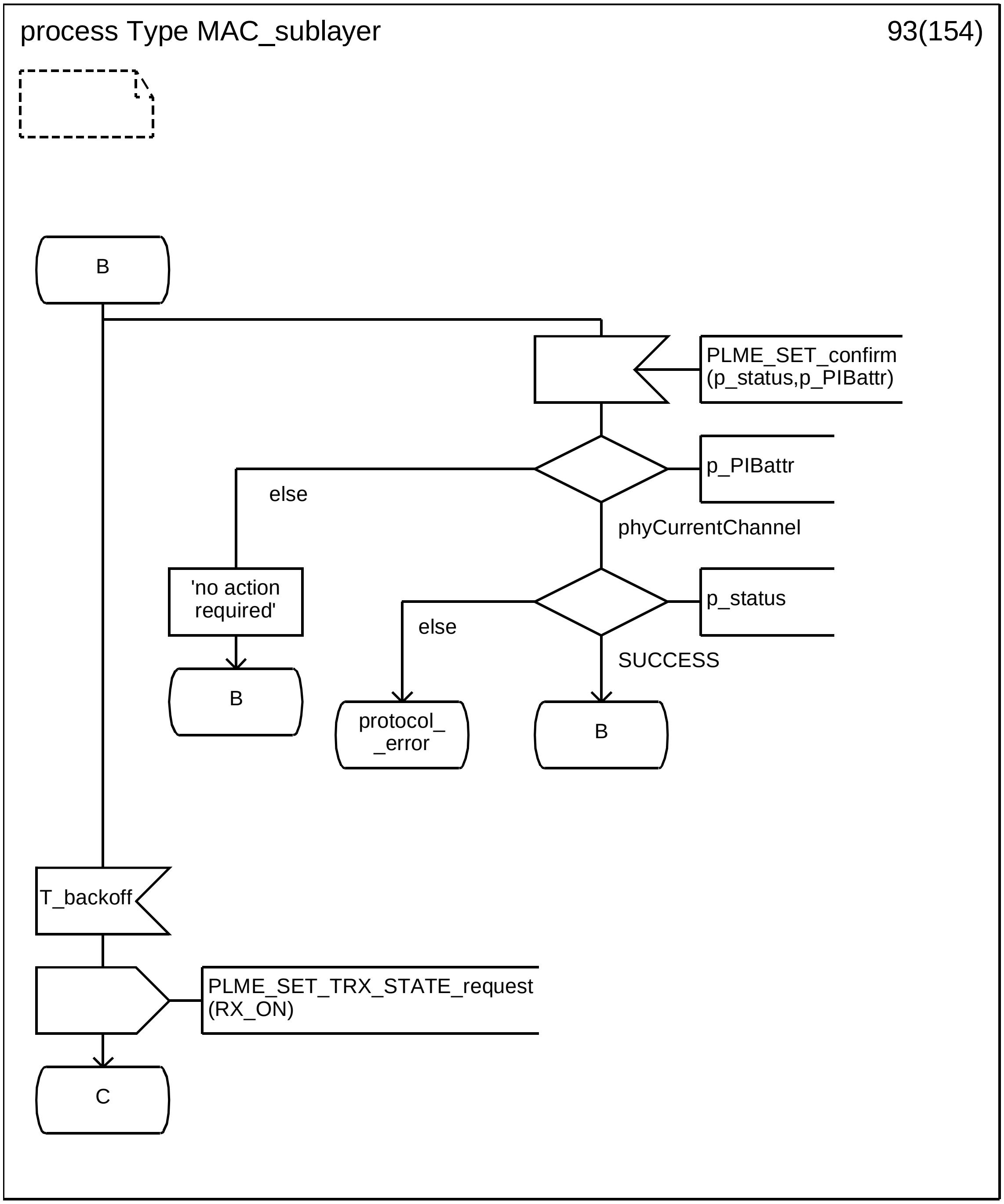
D.3.1.92 Process type MAC_sublayer (92)

.



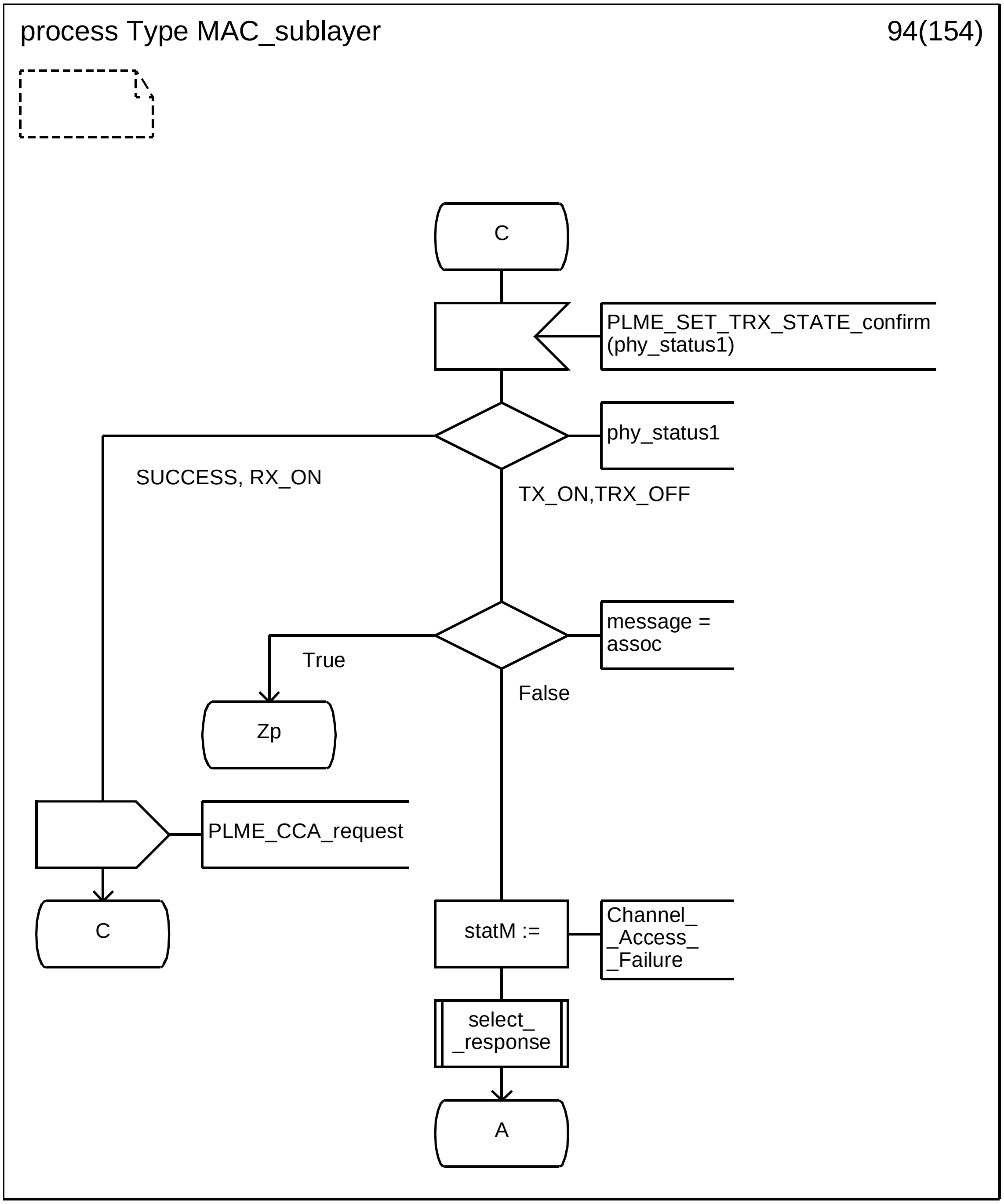
D.3.1.93 Process type MAC_sublayer (93)

.



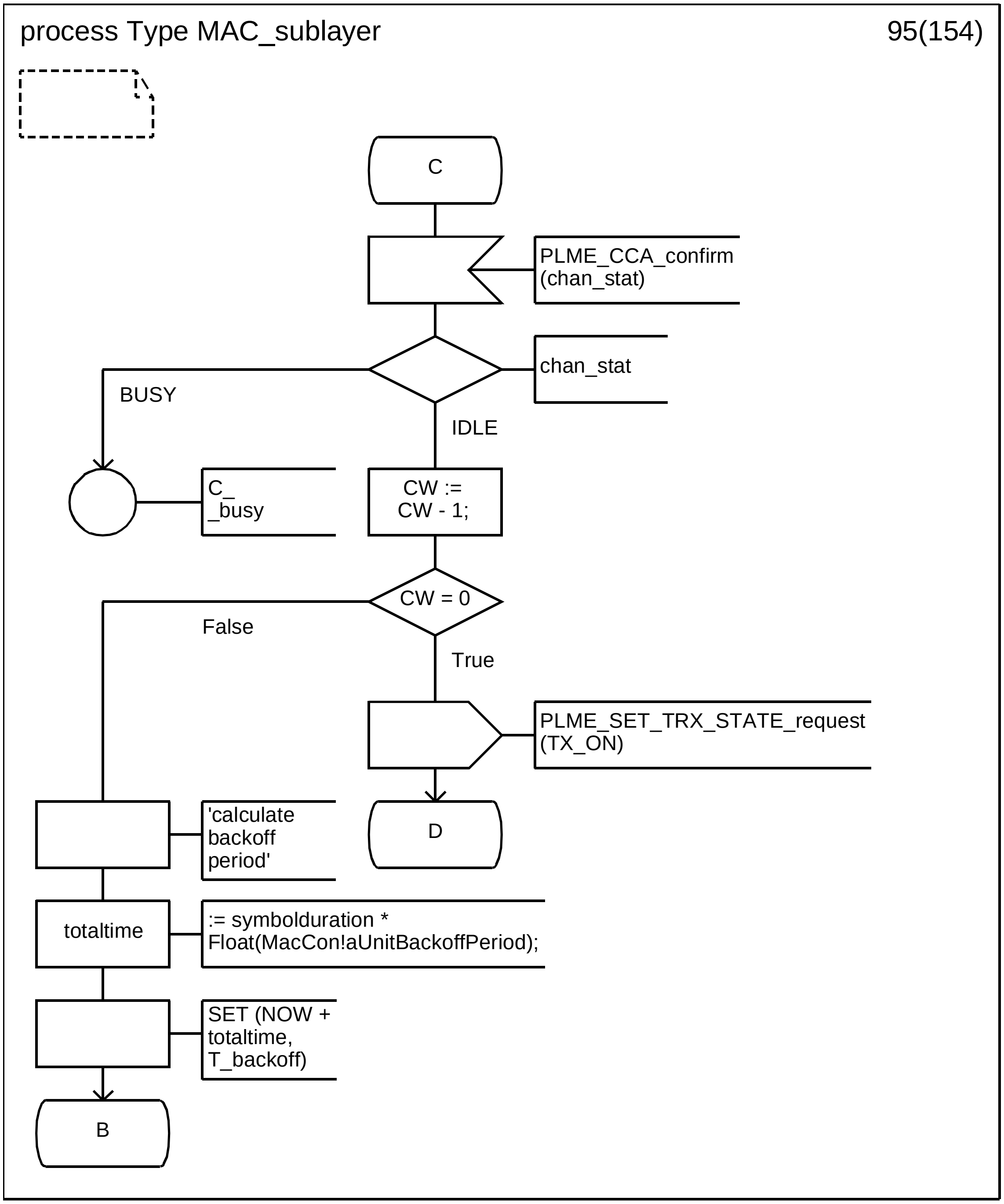
D.3.1.94 Process type MAC_sublayer (94)

.



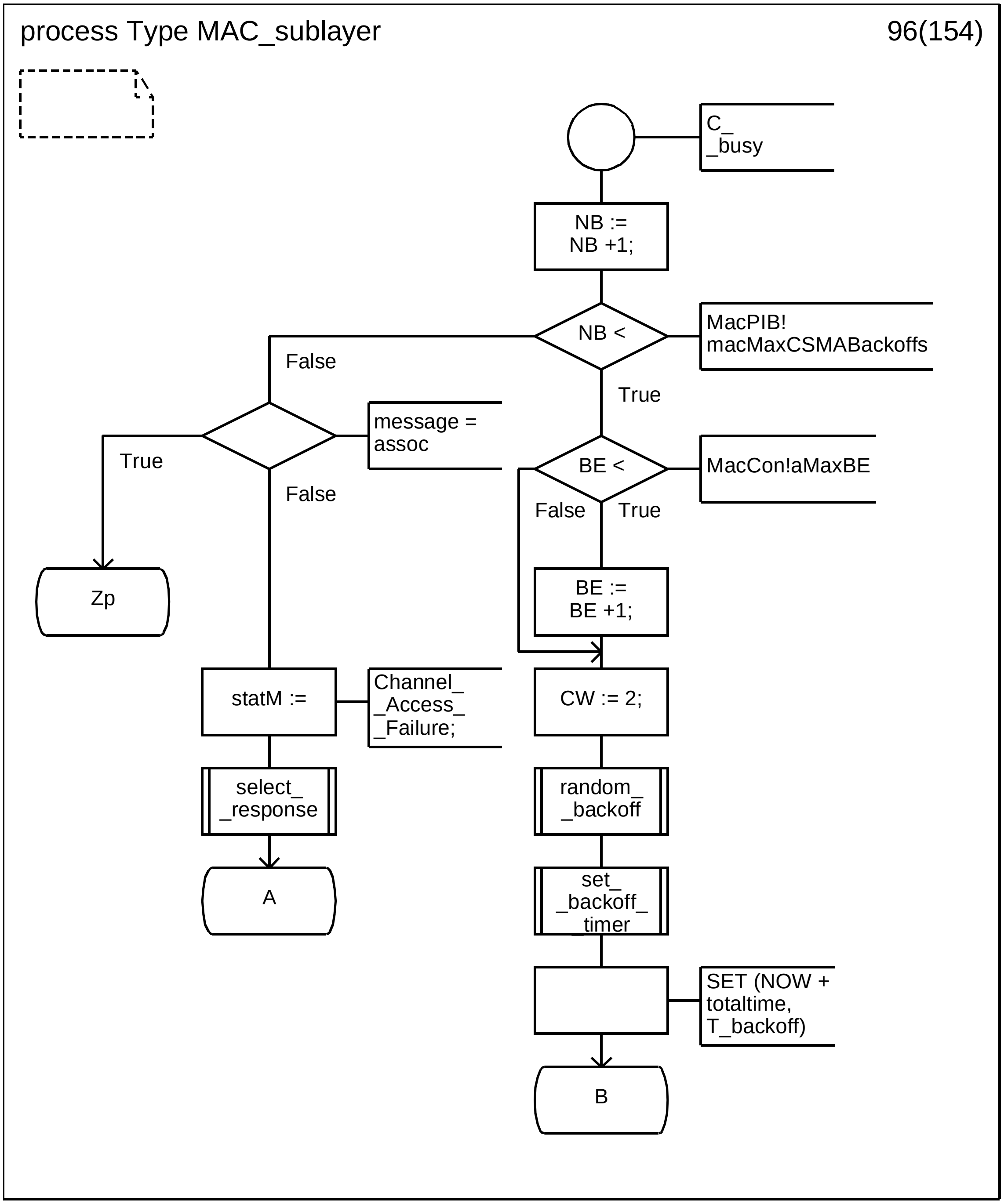
D.3.1.95 Process type MAC_sublayer (95)

.



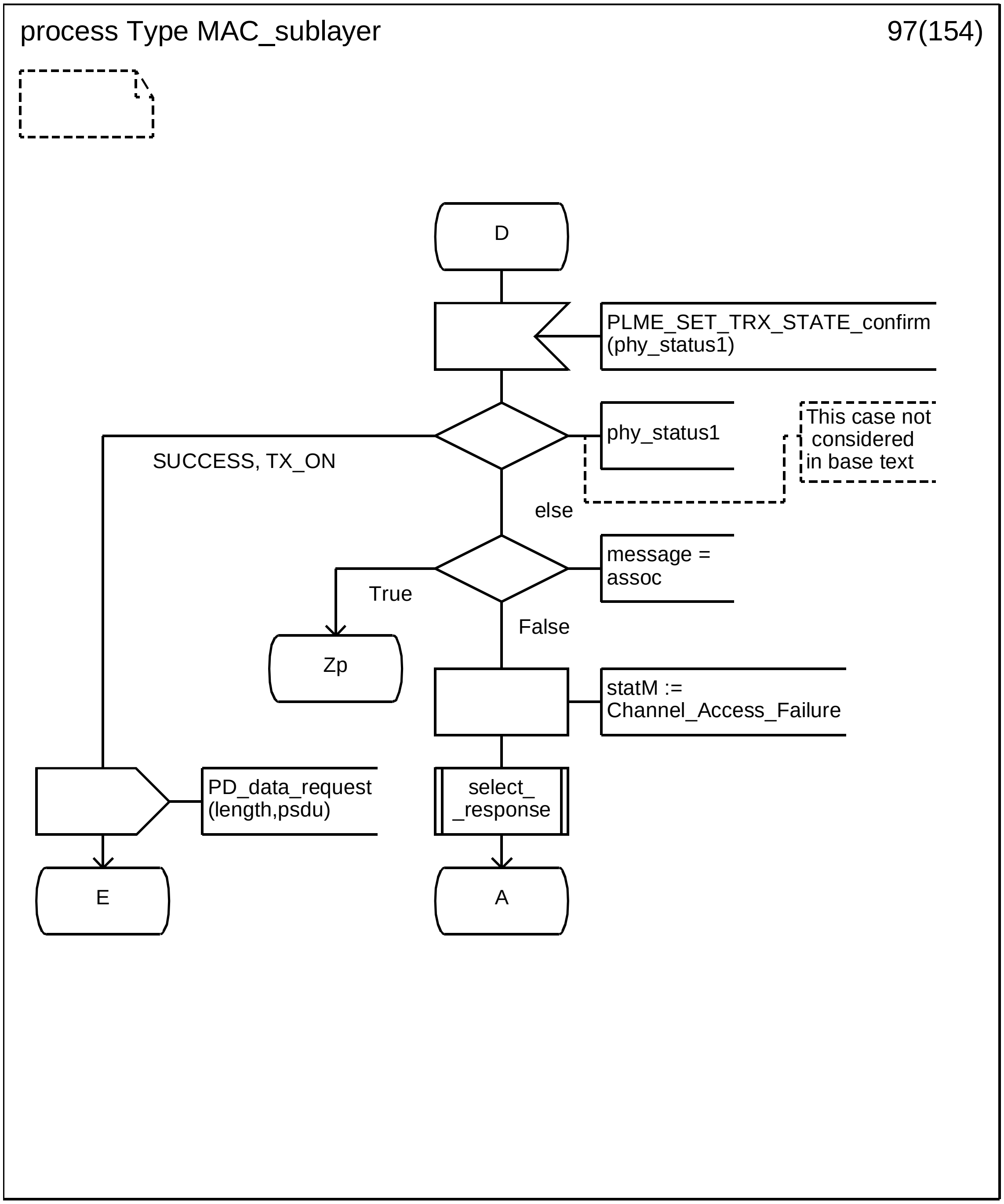
D.3.1.96 Process type MAC_sublayer (96)

.

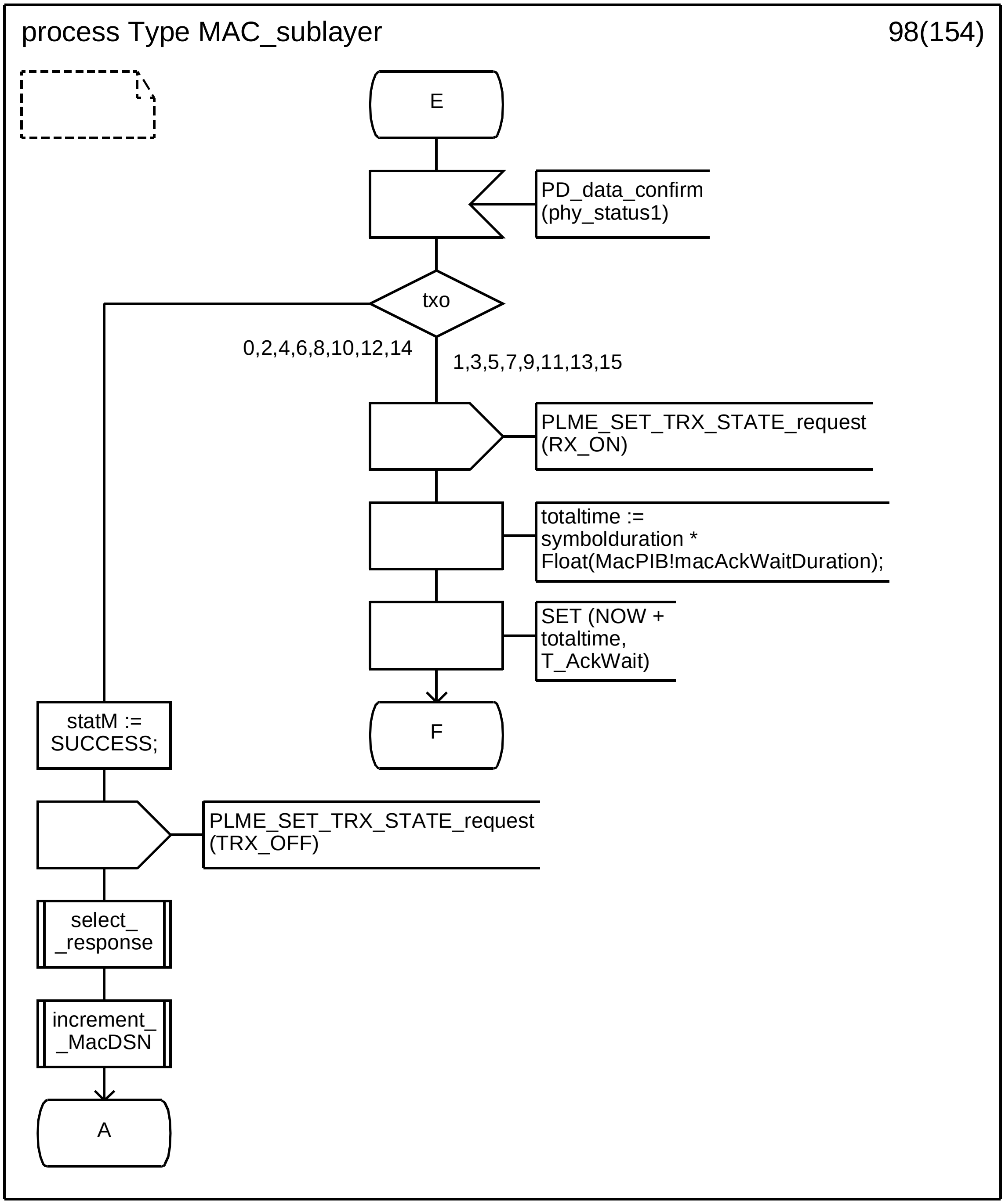


D.3.1.97 Process type MAC_sublayer (97)

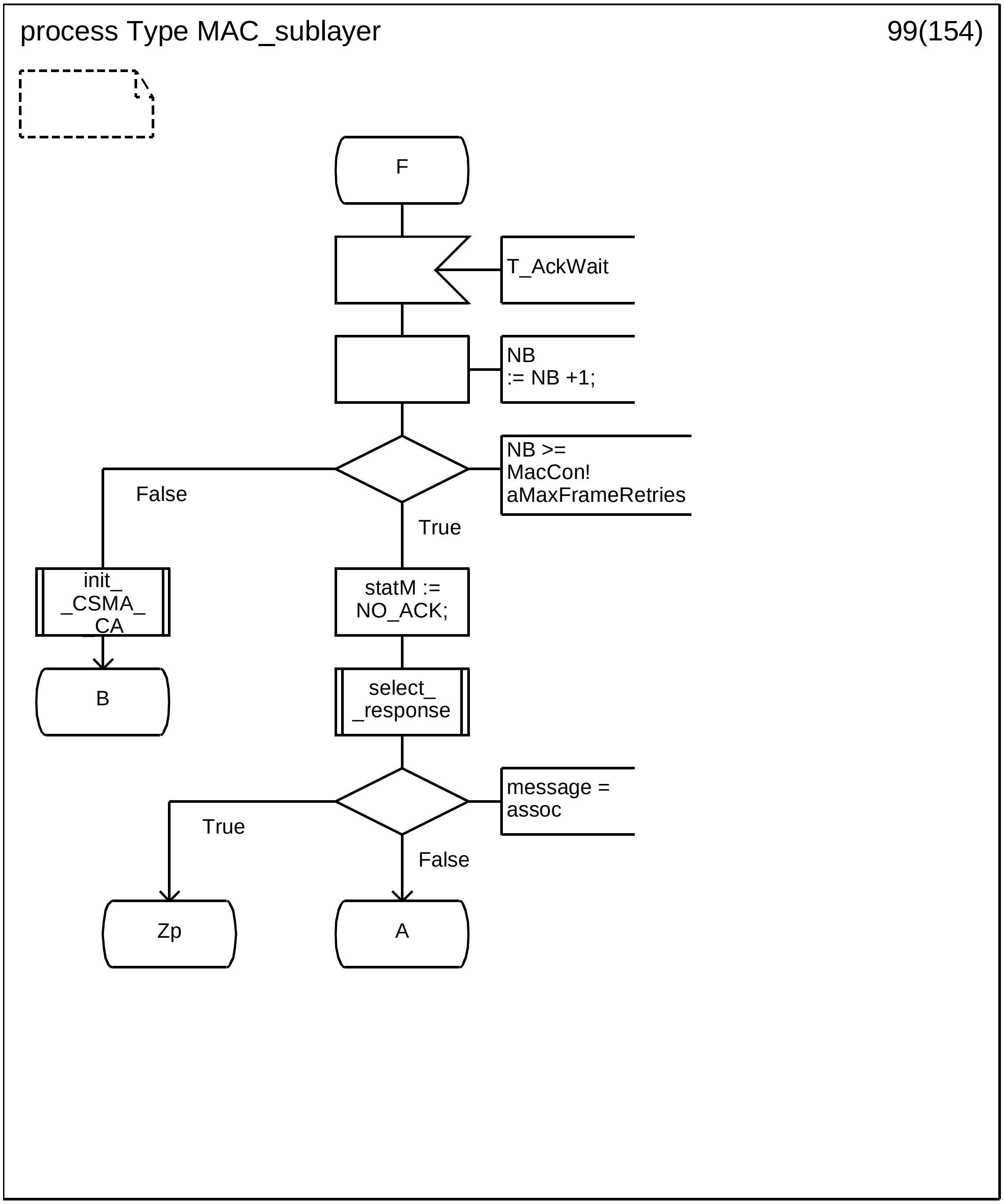
.



D.3.1.98 Process type MAC_sublayer (98)

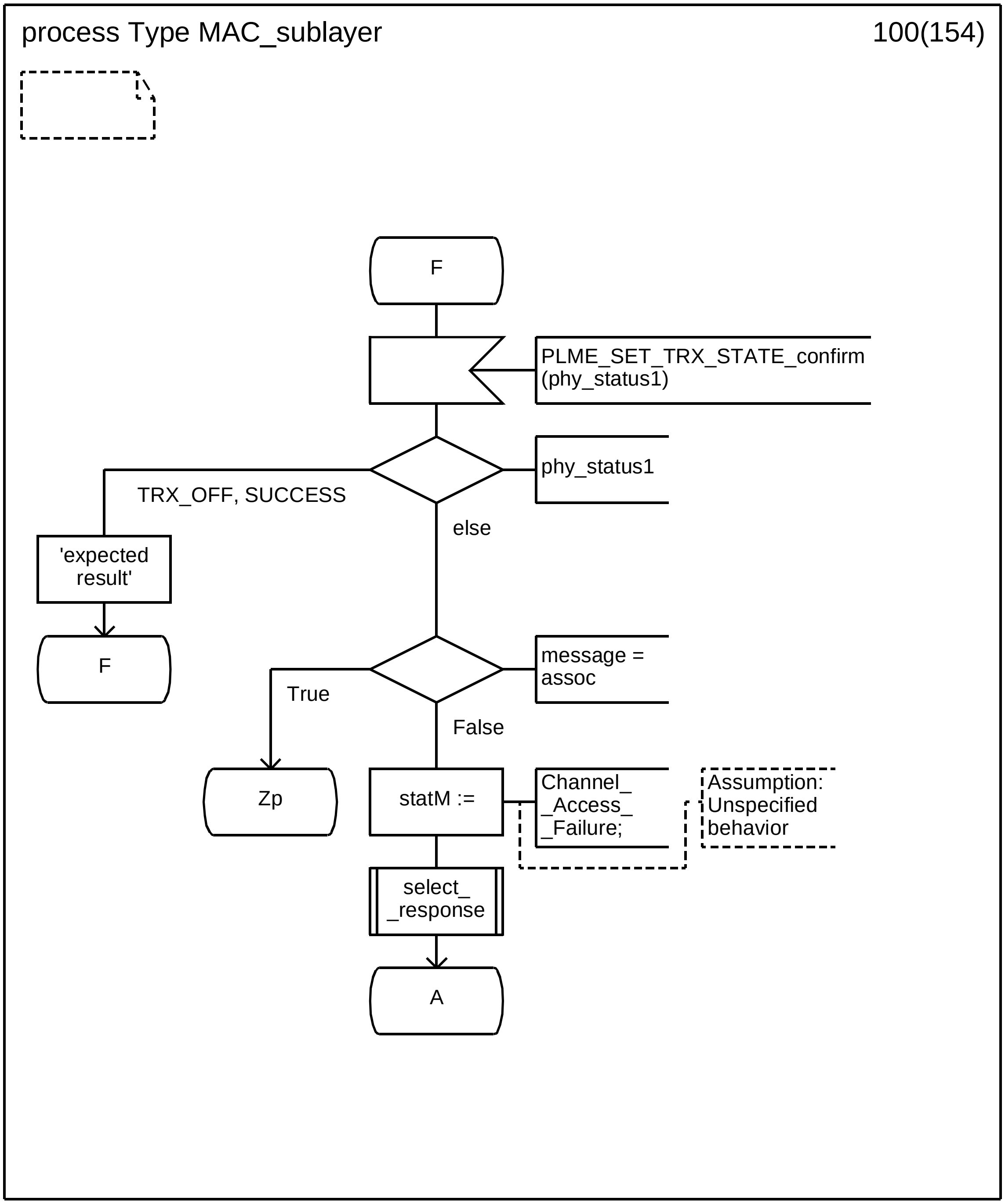


D.3.1.99 Process type MAC_sublayer (99)



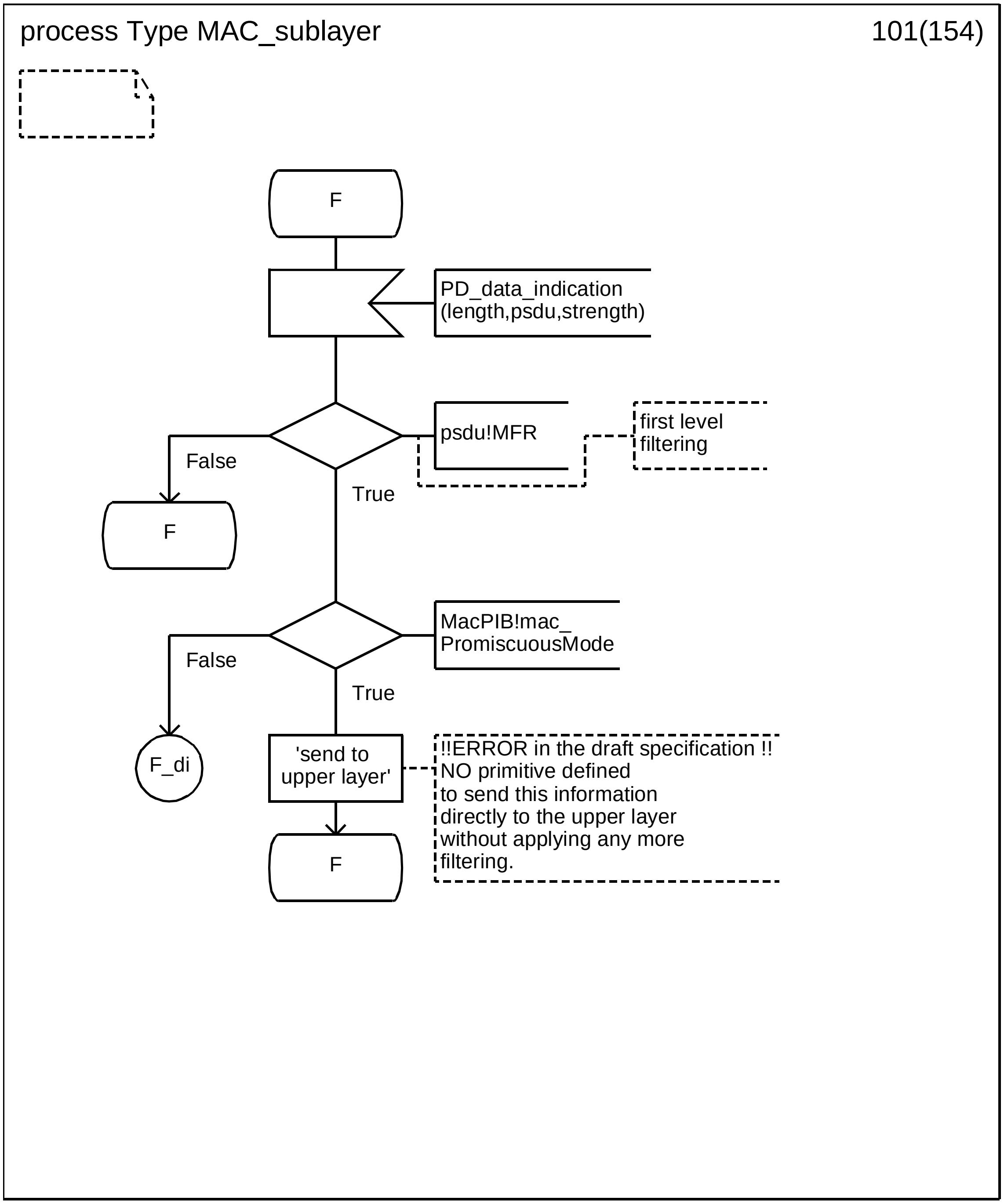
D.3.1.100 Process type MAC_sublayer (100)

.



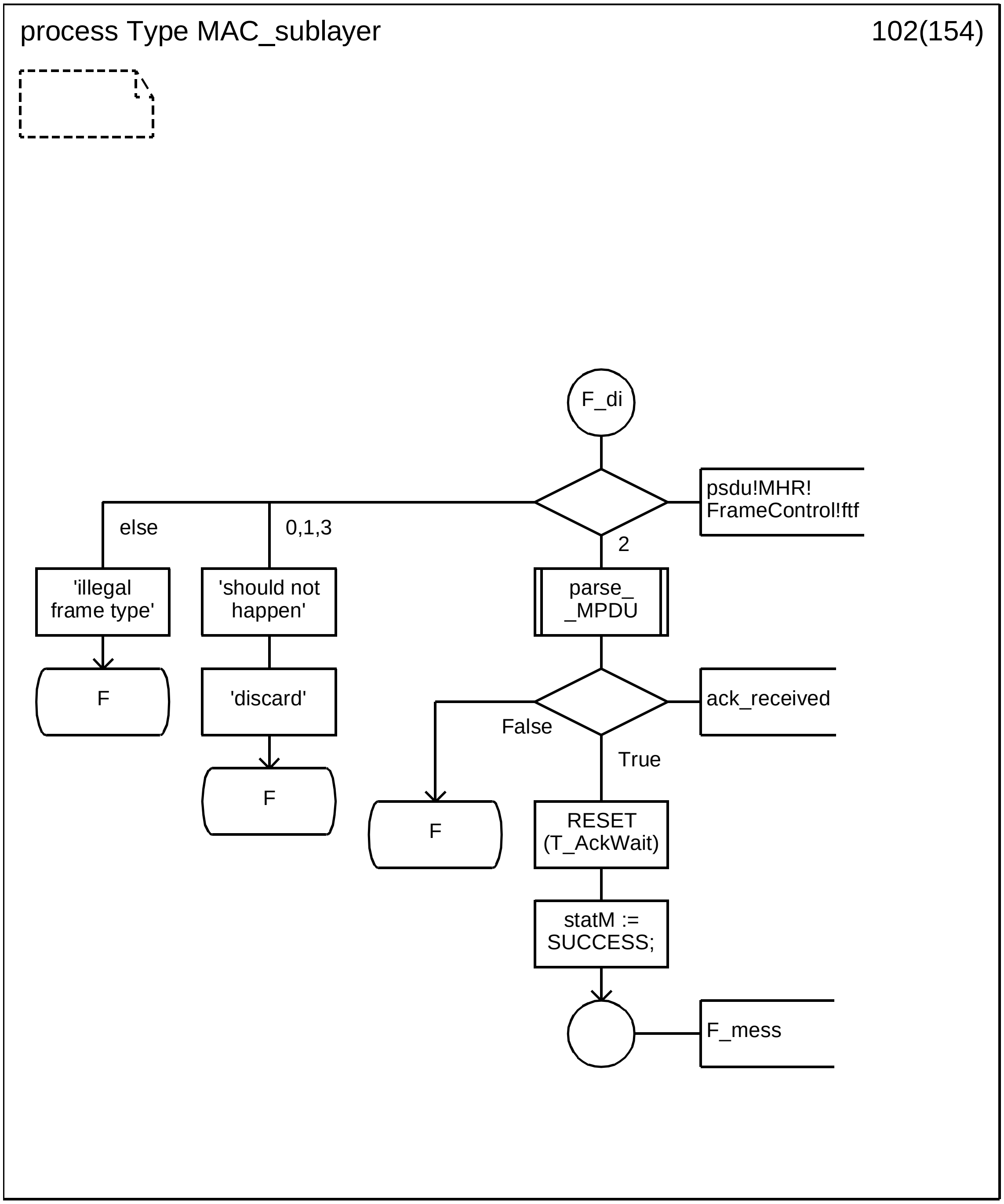
D.3.1.101 Process type MAC_sublayer (101)

.


process Type MAC_sublayer
101(154)
F
PD_data_indication (length,psdu,strength)
psdu!MFR
first level filtering
False
True
F
MacPIB!mac_ PromiscuousMode
False
True
F_di
'send to upper layer'
!!ERROR in the draft specification !! NO primitive defined to send this information directly to the upper layer without applying any more filtering.
F

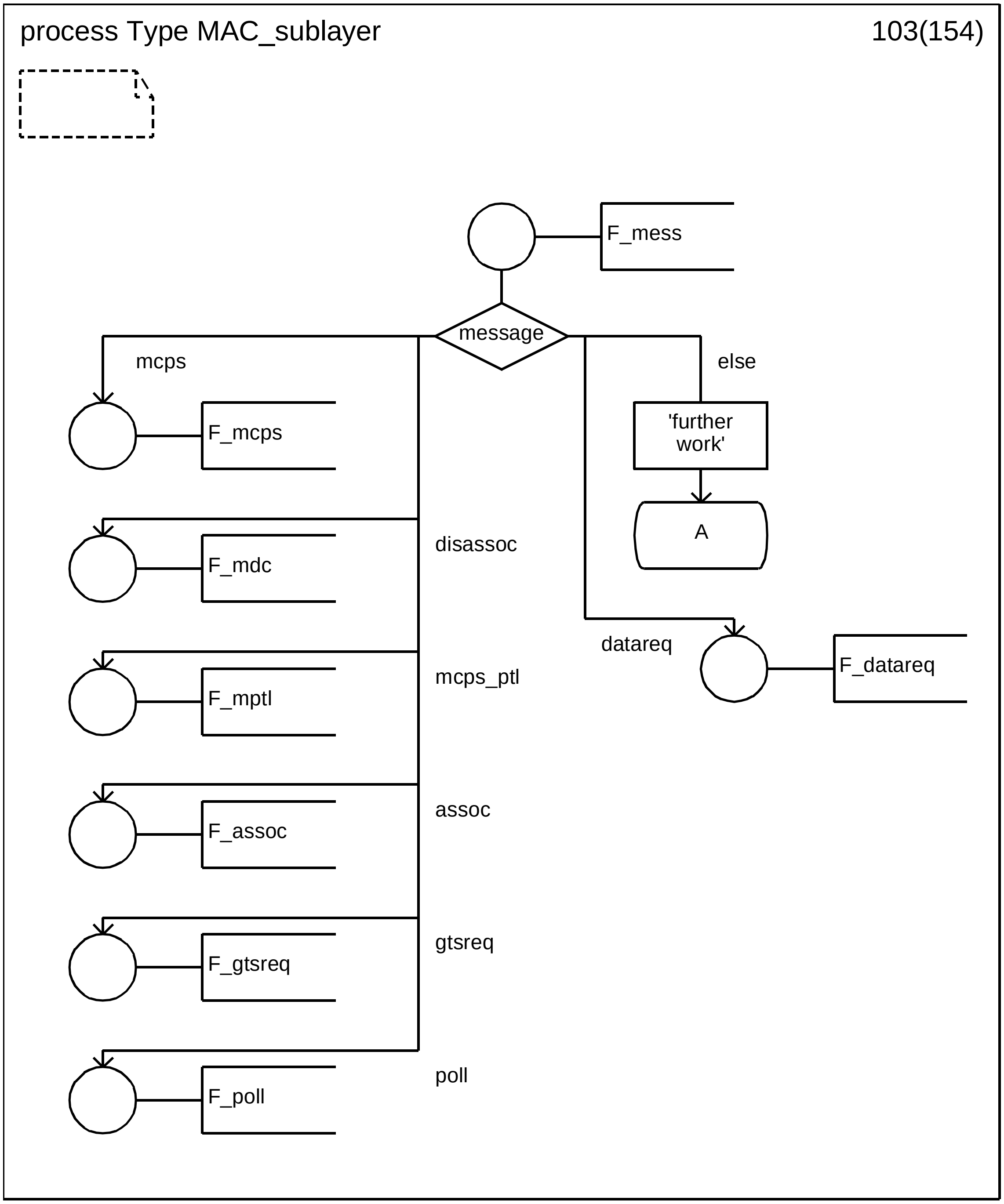
D.3.1.102 Process type MAC_sublayer (102)

.



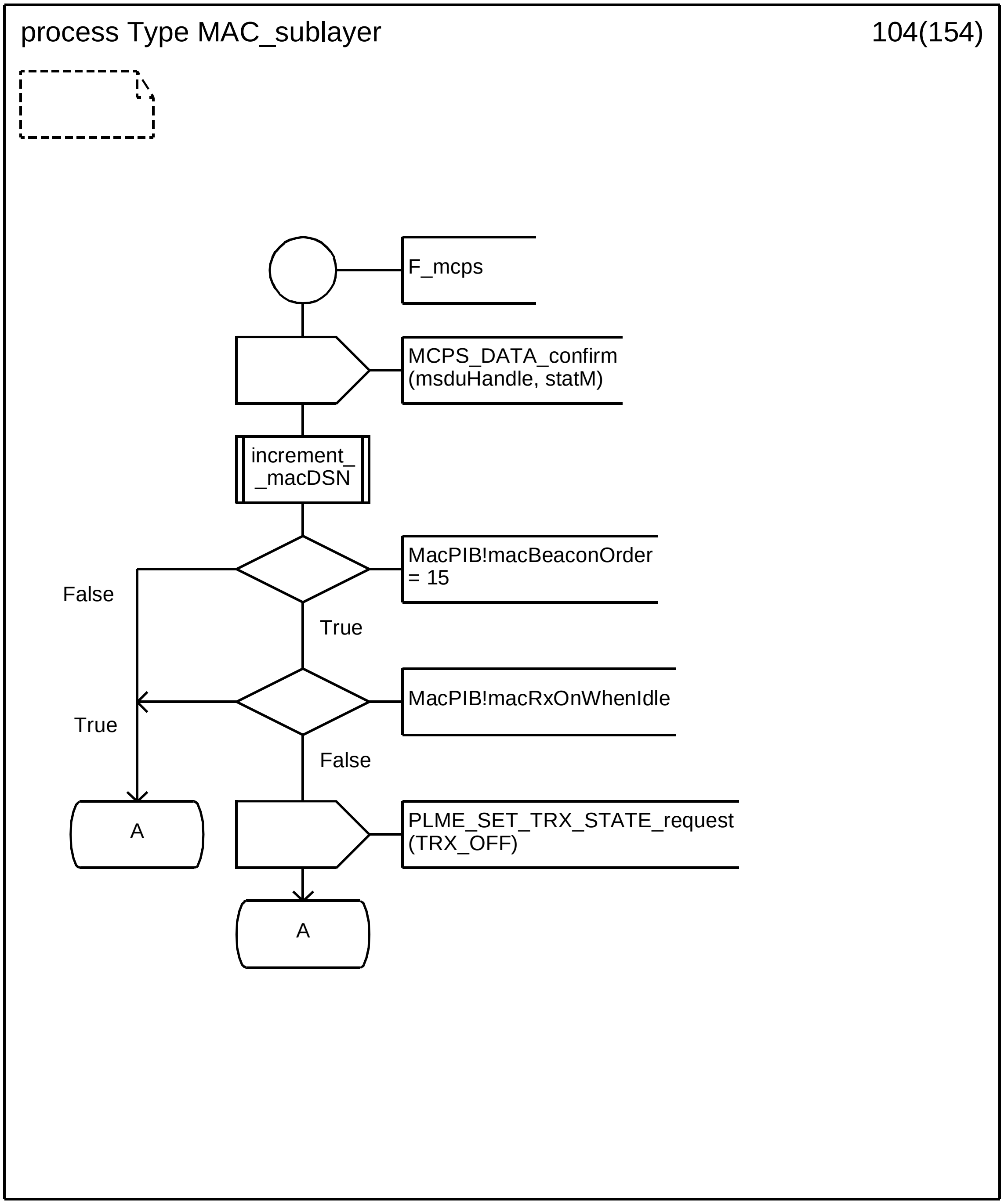
D.3.1.103 Process type MAC_sublayer (103)

.



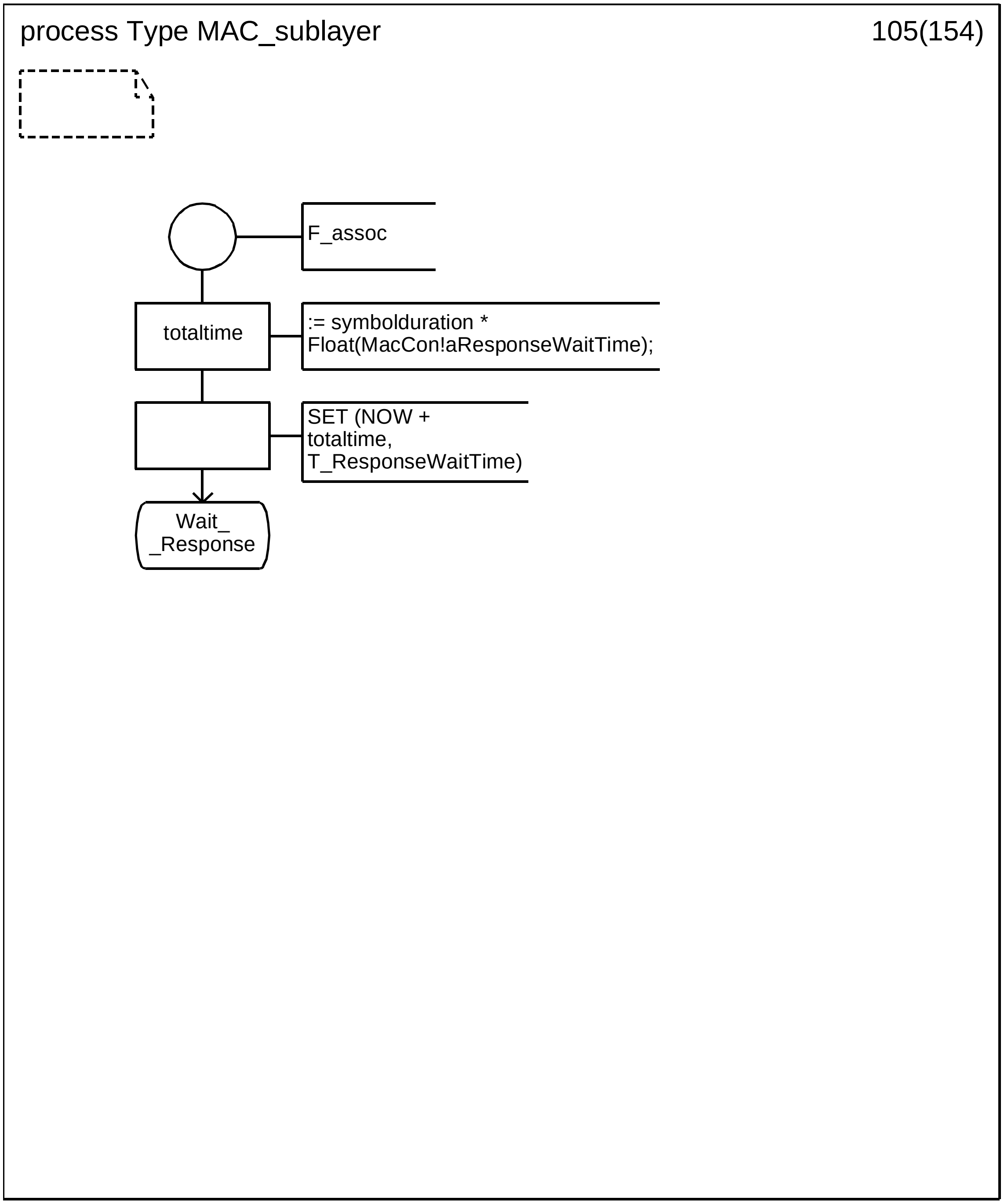
D.3.1.104 Process type MAC_sublayer (104)

.



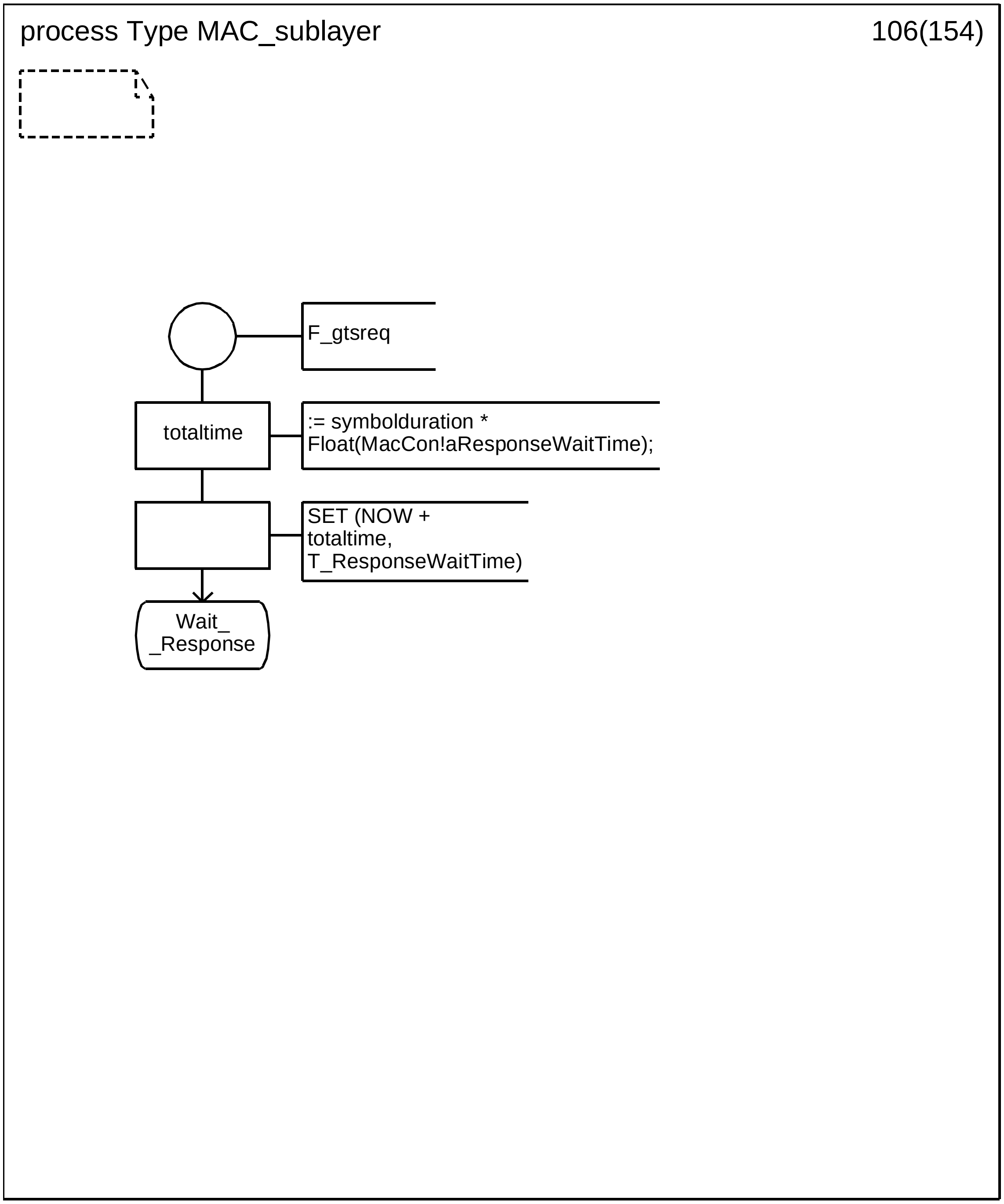
D.3.1.105 Process type MAC_sublayer (105)

.



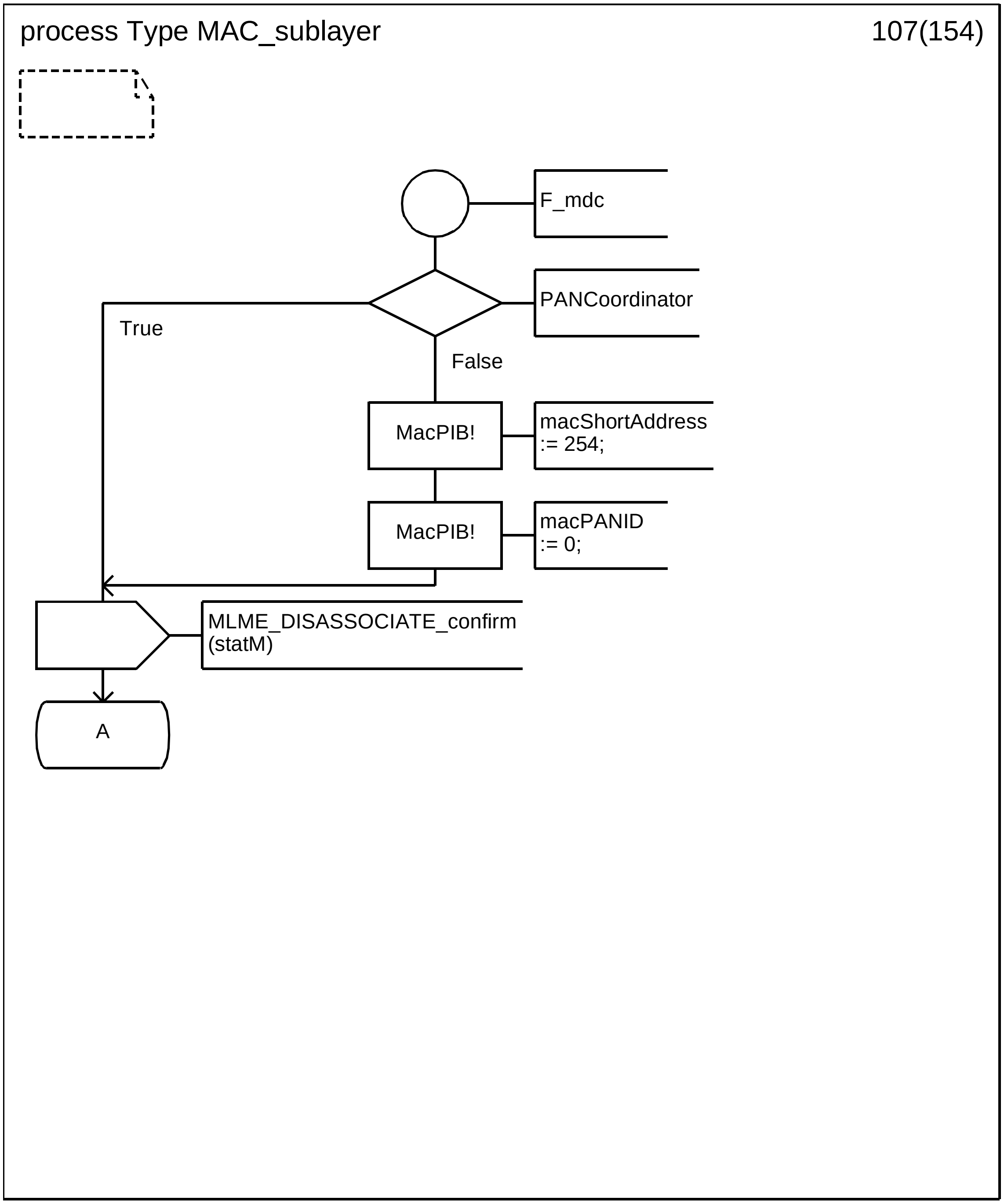
D.3.1.106 Process type MAC_sublayer (106)

.



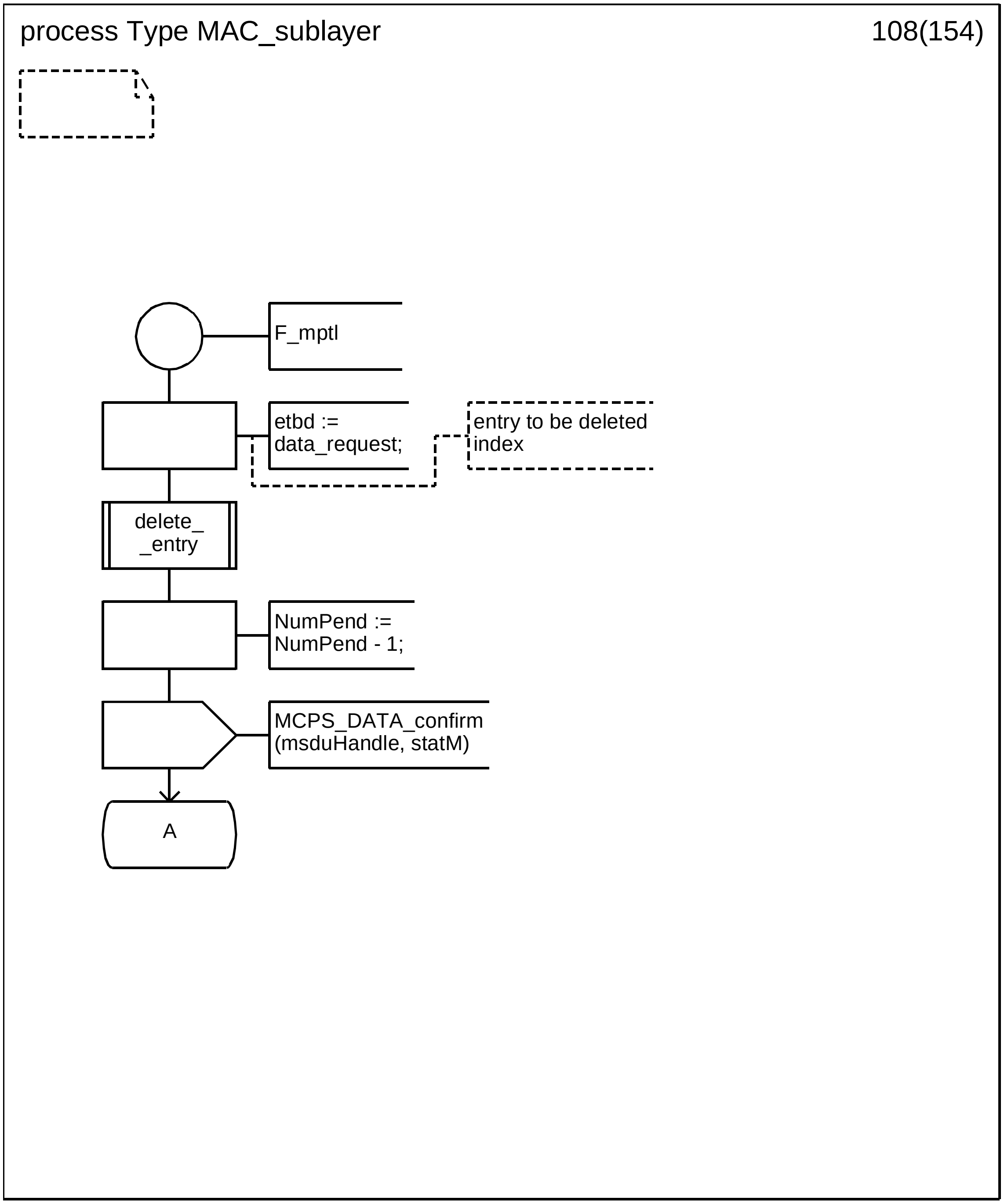
D.3.1.107 Process type MAC_sublayer (107)

.



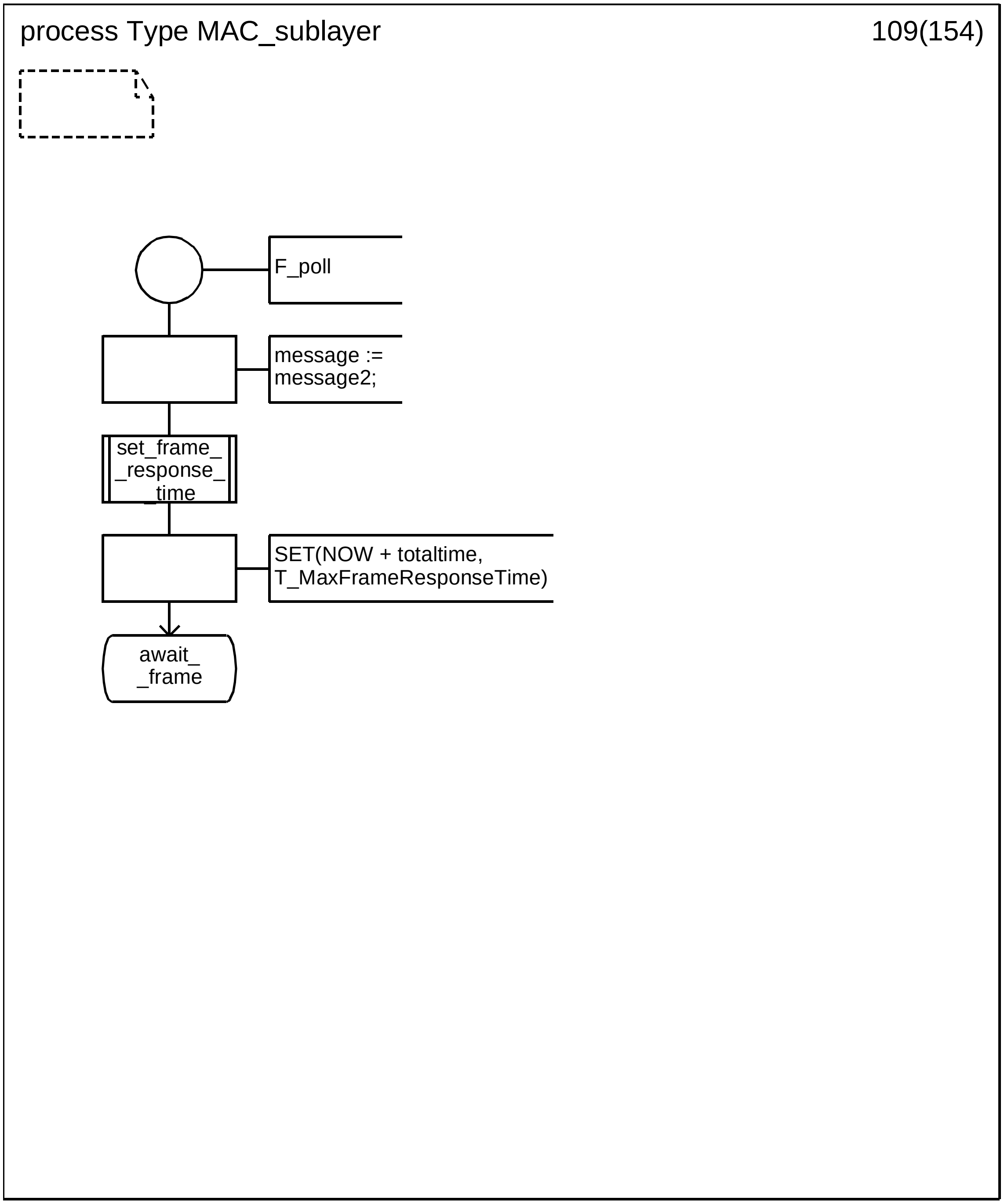
D.3.1.108 Process type MAC_sublayer (108)

.



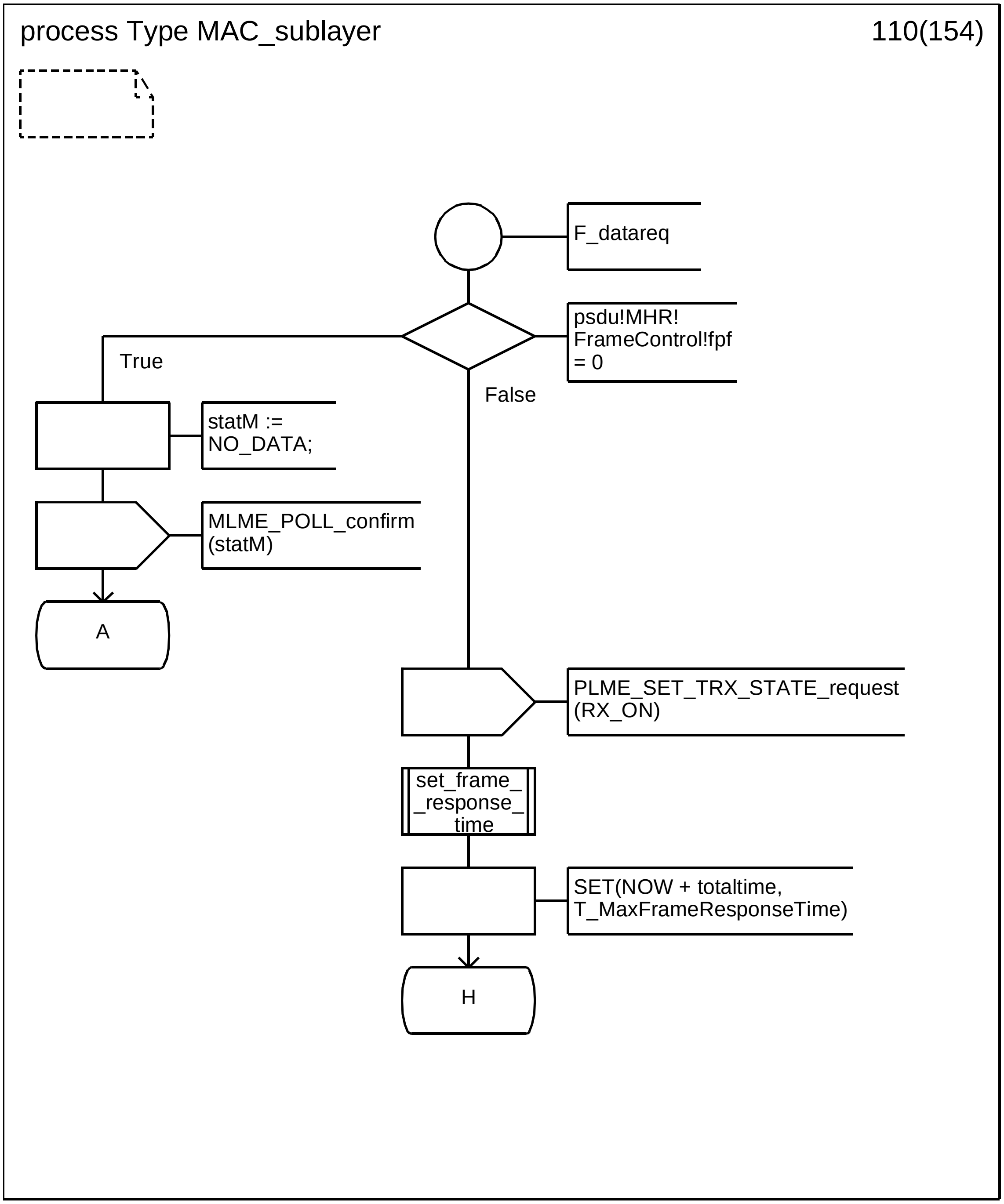
D.3.1.109 Process type MAC_sublayer (109)

.

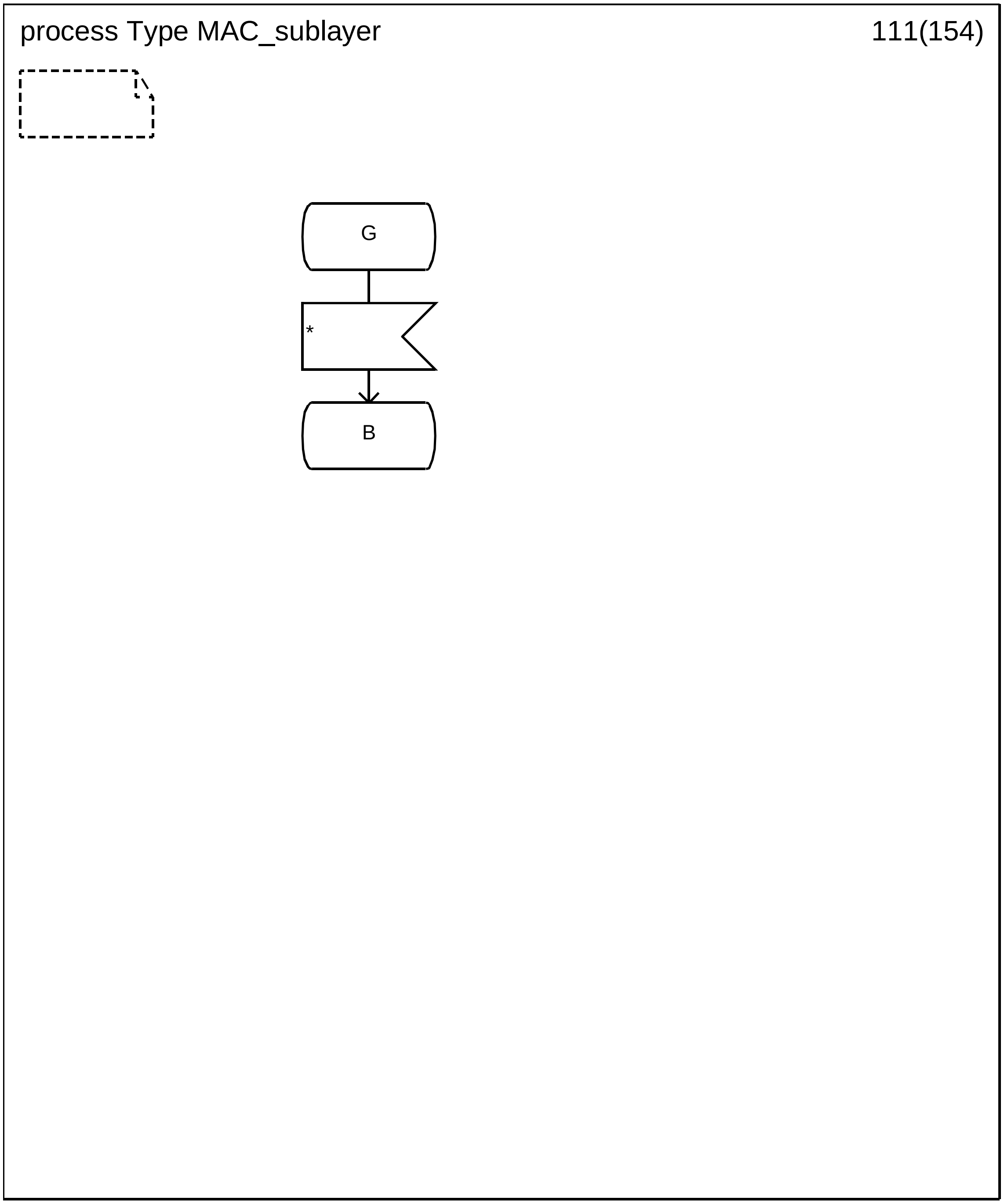


D.3.1.110 Process type MAC_sublayer (110)

.

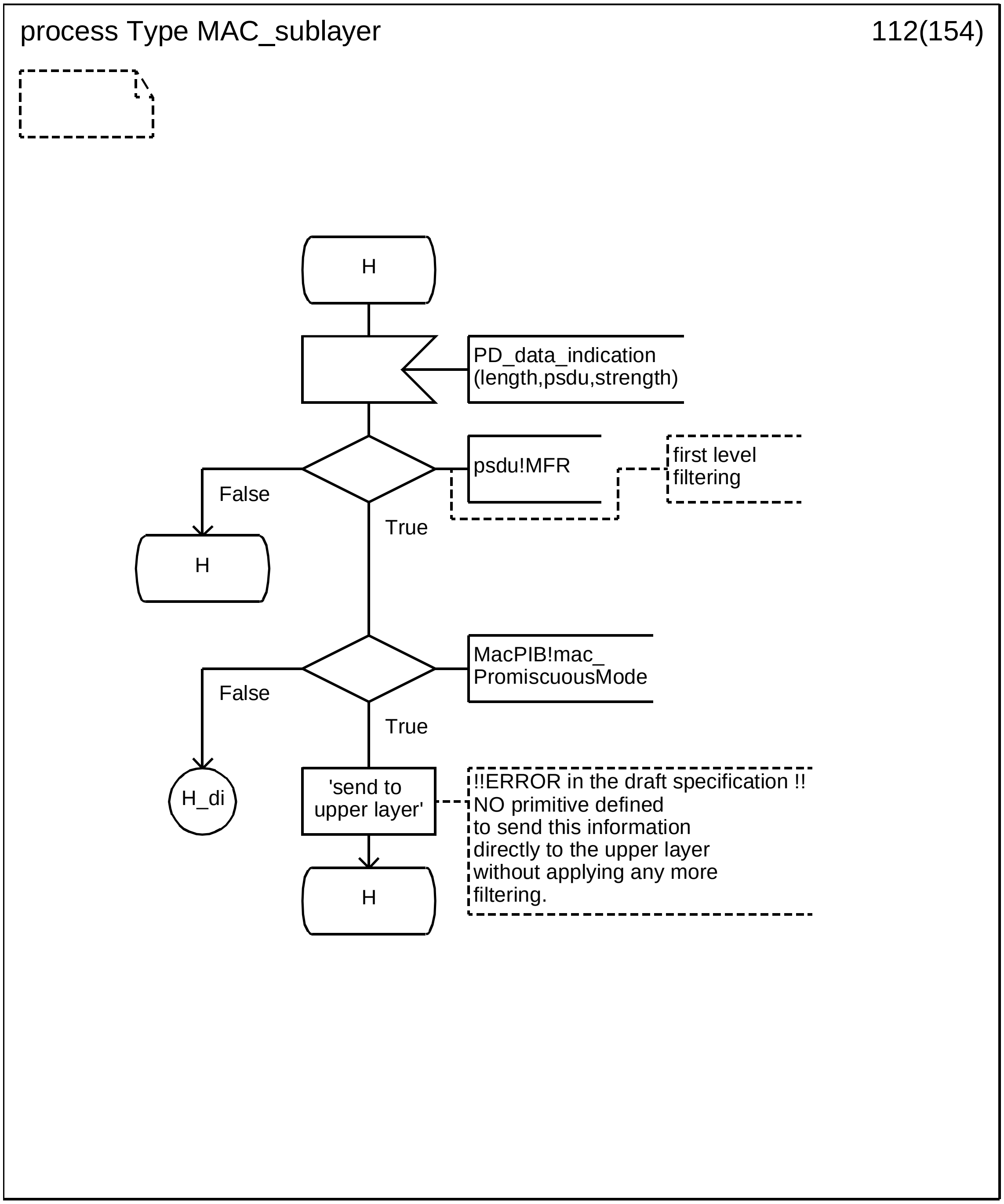


D.3.1.111 Process type MAC_sublayer (111)



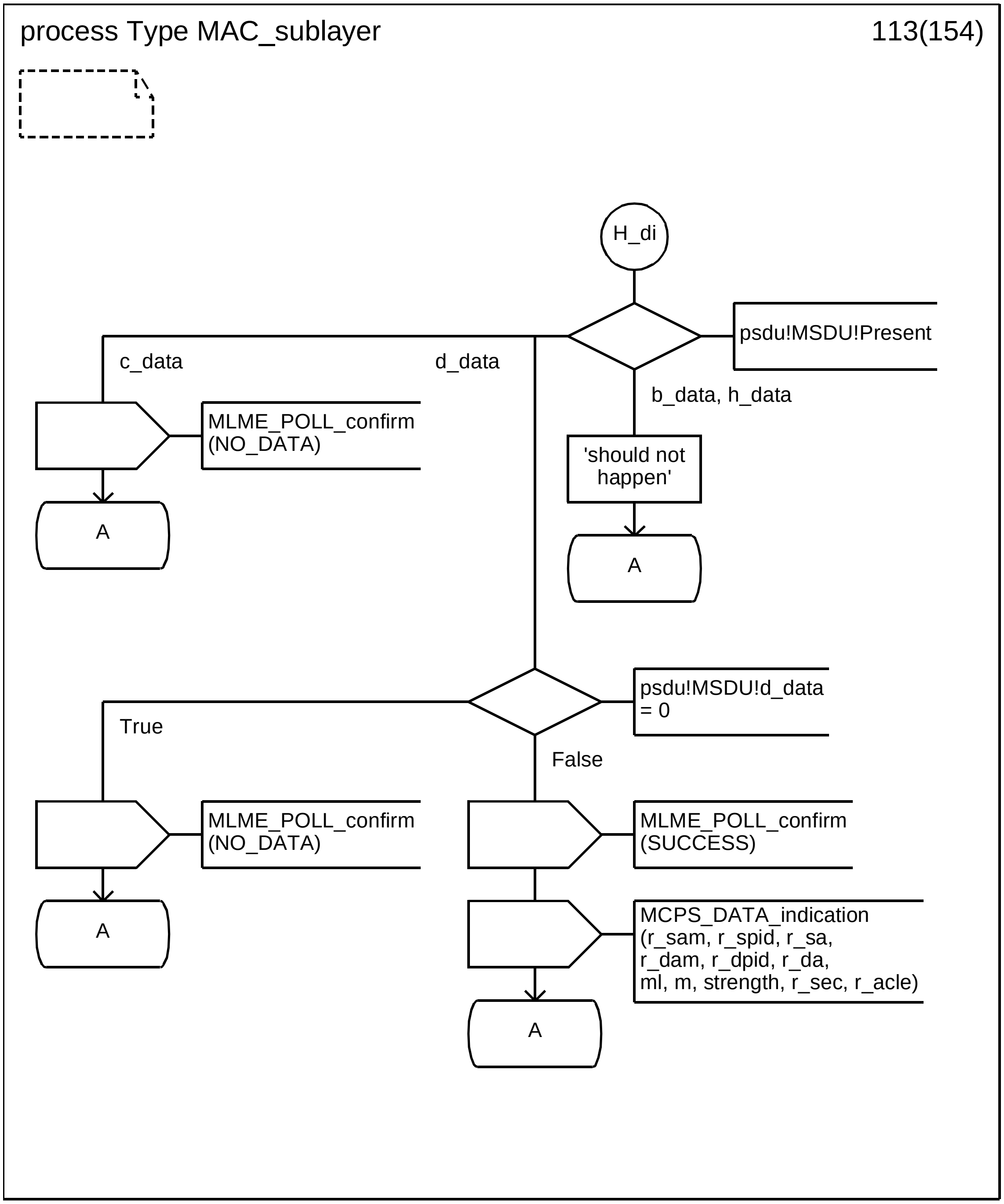
D.3.1.112 Process type MAC_sublayer (112)

.



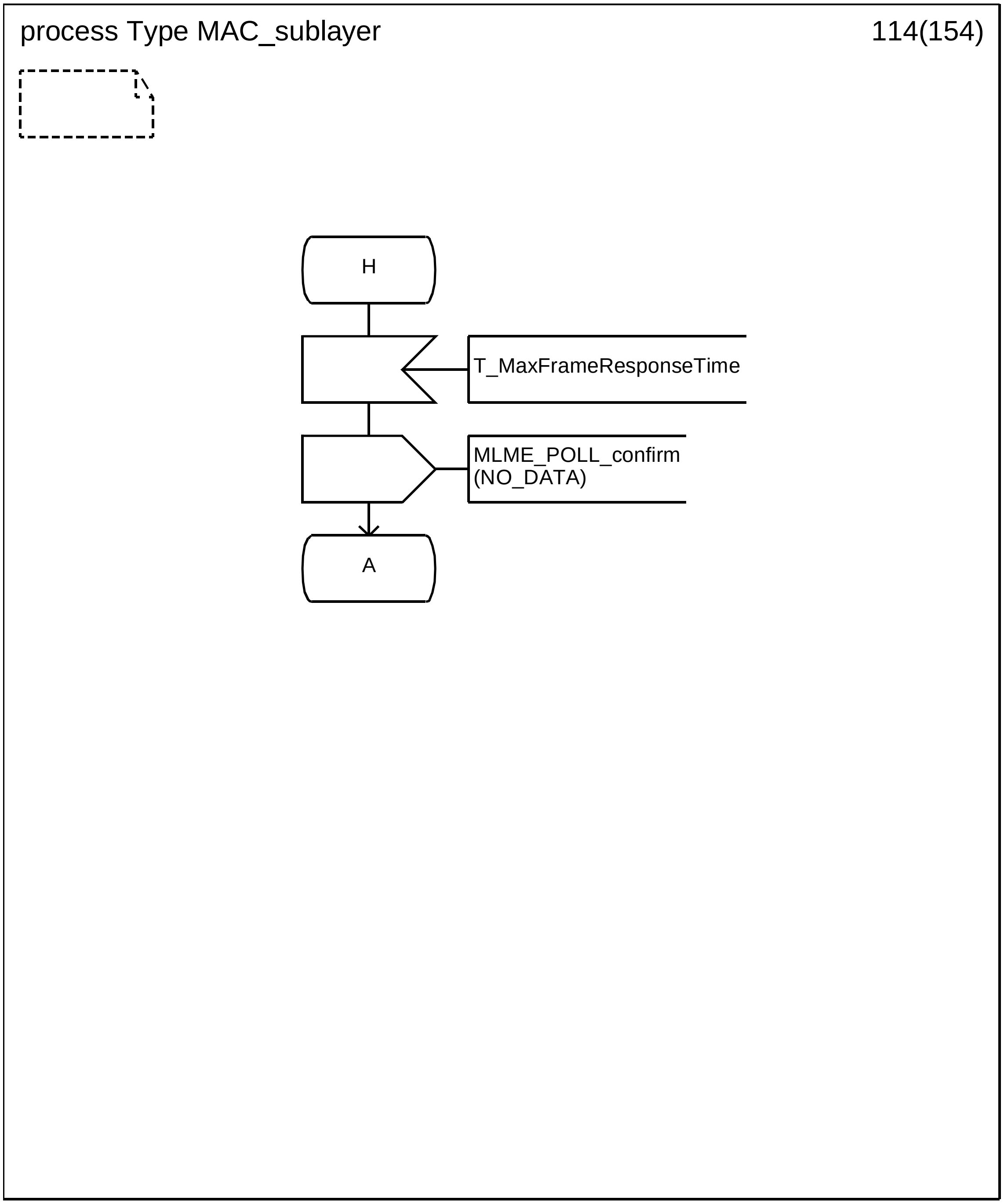
D.3.1.113 Process type MAC_sublayer (113)

.


process Type MAC_sublayer
113(154)
H_di
psdu!MSDU!Present
c_data
d_data
b_data, h_data
MLME_POLL_confirm (NO_DATA)
'should not happen'
A
A
psdu!MSDU!d_data = 0
True
False
MLME_POLL_confirm (NO_DATA)
MLME_POLL_confirm (SUCCESS)
A
MCPS_DATA_indication (r_sam, r_spid, r_sa, r_dam, r_dpid, r_da, ml, m, strength, r_sec, r_acle)
A

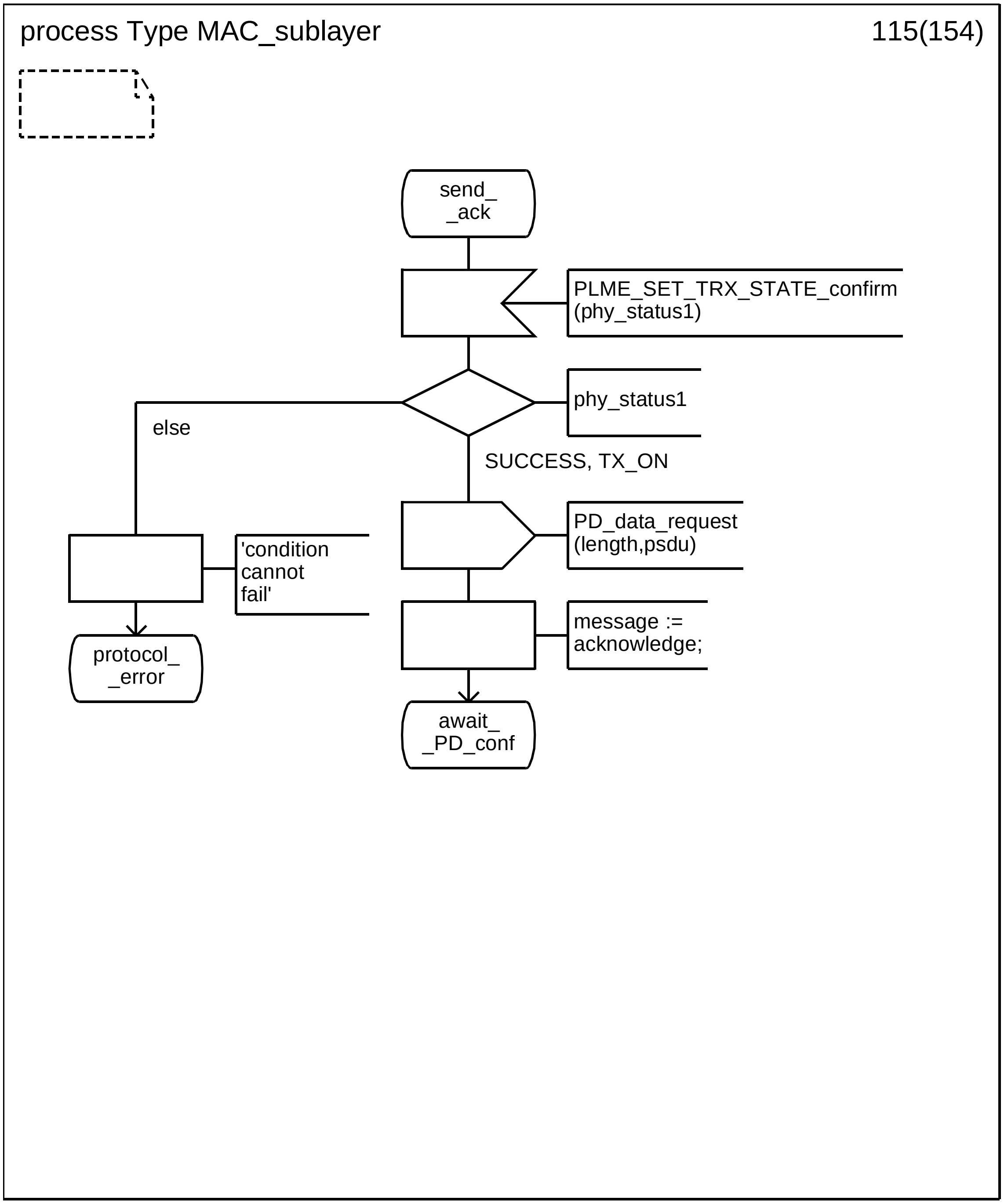
D.3.1.114 Process type MAC_sublayer (114)

.



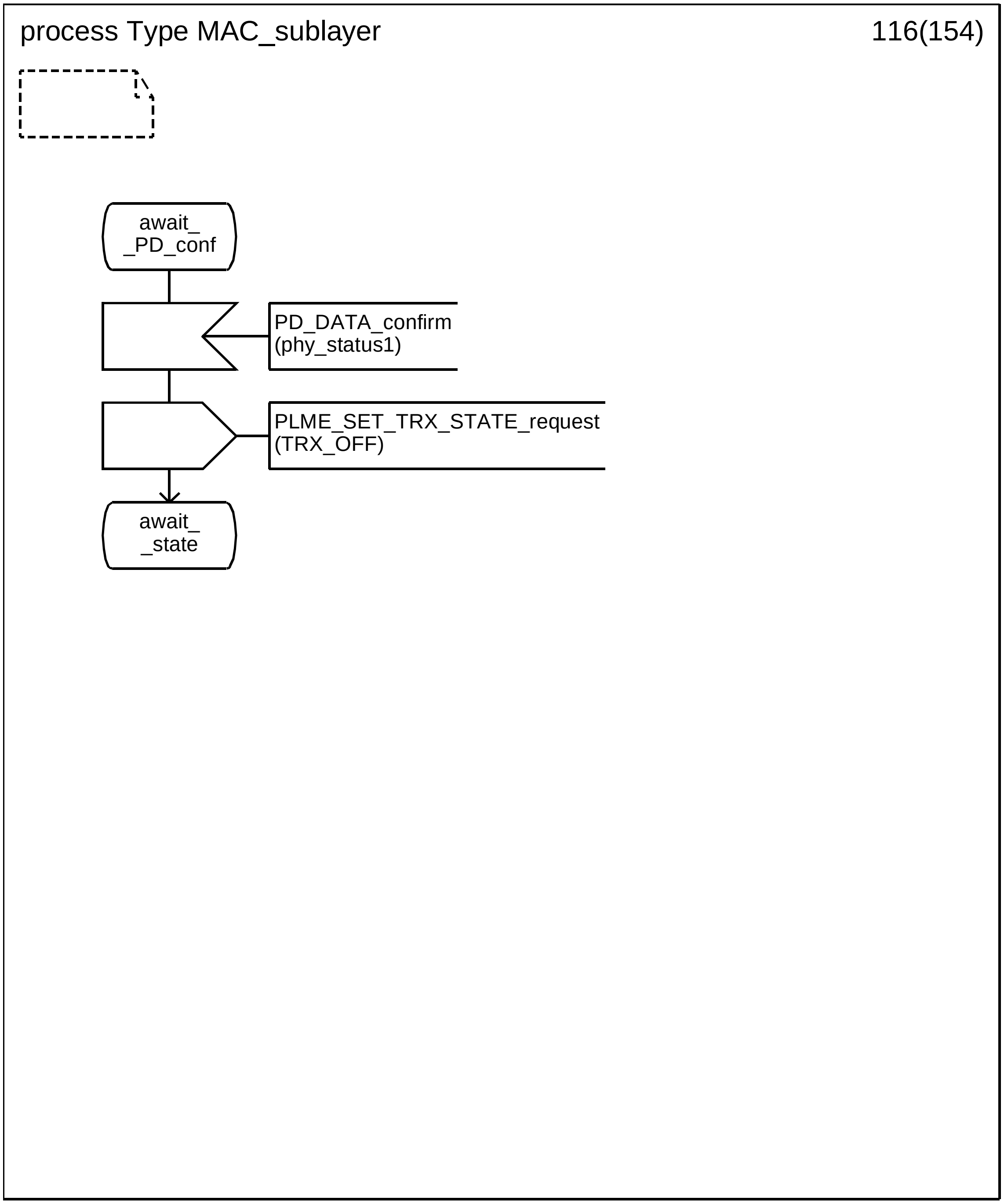
D.3.1.115 Process type MAC_sublayer (115)

.

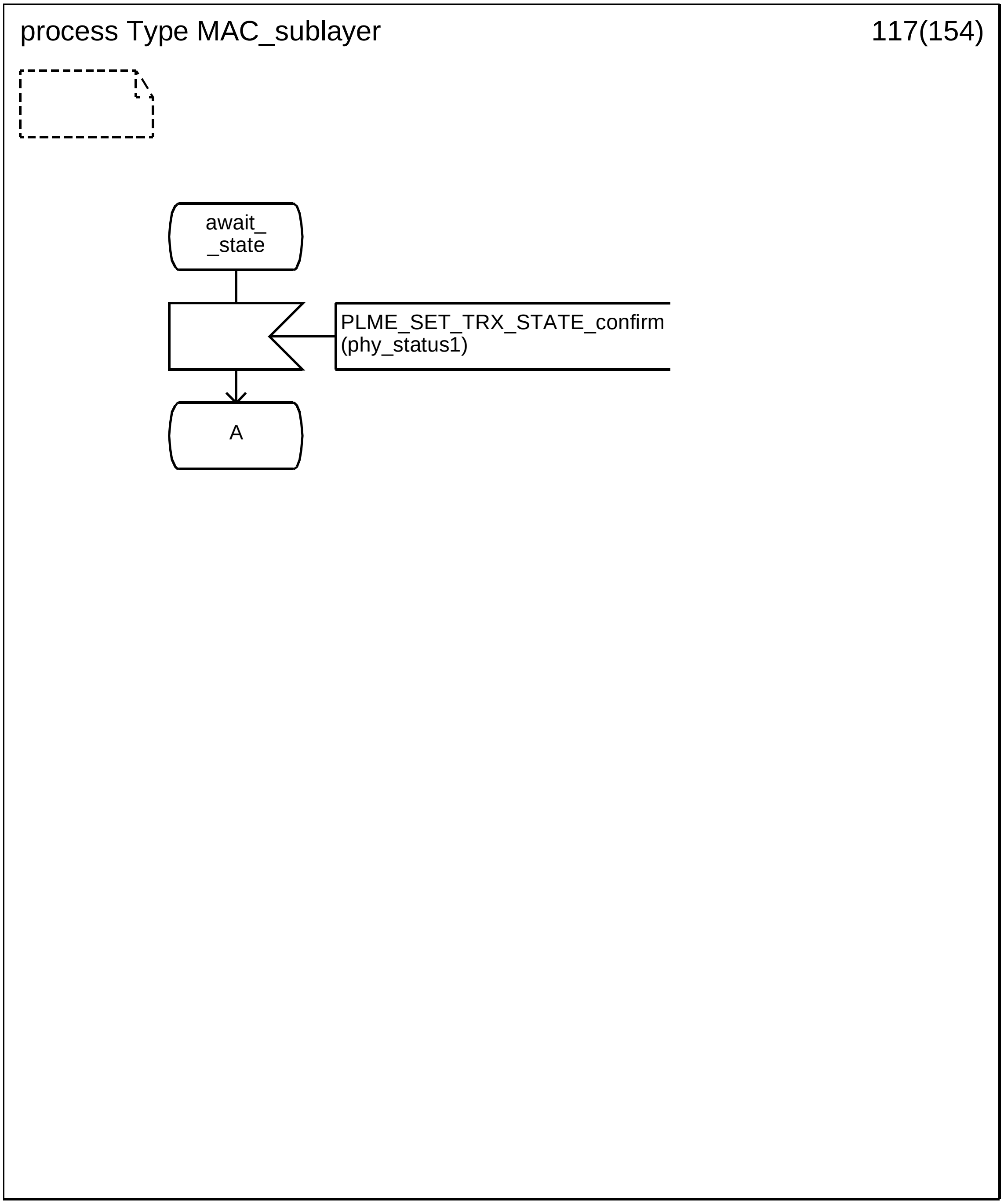


D.3.1.116 Process type MAC_sublayer (116)

.

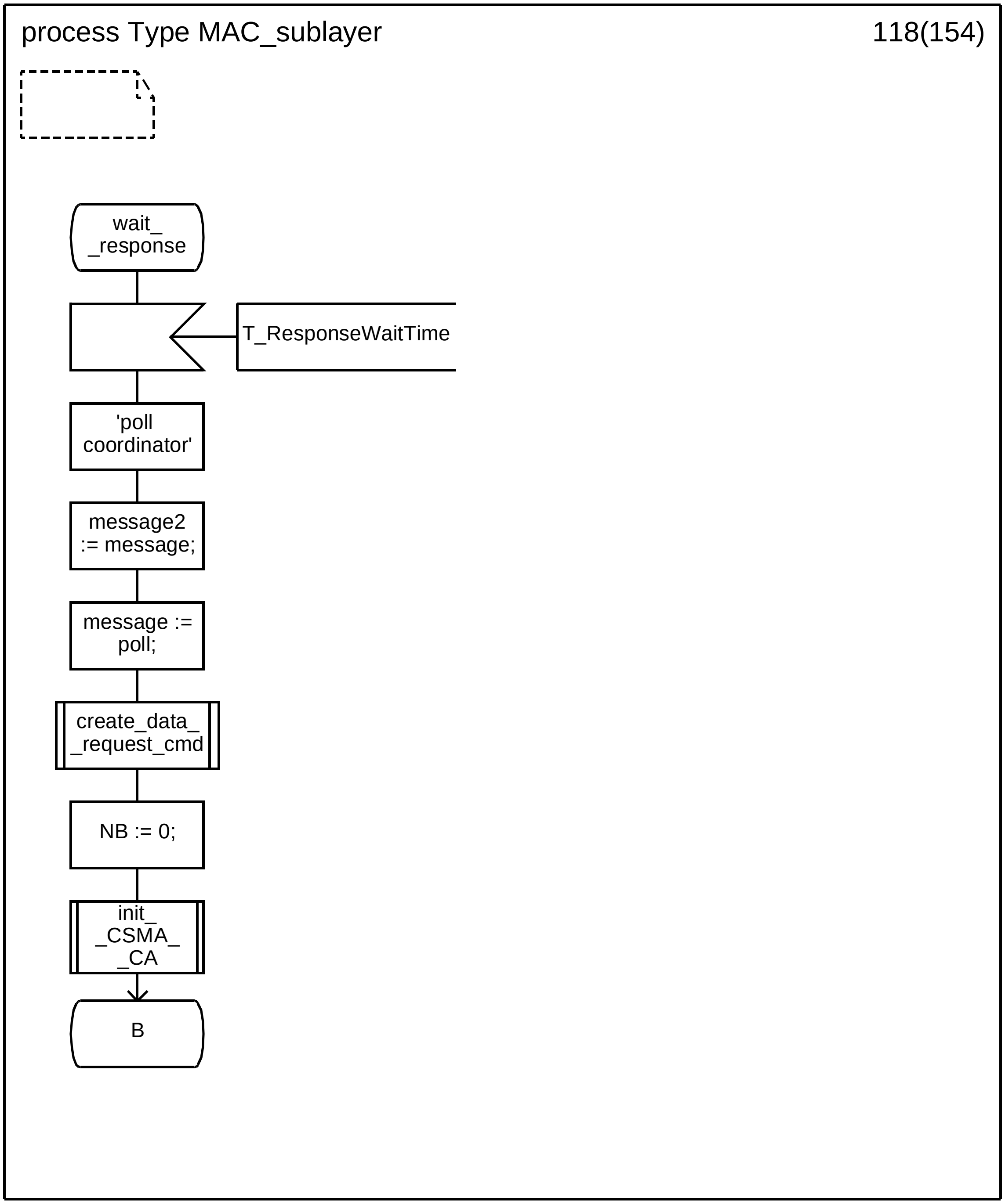


D.3.1.117 Process type MAC_sublayer (117)



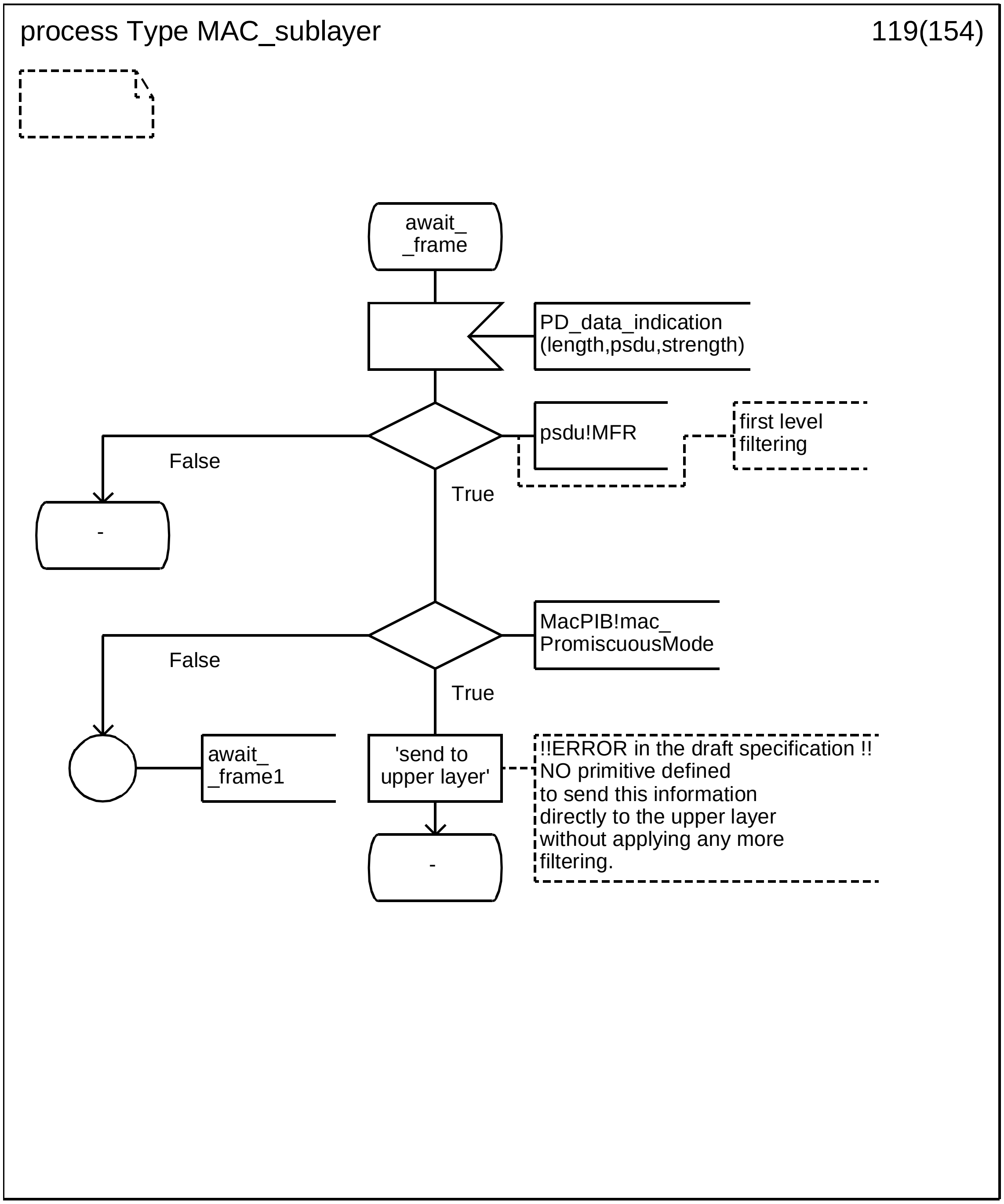
D.3.1.118 Process type MAC_sublayer (118)

.



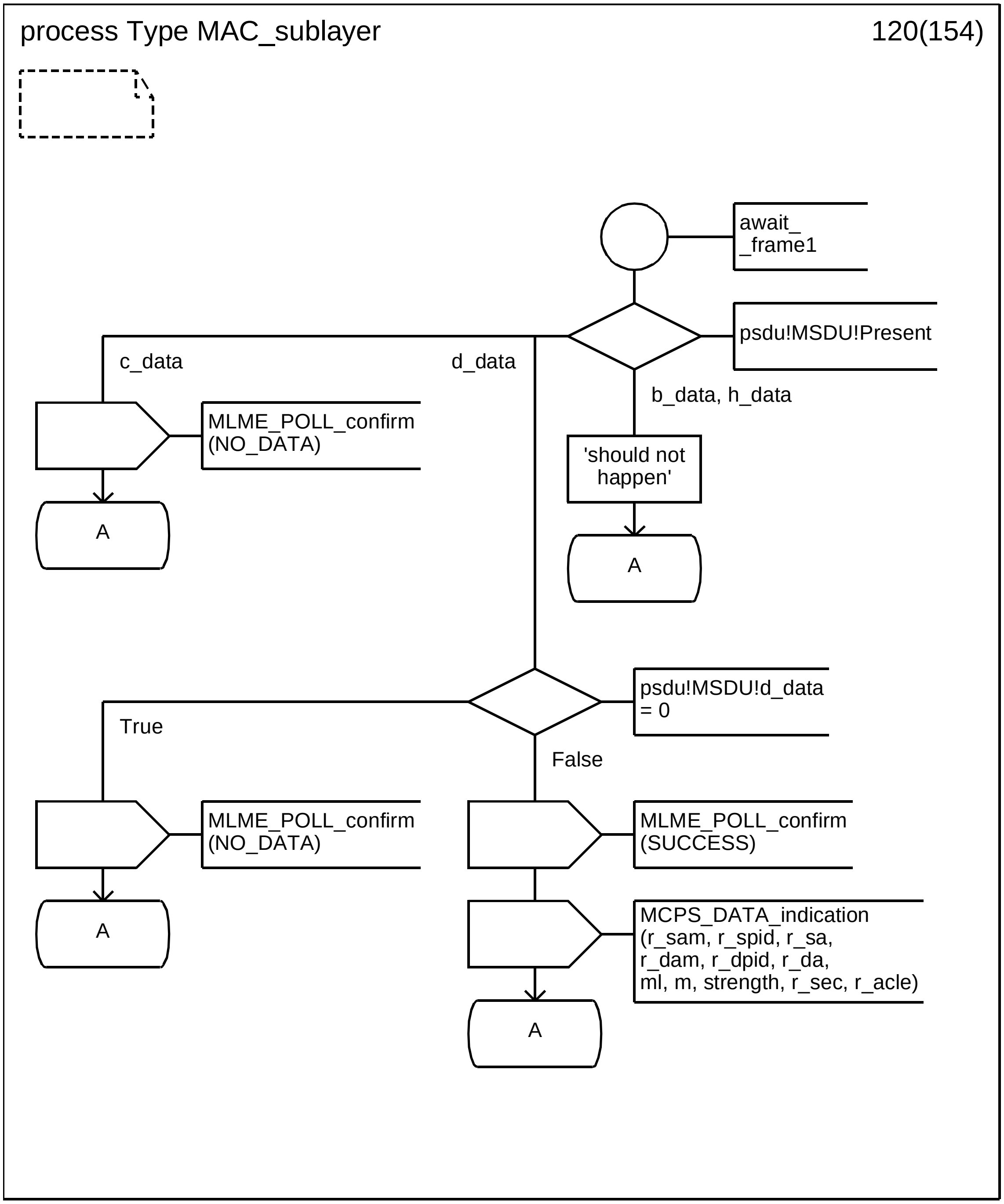
D.3.1.119 Process type MAC_sublayer (119)

.


process Type MAC_sublayer
119(154)
await_ _frame
PD_data_indication (length,psdu,strength)
psdu!MFR
first level filtering
False
True
-
MacPIB!mac_ PromiscuousMode
False
True
await_ _frame1
'send to upper layer'
!!ERROR in the draft specification !! NO primitive defined to send this information directly to the upper layer without applying any more filtering.
-

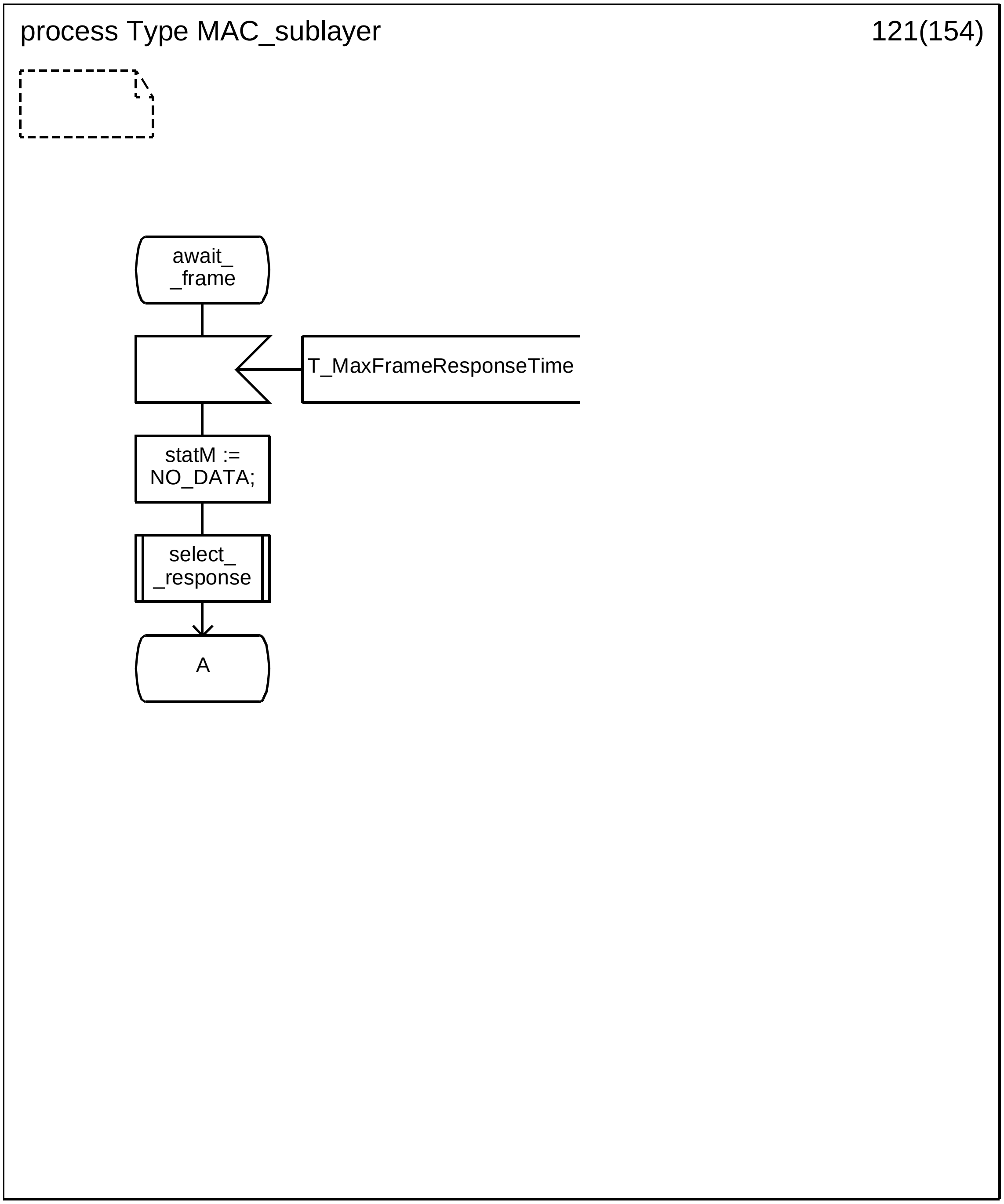
D.3.1.120 Process type MAC_sublayer (120)

.



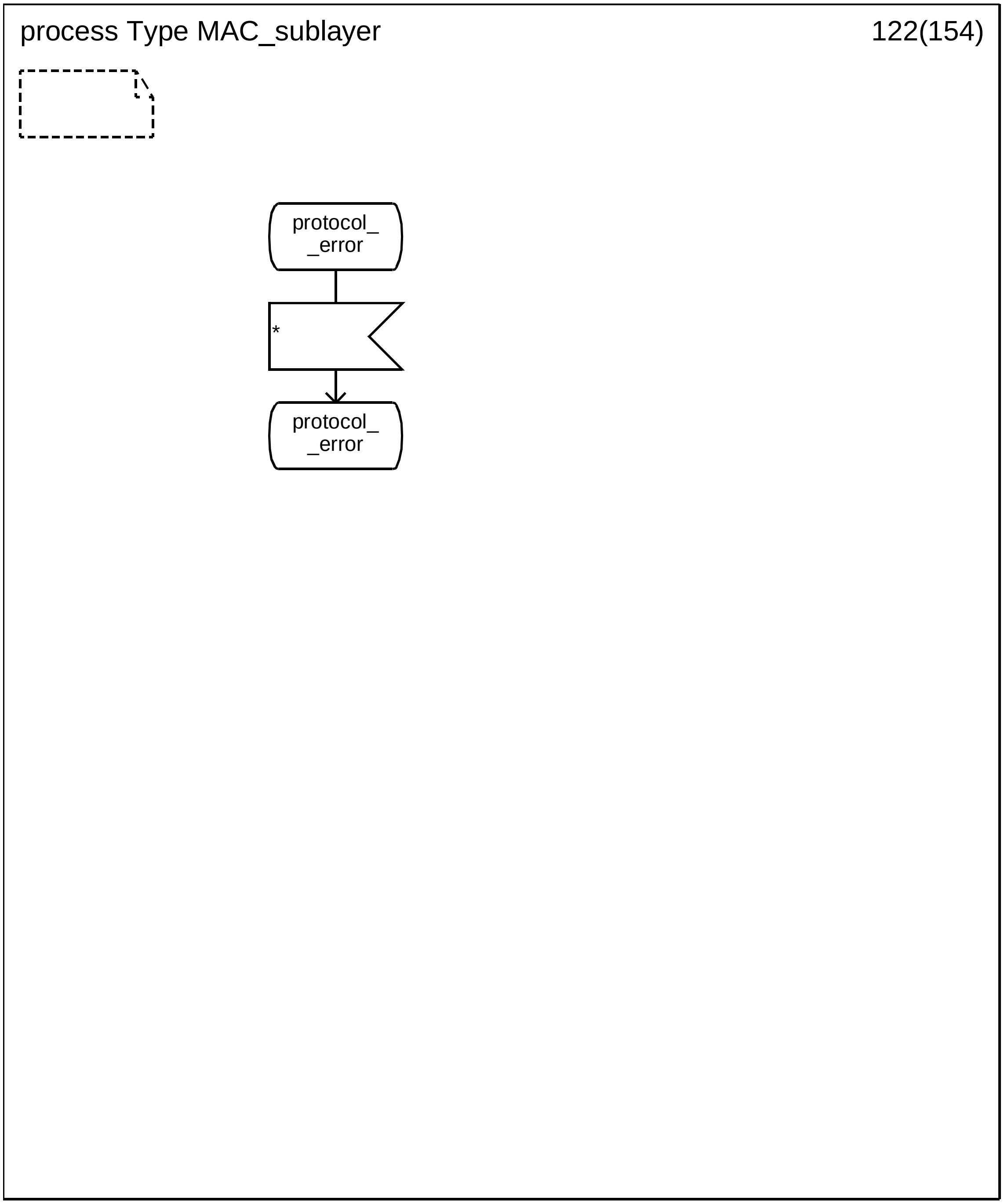
D.3.1.121 Process type MAC_sublayer (121)

.



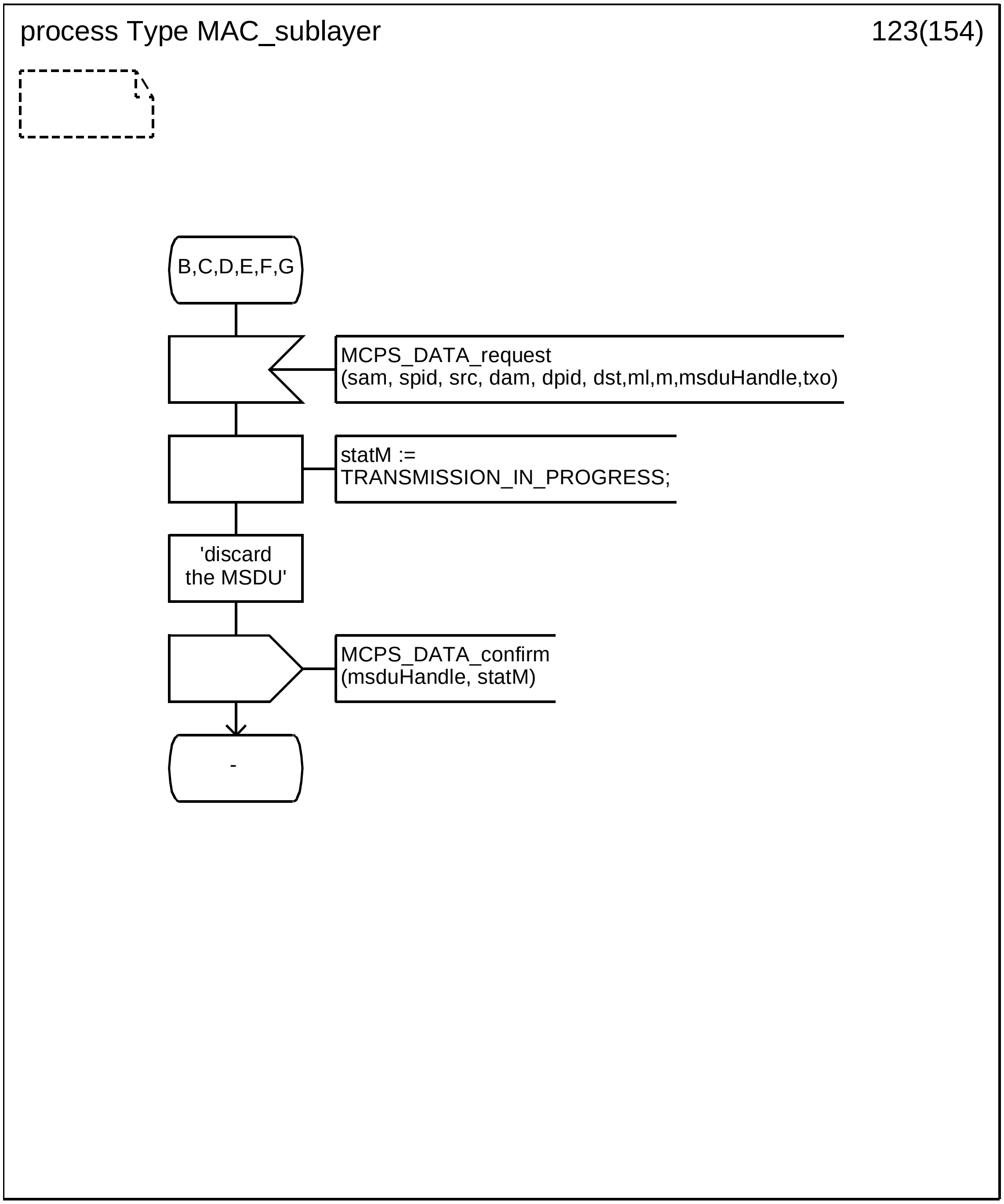
D.3.1.122 Process type MAC_sublayer (122)

.



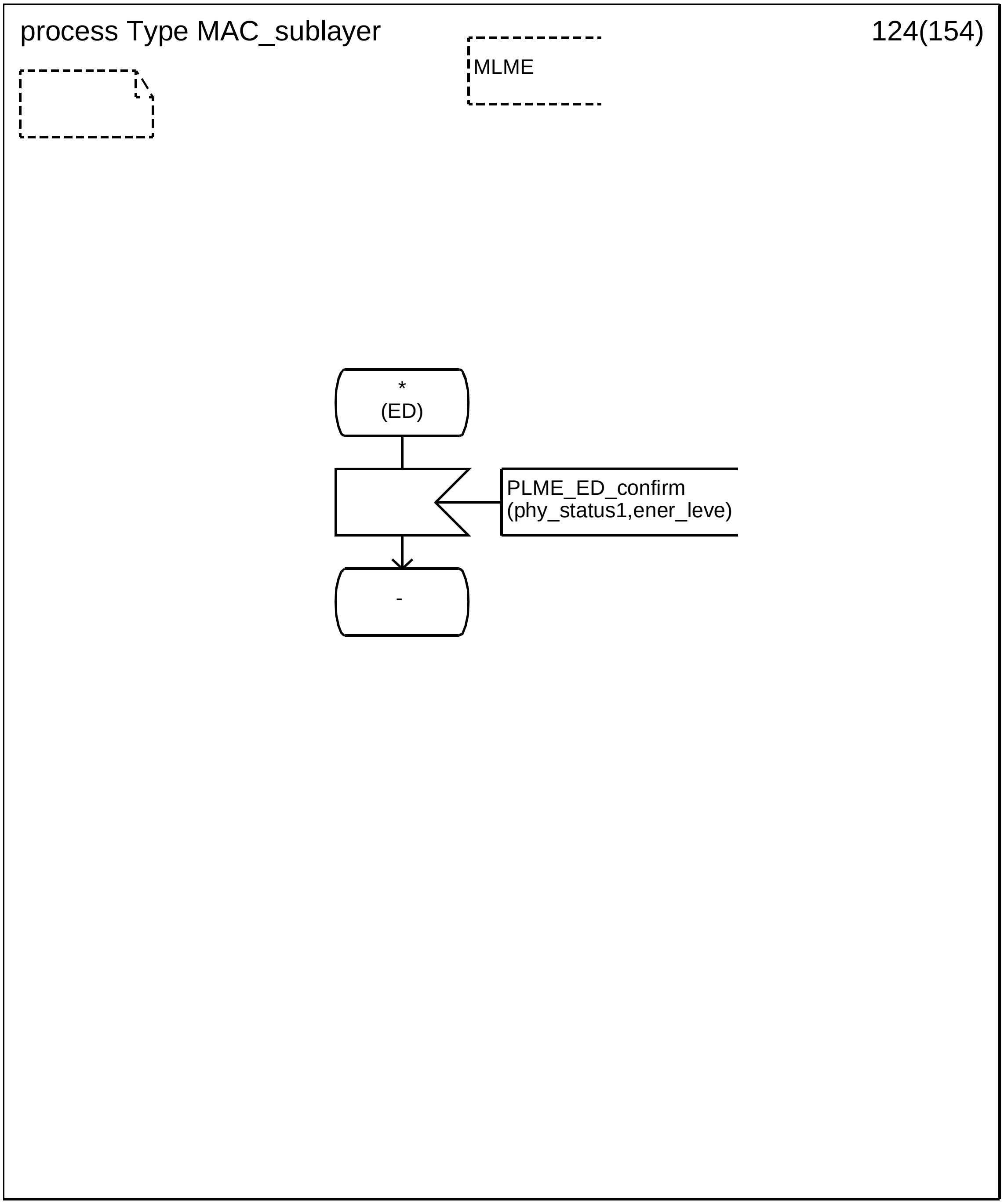
D.3.1.123 Process type MAC_sublayer (123)

.


process Type MAC_sublayer
123(154)
B,C,D,E,F,G
MCPS_DATA_request
(sam, spid, src, dam, dpid, dst,ml,m,msduHandle,txo)
statM :=
TRANSMISSION_IN_PROGRESS;
'discard
the MSDU'
MCPS_DATA_confirm
(msduHandle, statM)
-

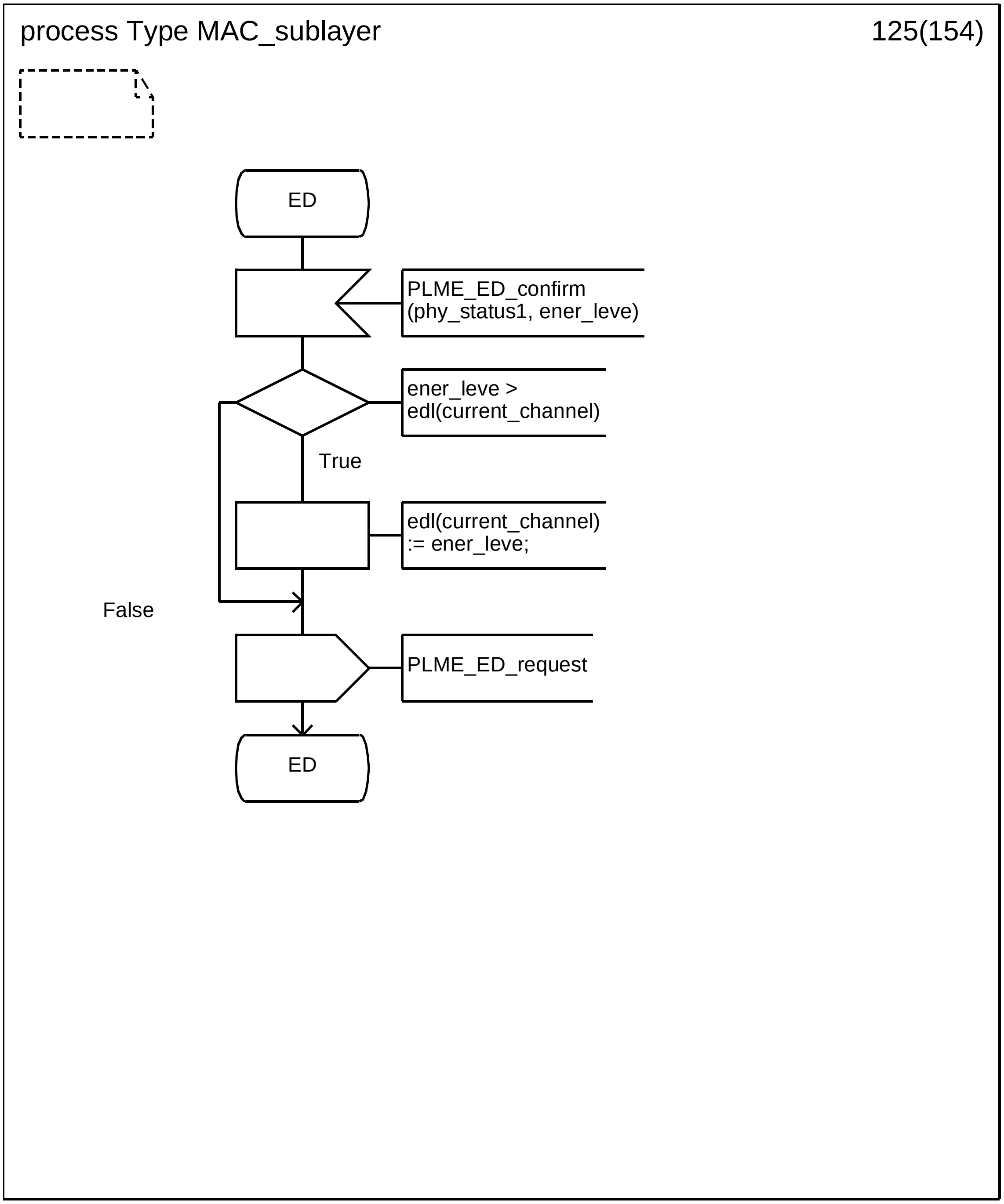
D.3.1.124 Process type MAC_sublayer (124)

.

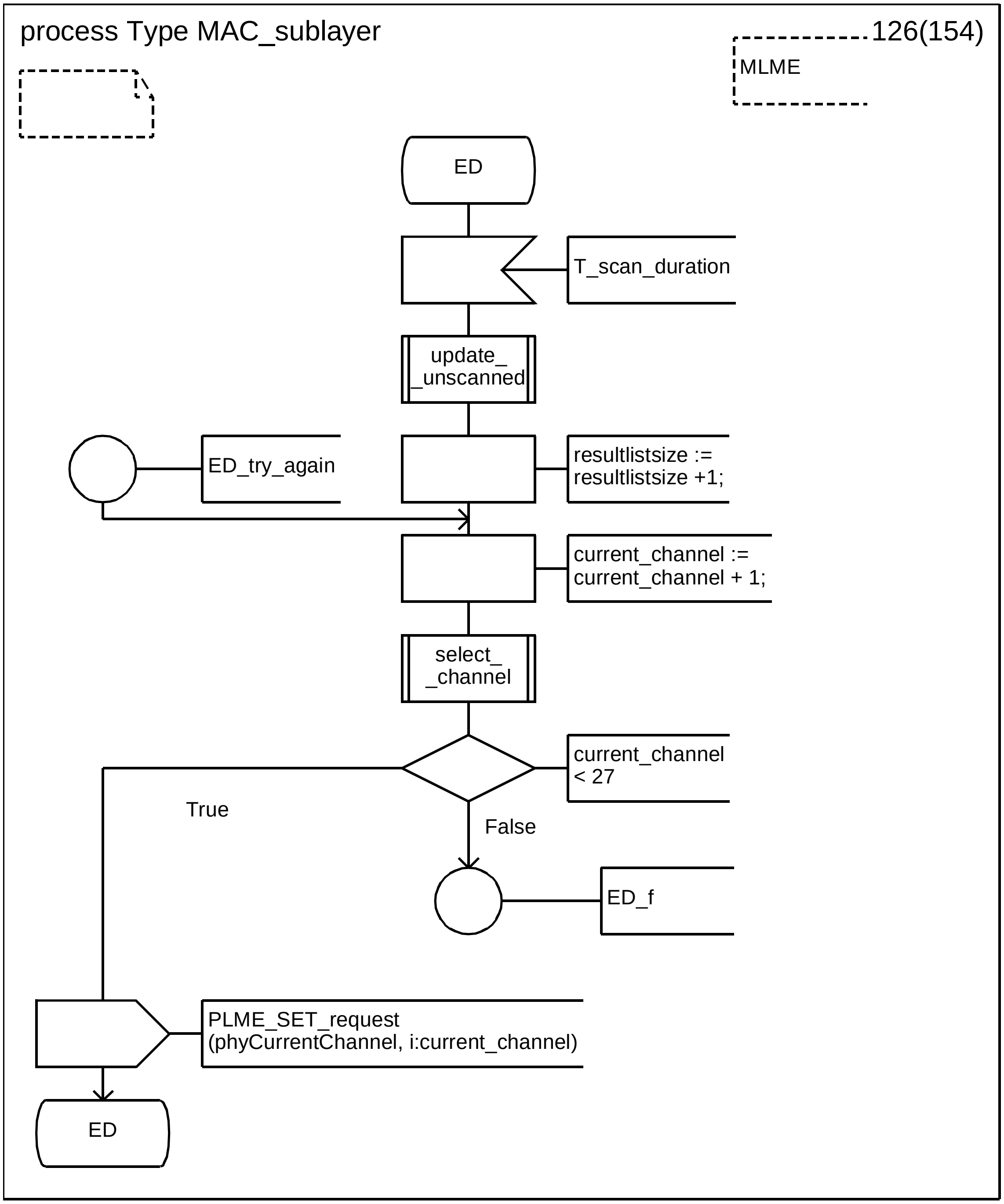


D.3.1.125 Process type MAC_sublayer (125)

.

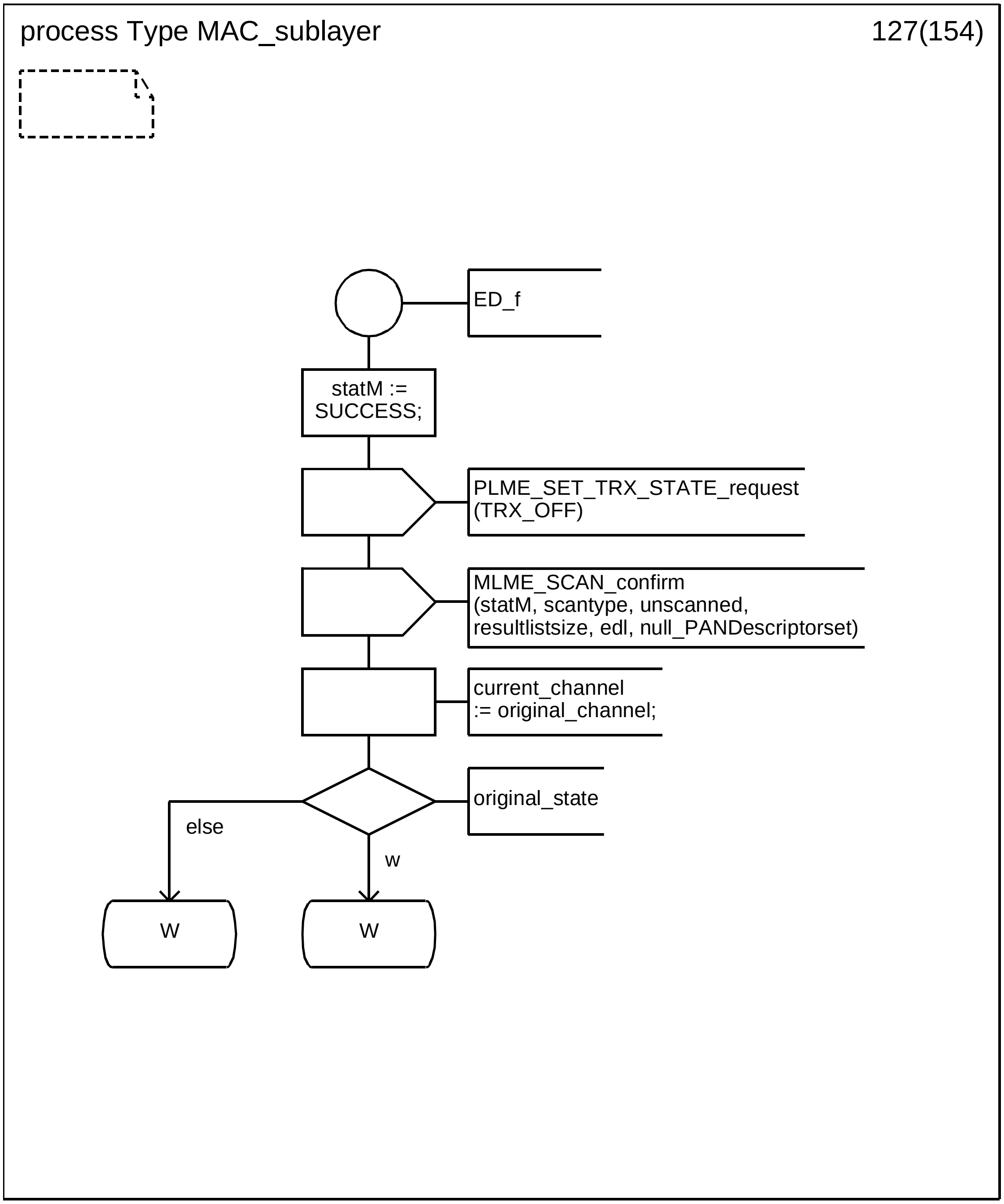


D.3.1.126 Process type MAC_sublayer (126)



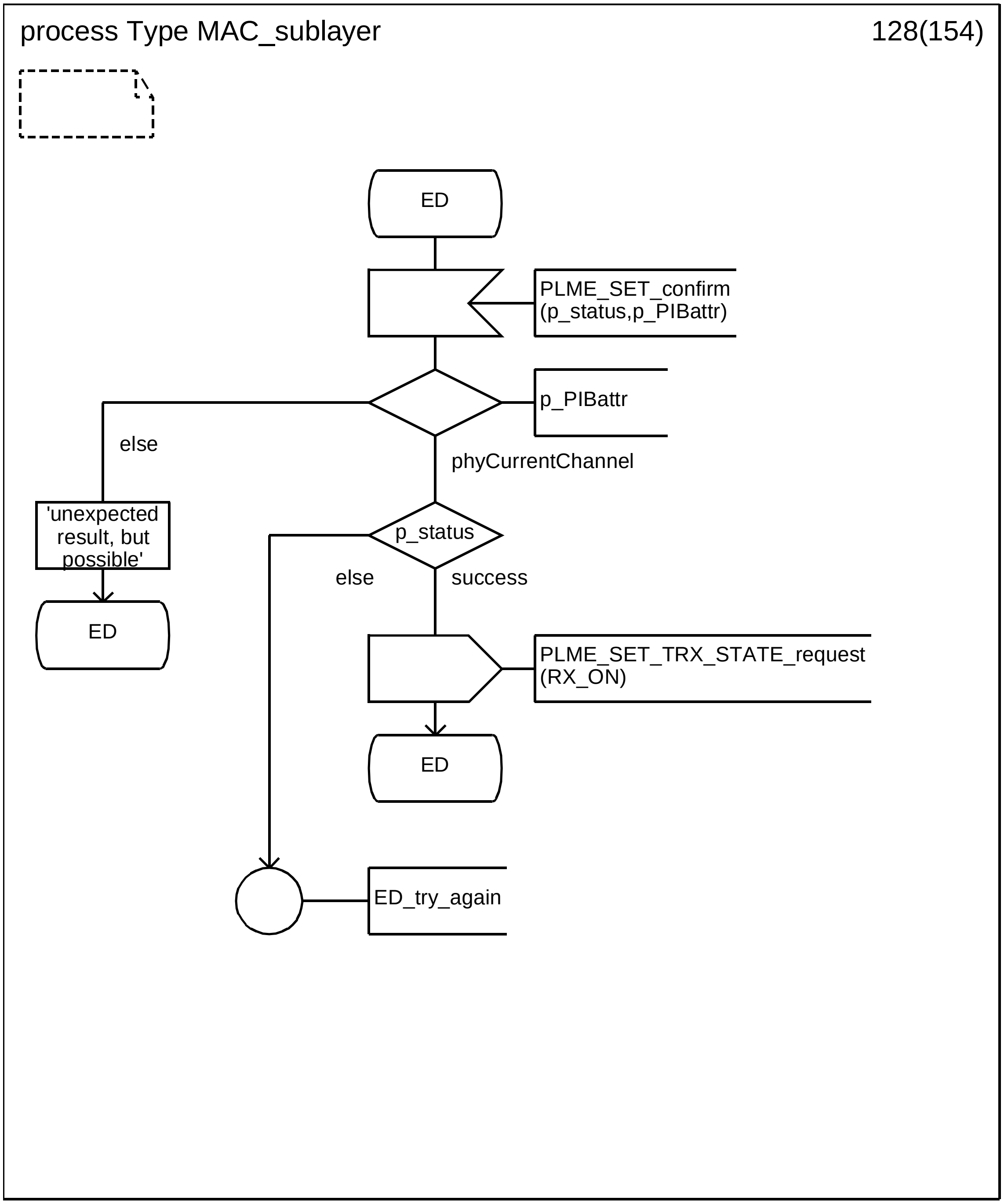
D.3.1.127 Process type MAC_sublayer (127)

.


process Type MAC_sublayer
127(154)
ED_f
statM :=
SUCCESS;
PLME_SET_TRX_STATE_request
(TRX_OFF)
MLME_SCAN_confirm
(statM, scantype, unscanned,
resultlistsize, edl, null_PANDescriptorset)
current_channel
:= original_channel;
original_state
else
w
W
W

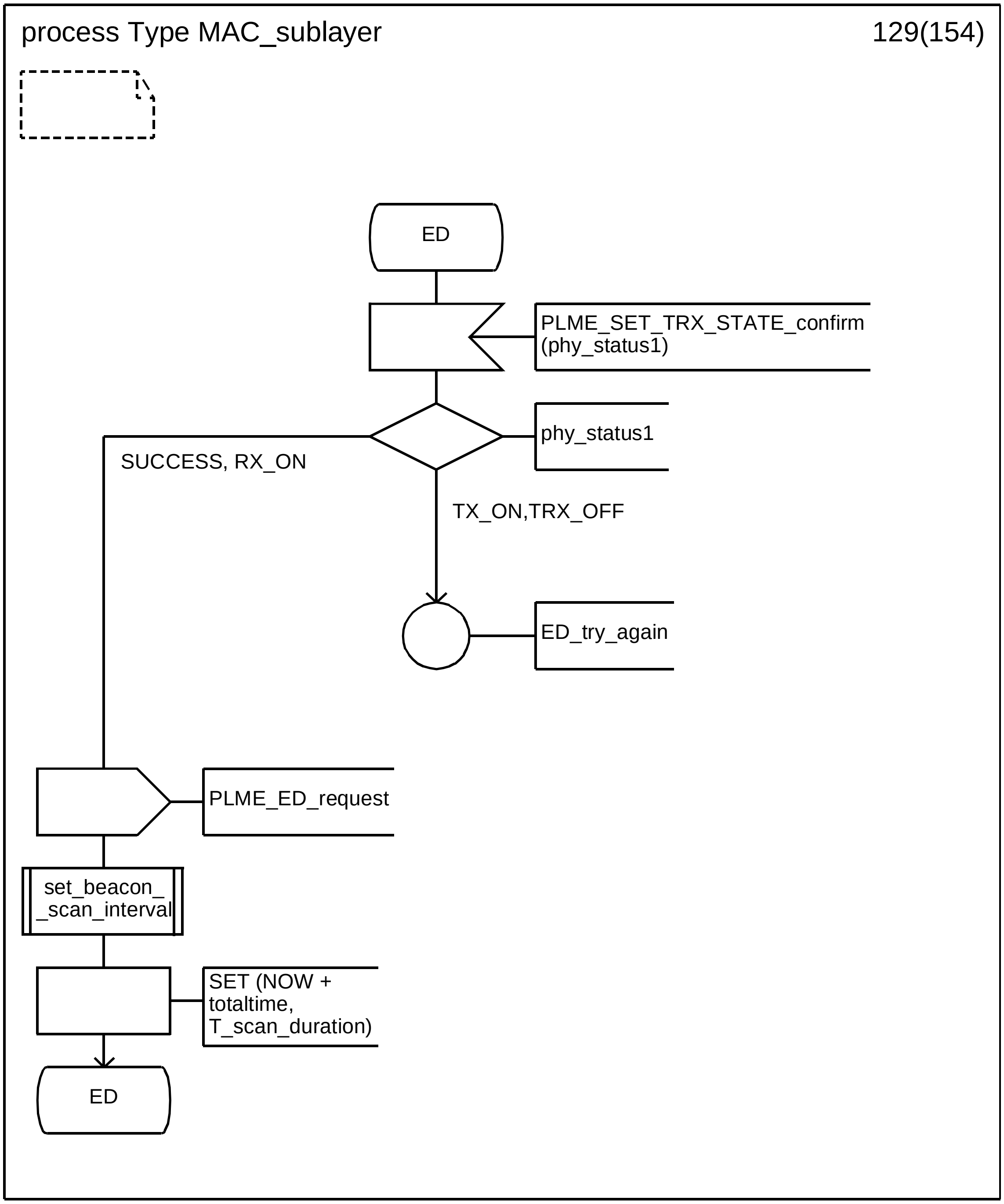
D.3.1.128 Process type MAC_sublayer (128)

.

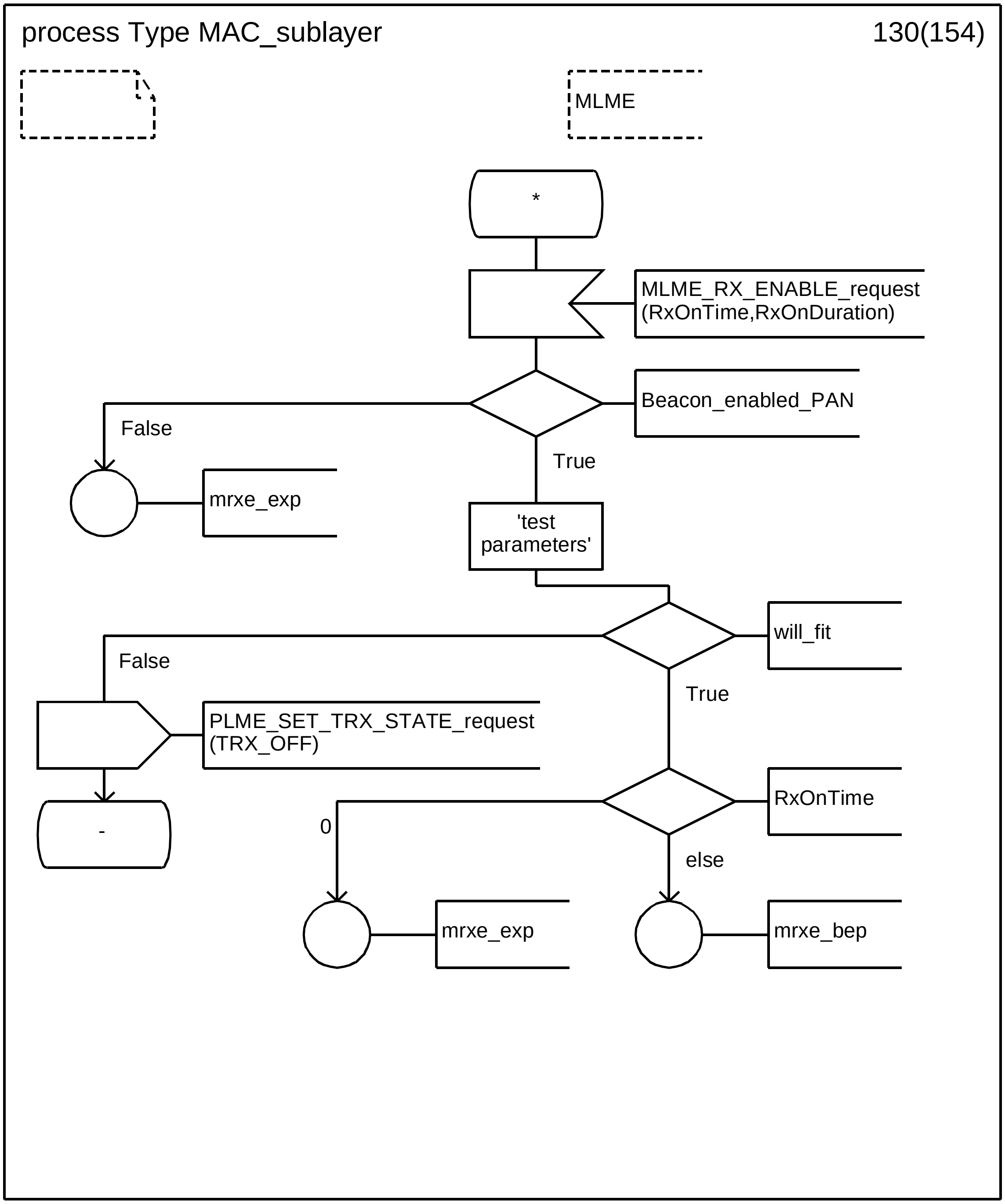


D.3.1.129 Process type MAC_sublayer (129)

.

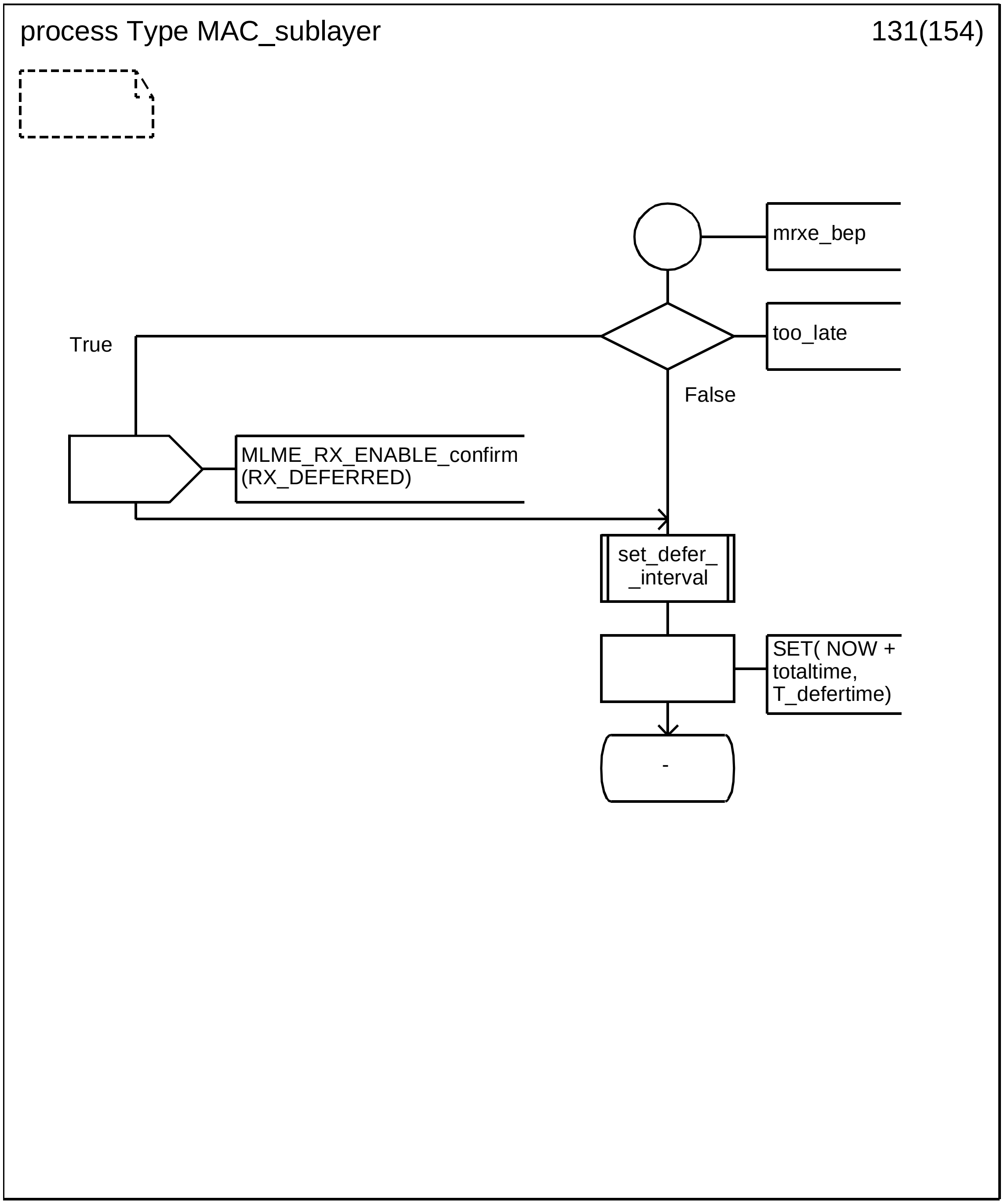

process Type MAC_sublayer
129(154)
ED
PLME_SET_TRX_STATE_confirm (phy_status1)
phy_status1
SUCCESS, RX_ON
TX_ON,TRX_OFF
ED_try_again
PLME_ED_request
set_beacon_ _scan_interval
SET (NOW + totaltime, T_scan_duration)
ED

D.3.1.130 Process type MAC_sublayer (130)



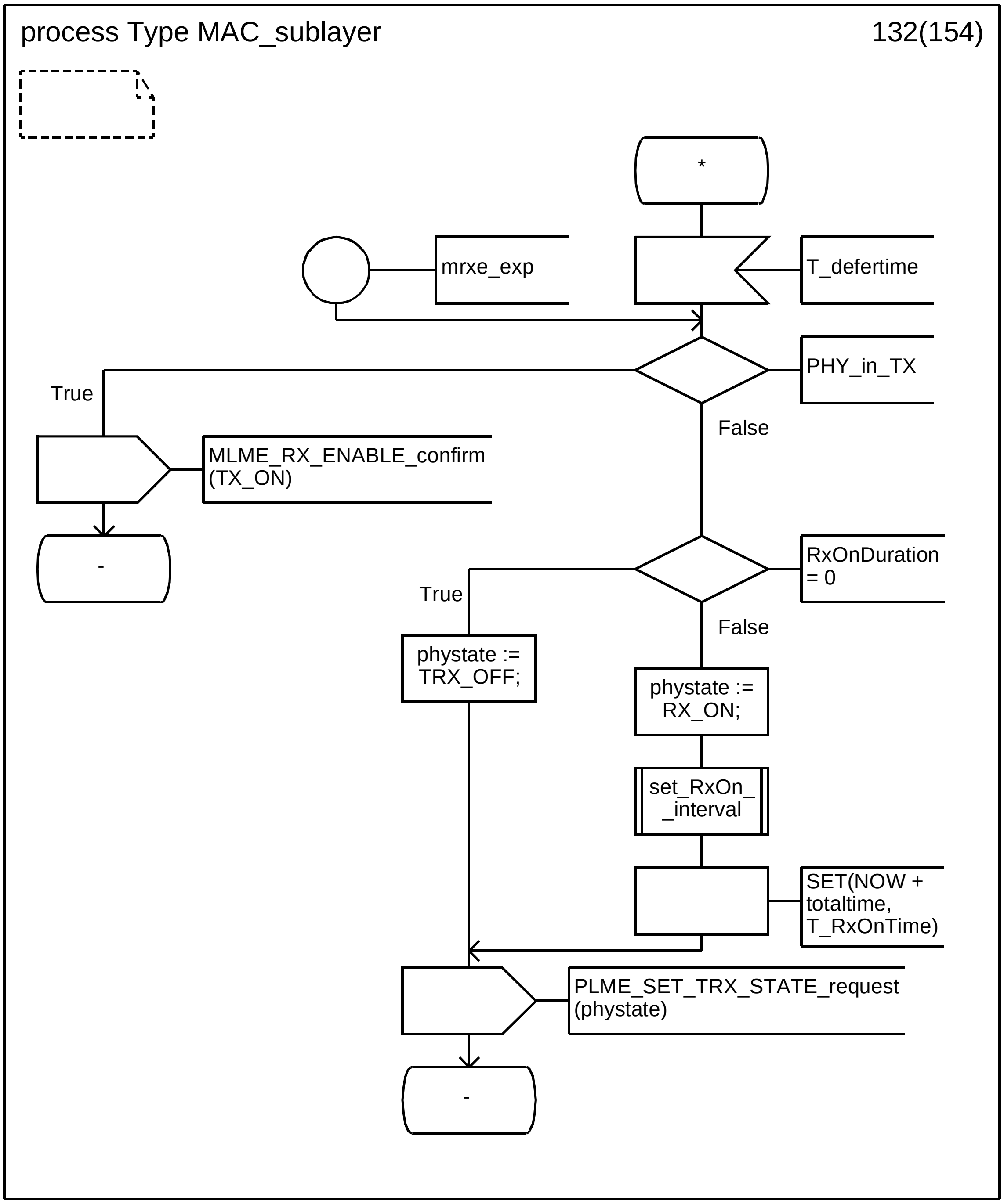
D.3.1.131 Process type MAC_sublayer (131)

.



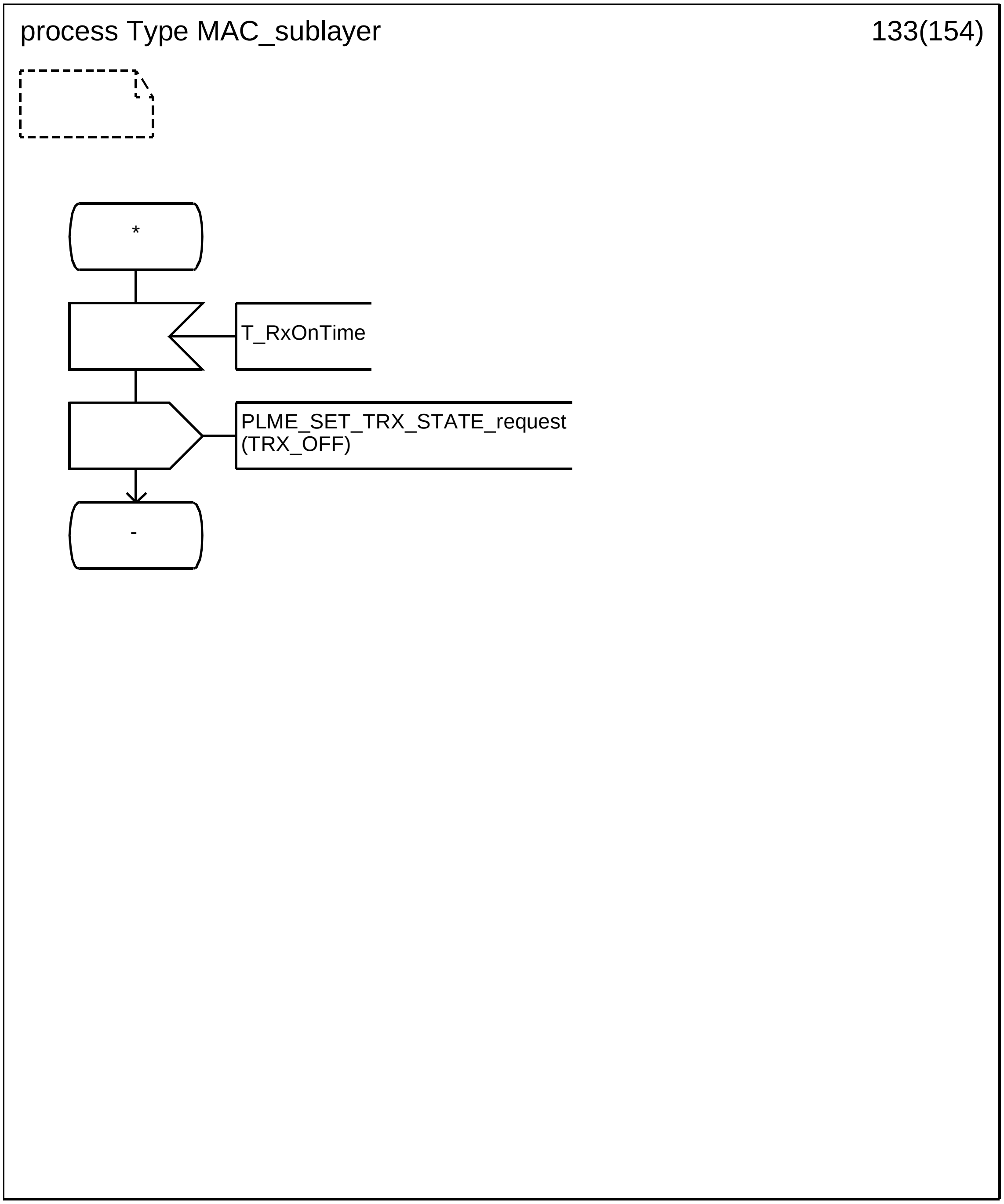
D.3.1.132 Process type MAC_sublayer (132)

.



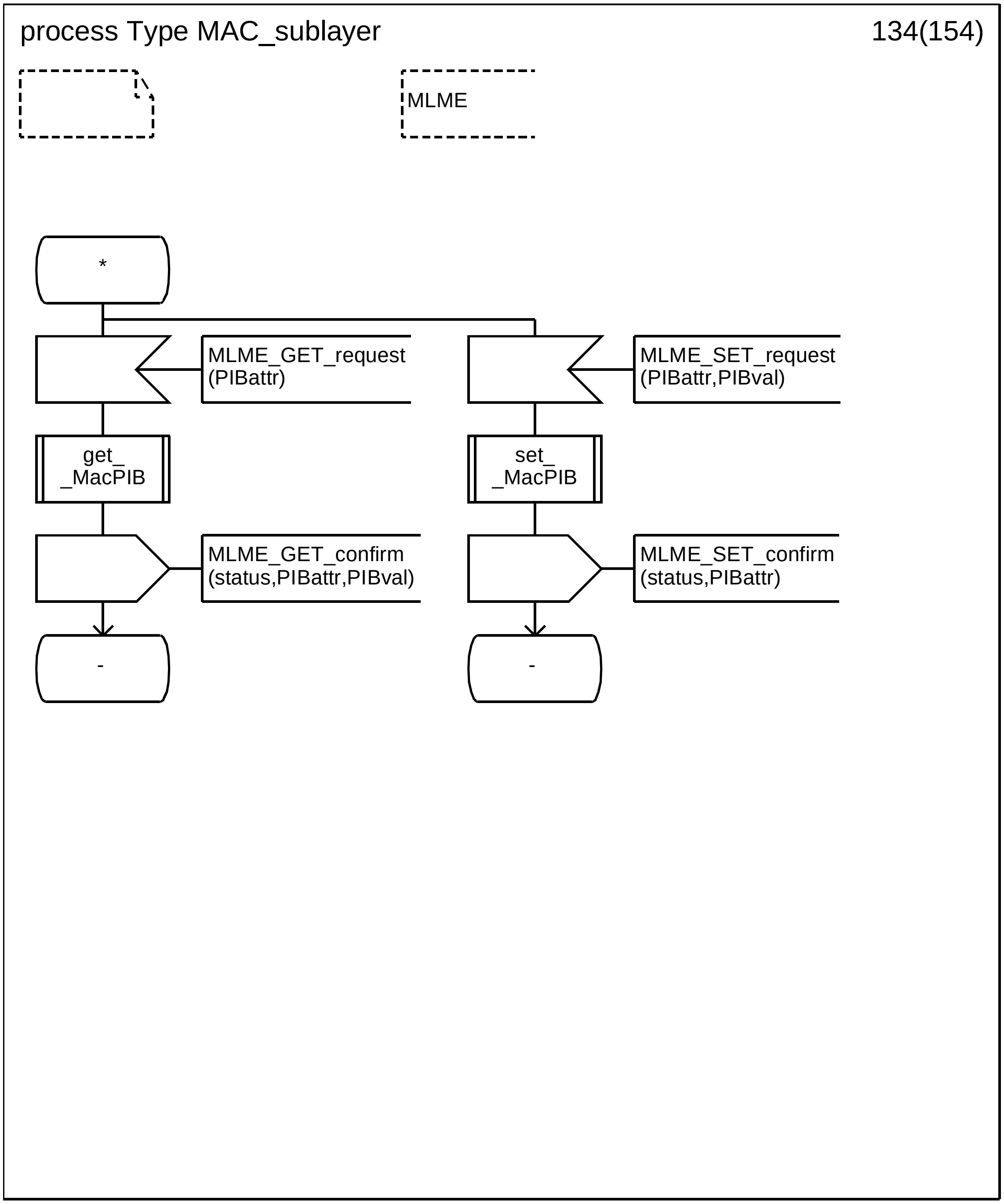
D.3.1.133 Process type MAC_sublayer (133)

.



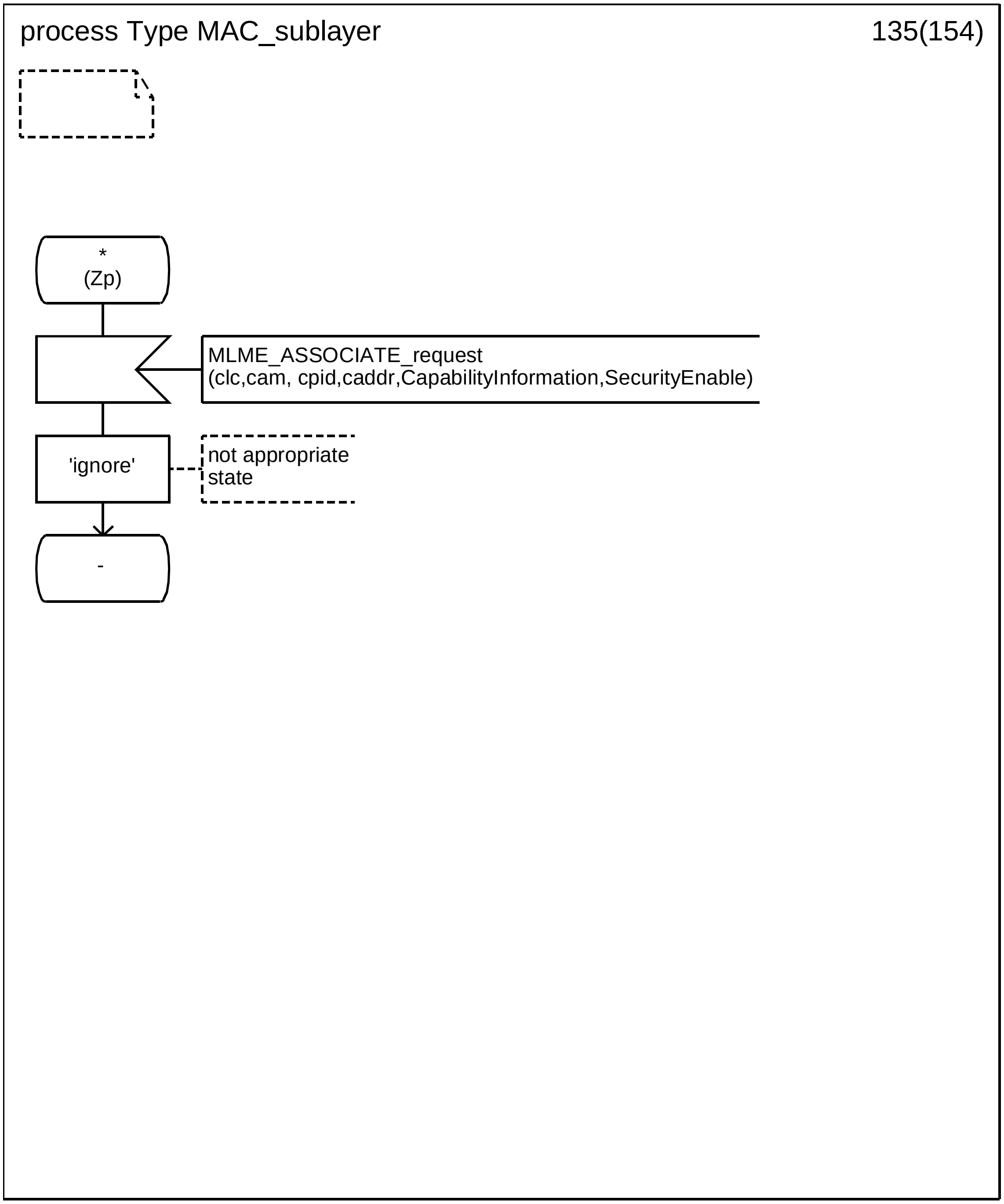
D.3.1.134 Process type MAC_sublayer (134)

.



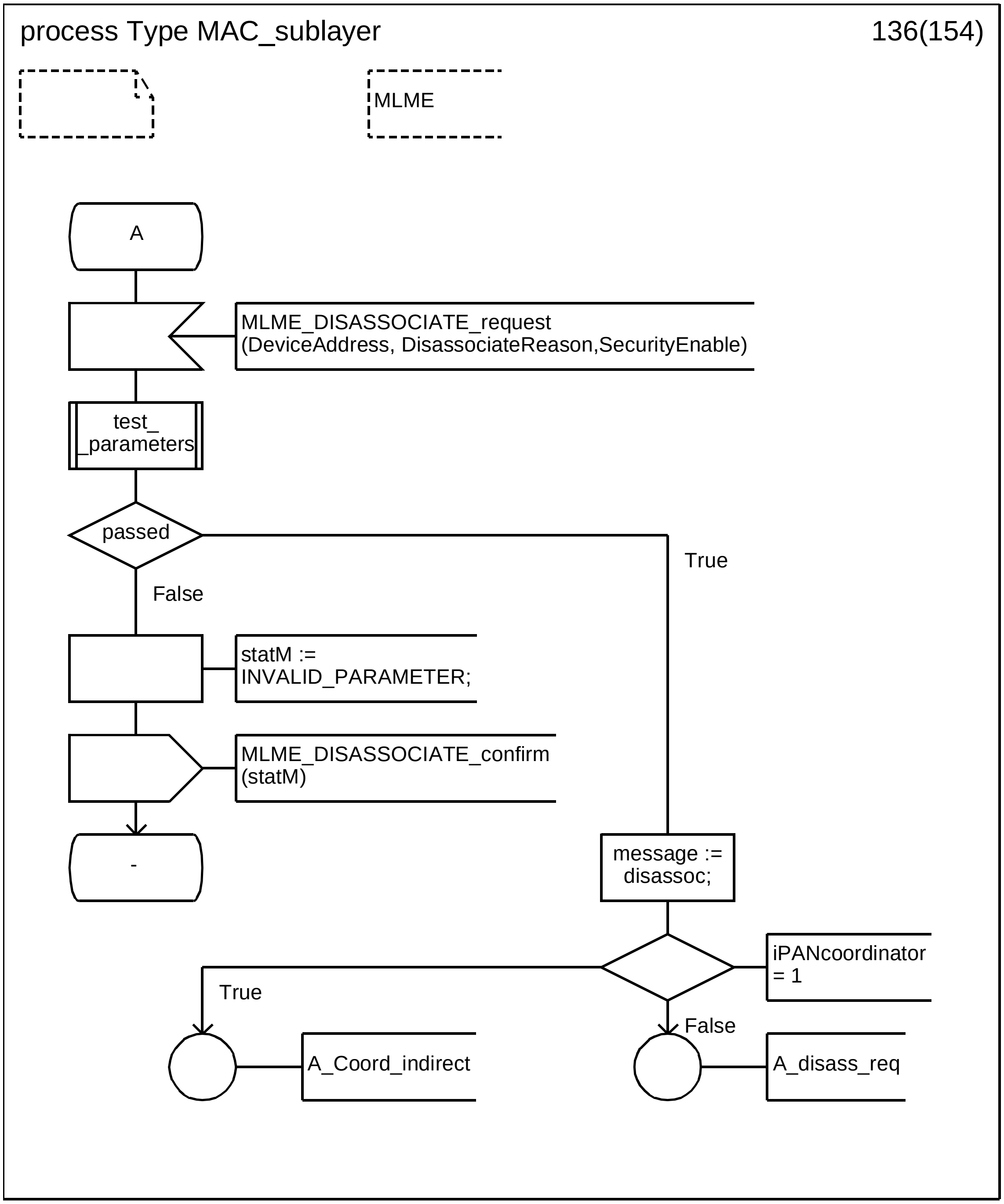
D.3.1.135 Process type MAC_sublayer (135)

.


process Type MAC_sublayer
135(154)
*
(Zp)
MLME_ASSOCIATE_request
(clc,cam, cpid,caddr,CapabilityInformation,SecurityEnable)
'ignore'
not appropriate
state
-

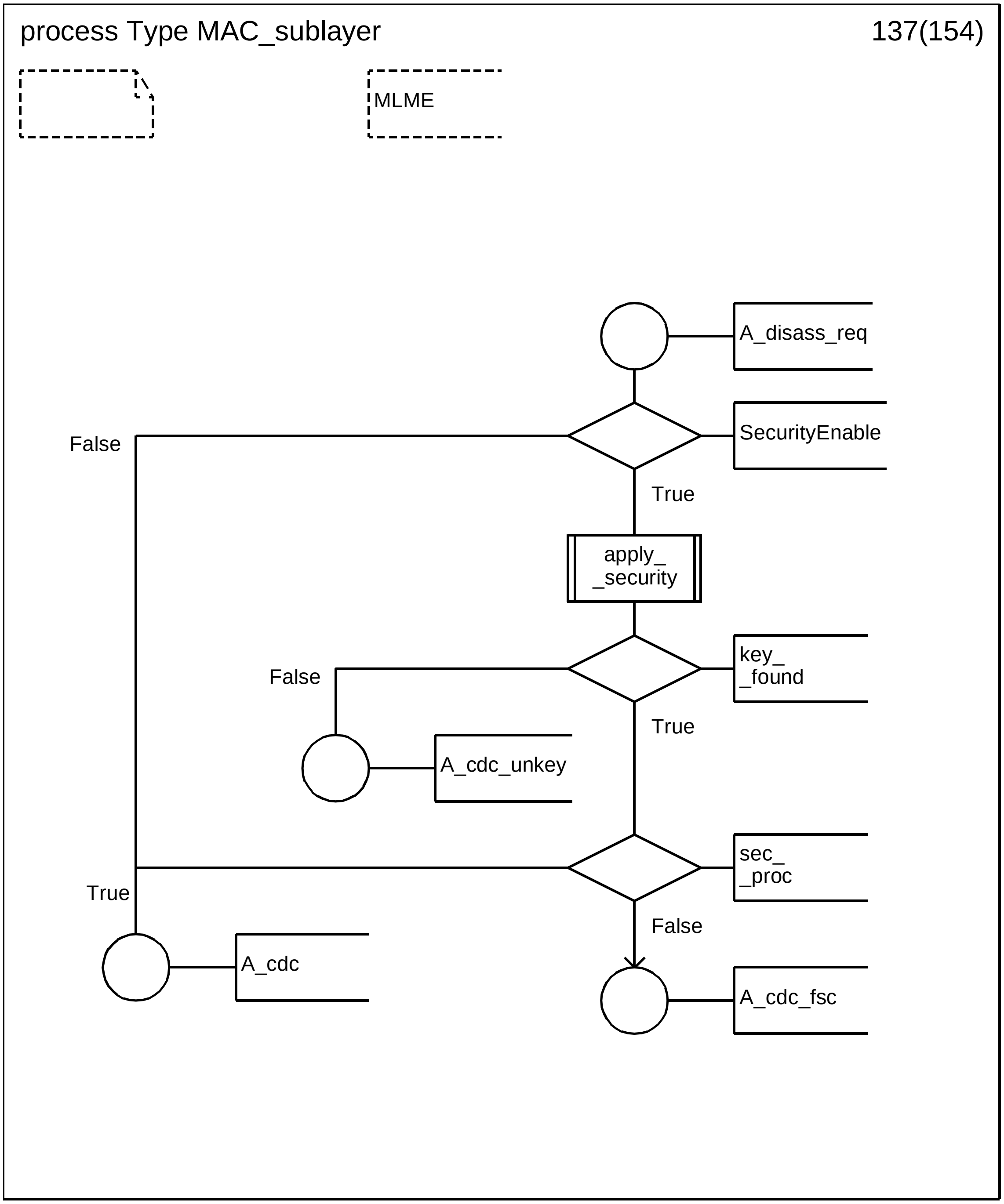
D.3.1.136 Process type MAC_sublayer (136)

.



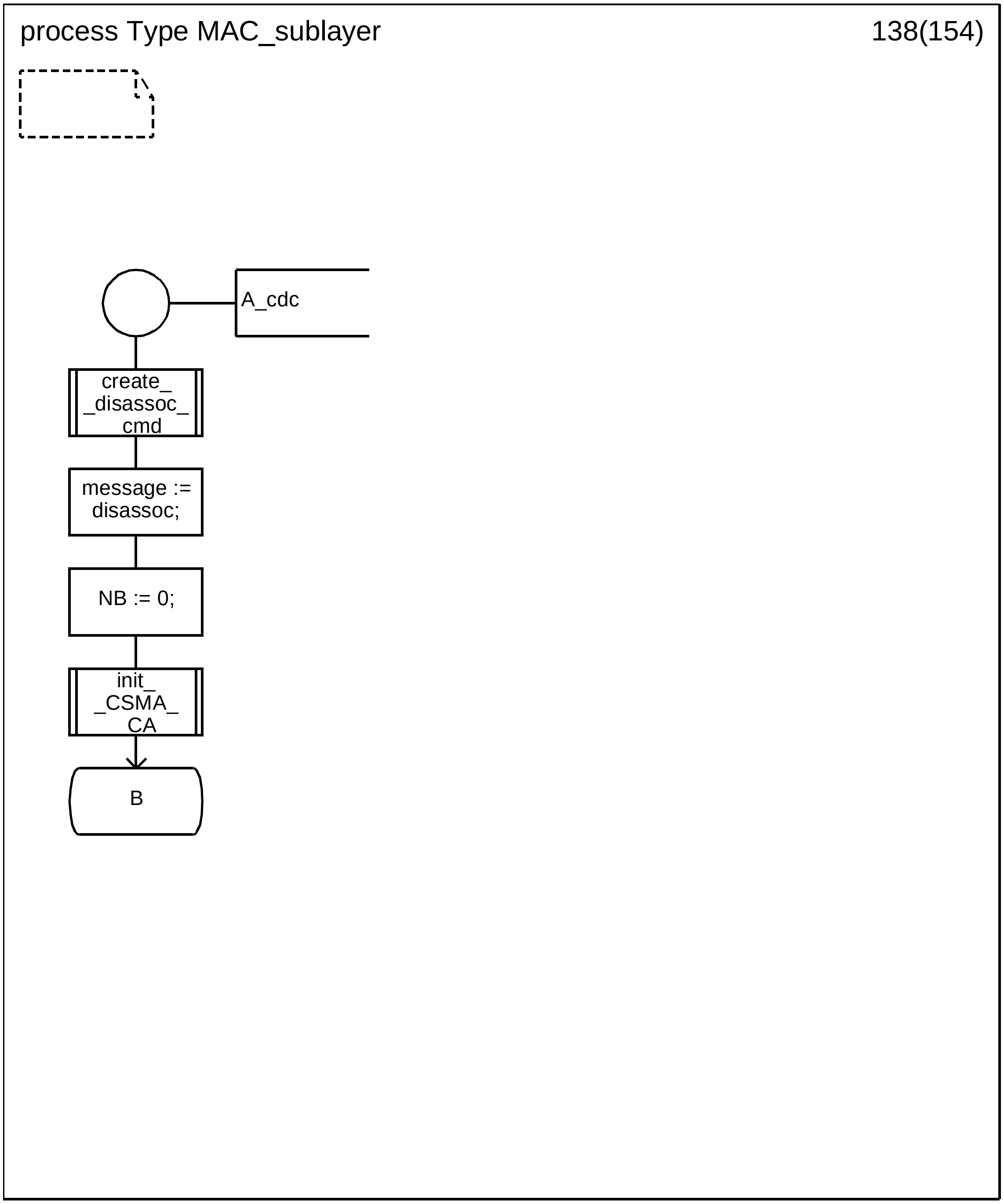
D.3.1.137 Process type MAC_sublayer (137)

.



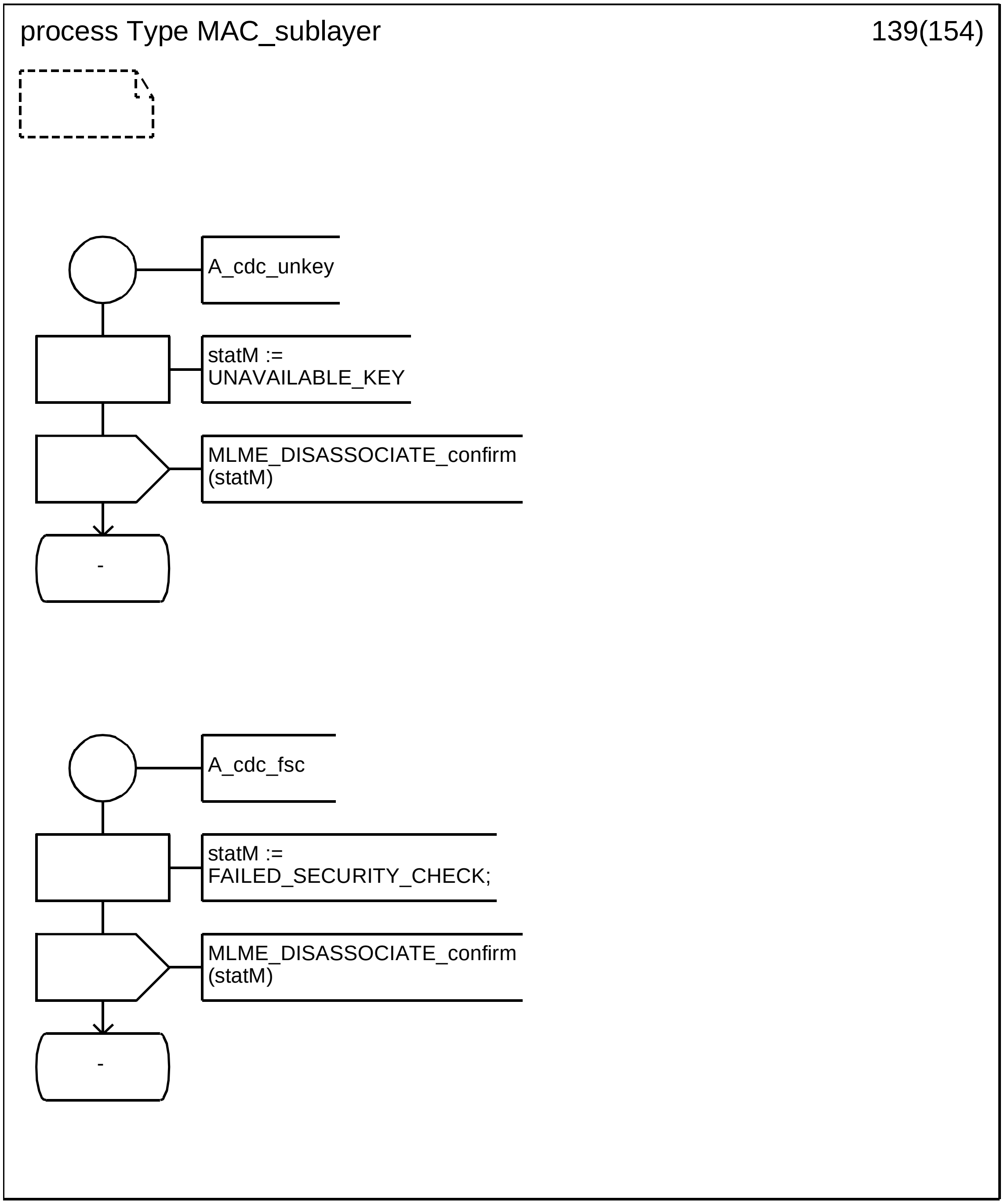
D.3.1.138 Process type MAC_sublayer (138)

r.

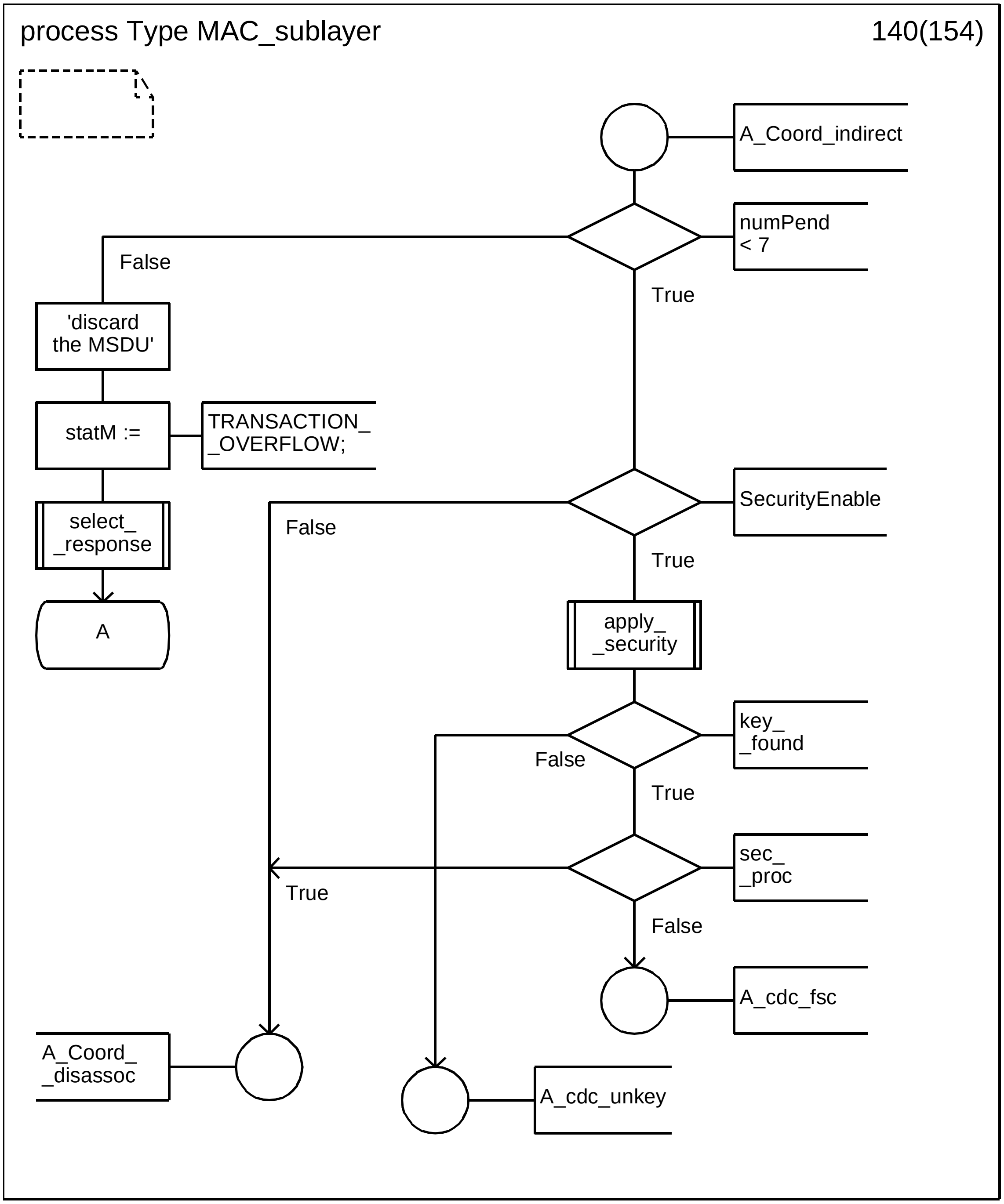


D.3.1.139 Process type MAC_sublayer (139)

.

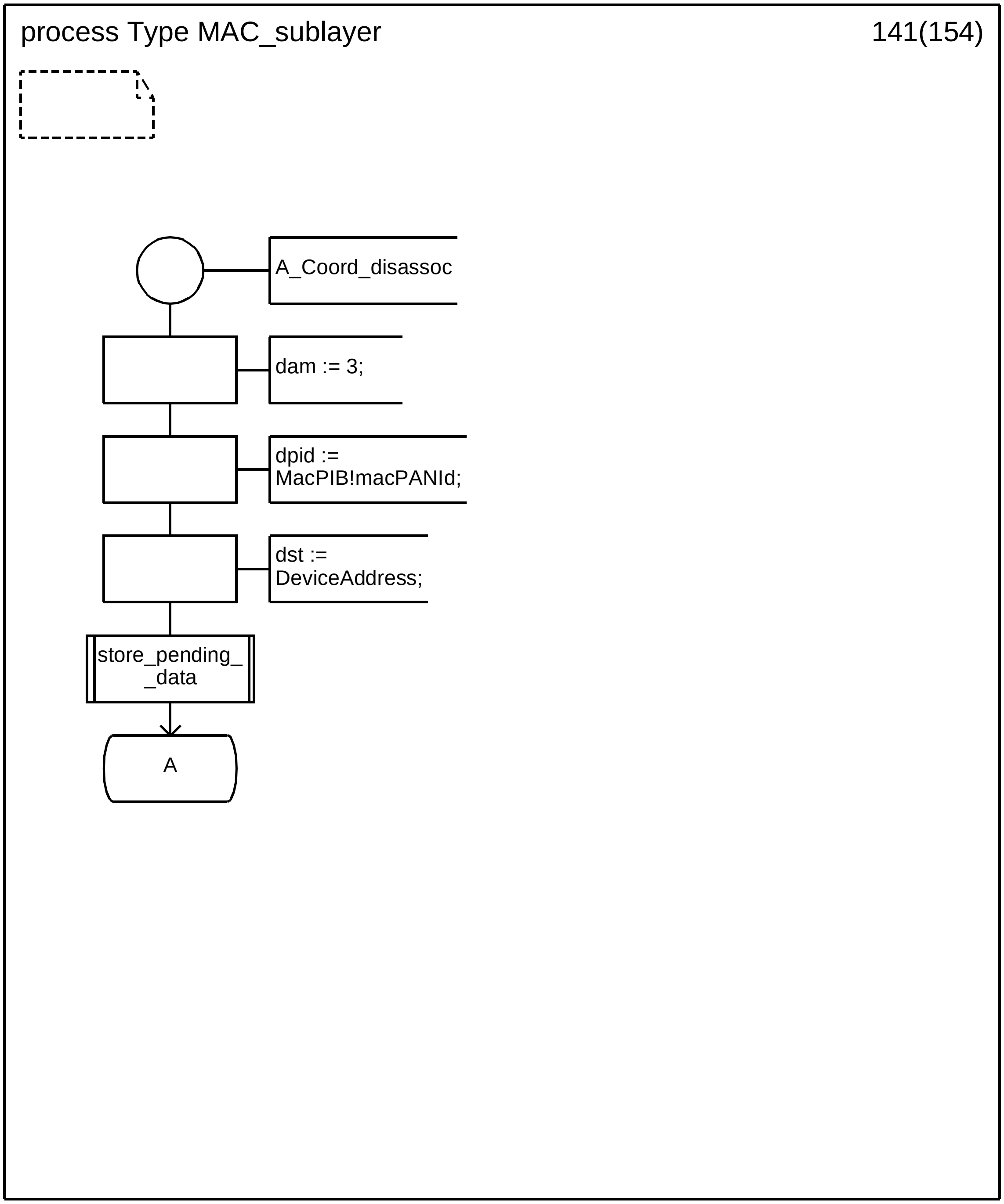


D.3.1.140 Process type MAC_sublayer (140)



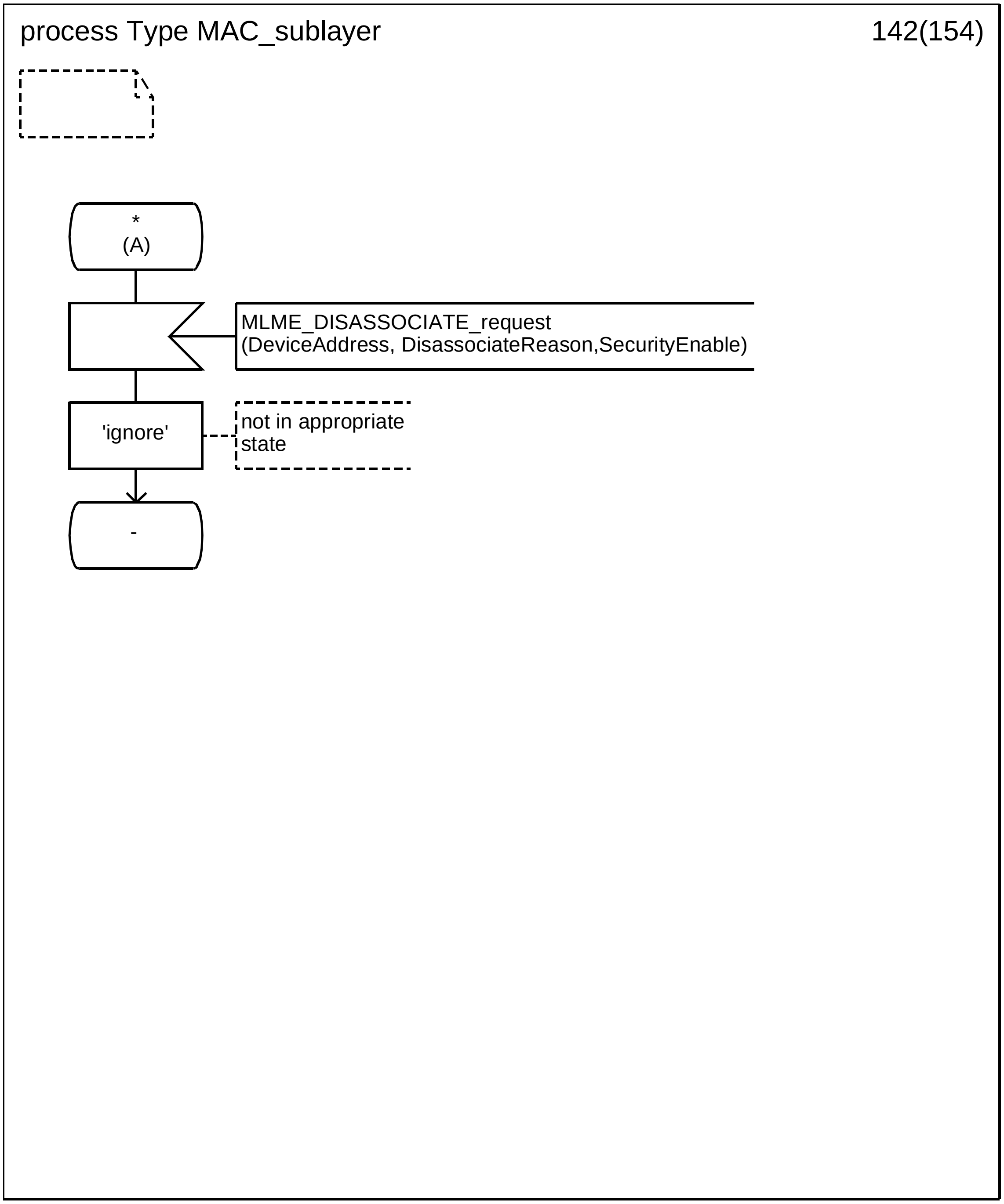
D.3.1.141 Process type MAC_sublayer (141)

r.



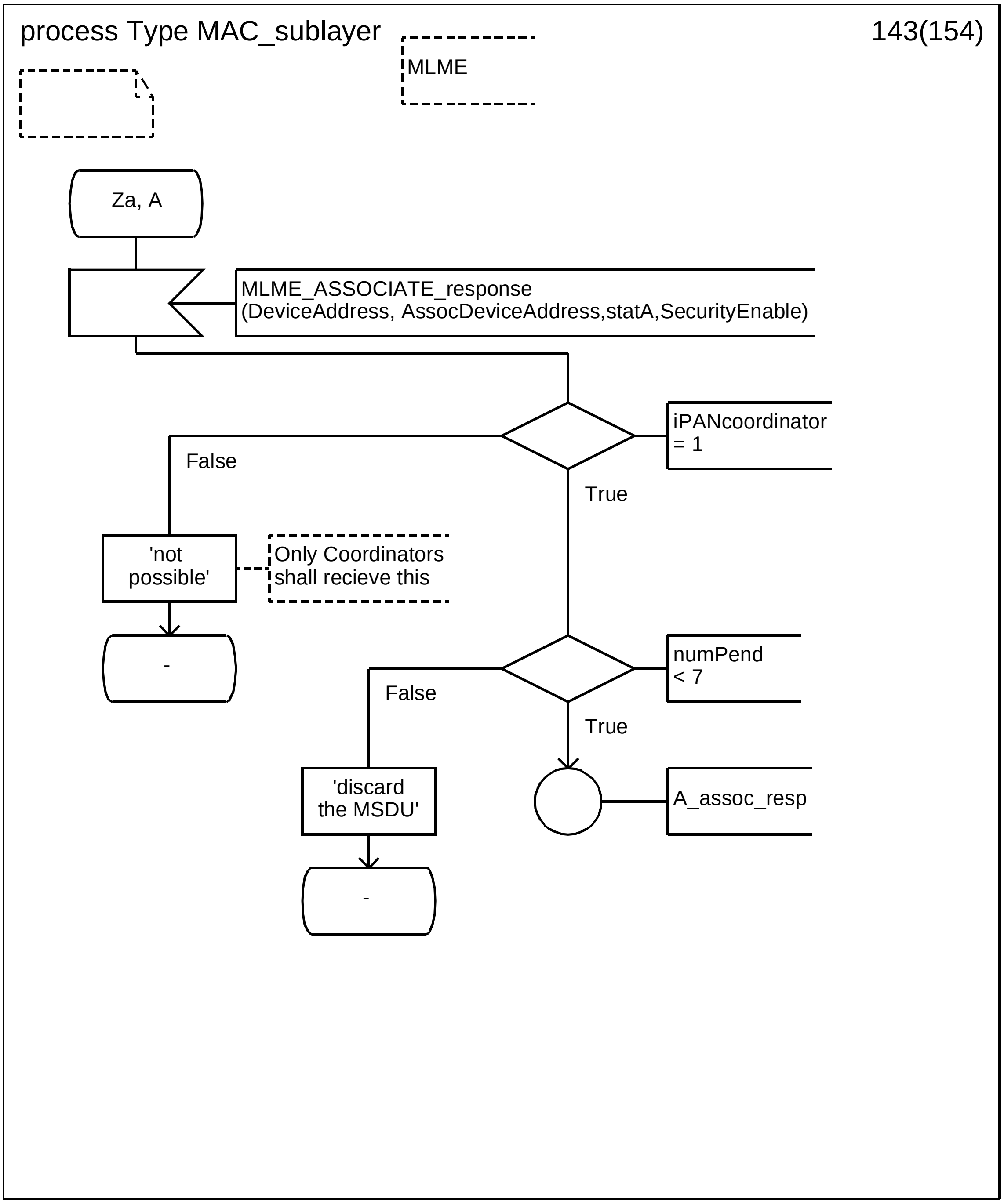
D.3.1.142 Process type MAC_sublayer (142)

.


process Type MAC_sublayer
142(154)
*
(A)
MLME_DISASSOCIATE_request
(DeviceAddress, DisassociateReason,SecurityEnable)
'ignore'
not in appropriate
state
-

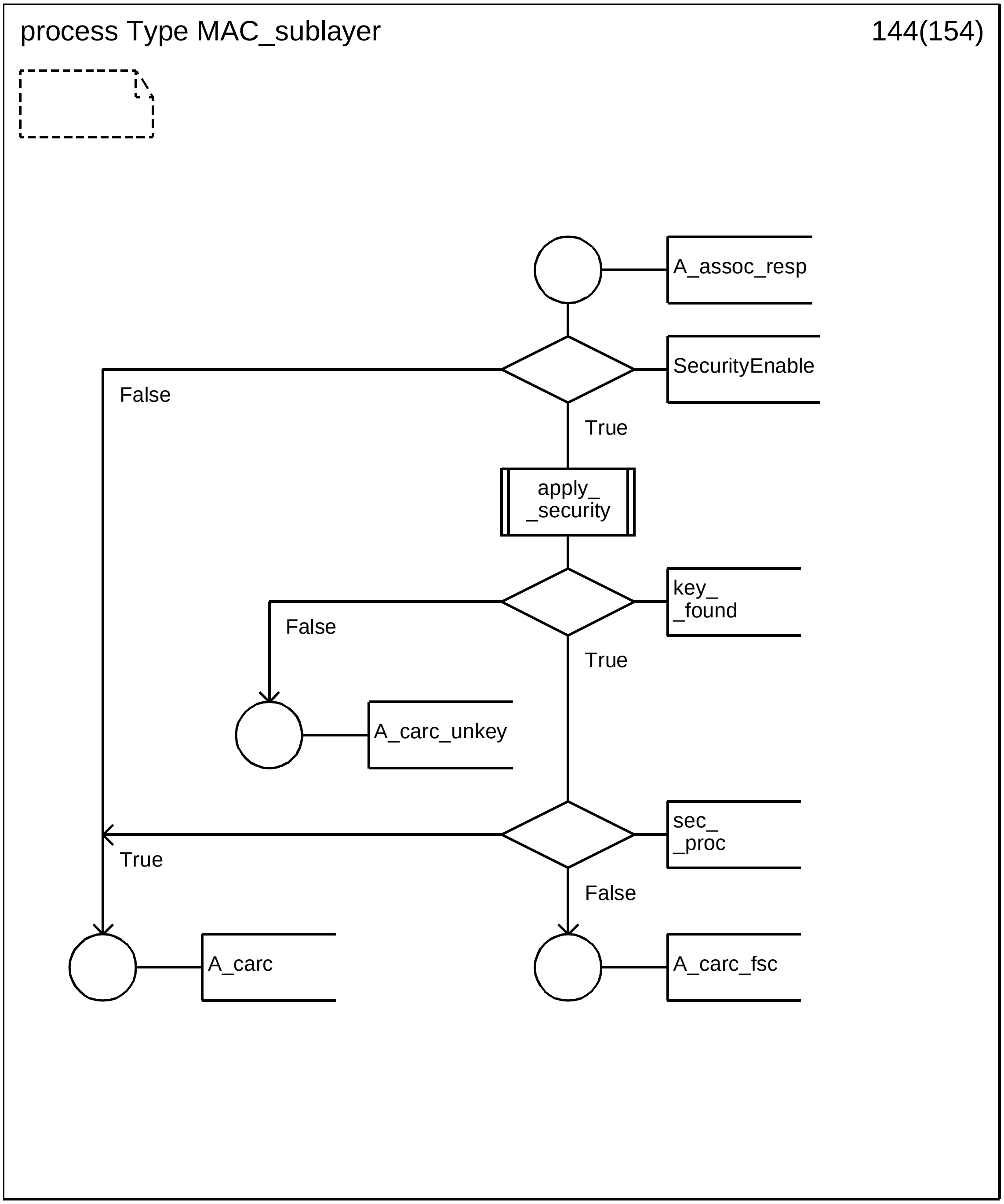
D.3.1.143 Process type MAC_sublayer (143)

.



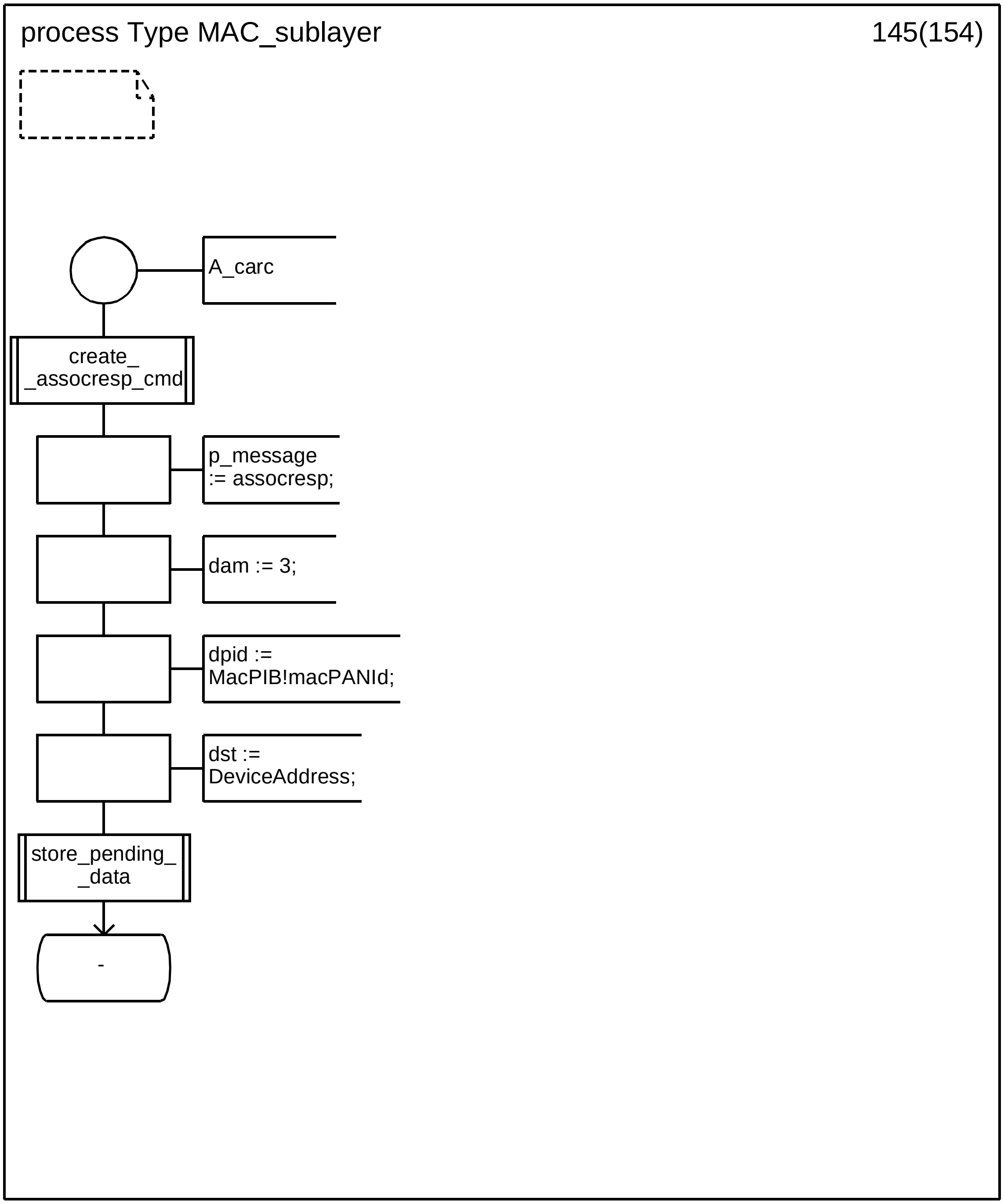
D.3.1.144 Process type MAC_sublayer (144)

.



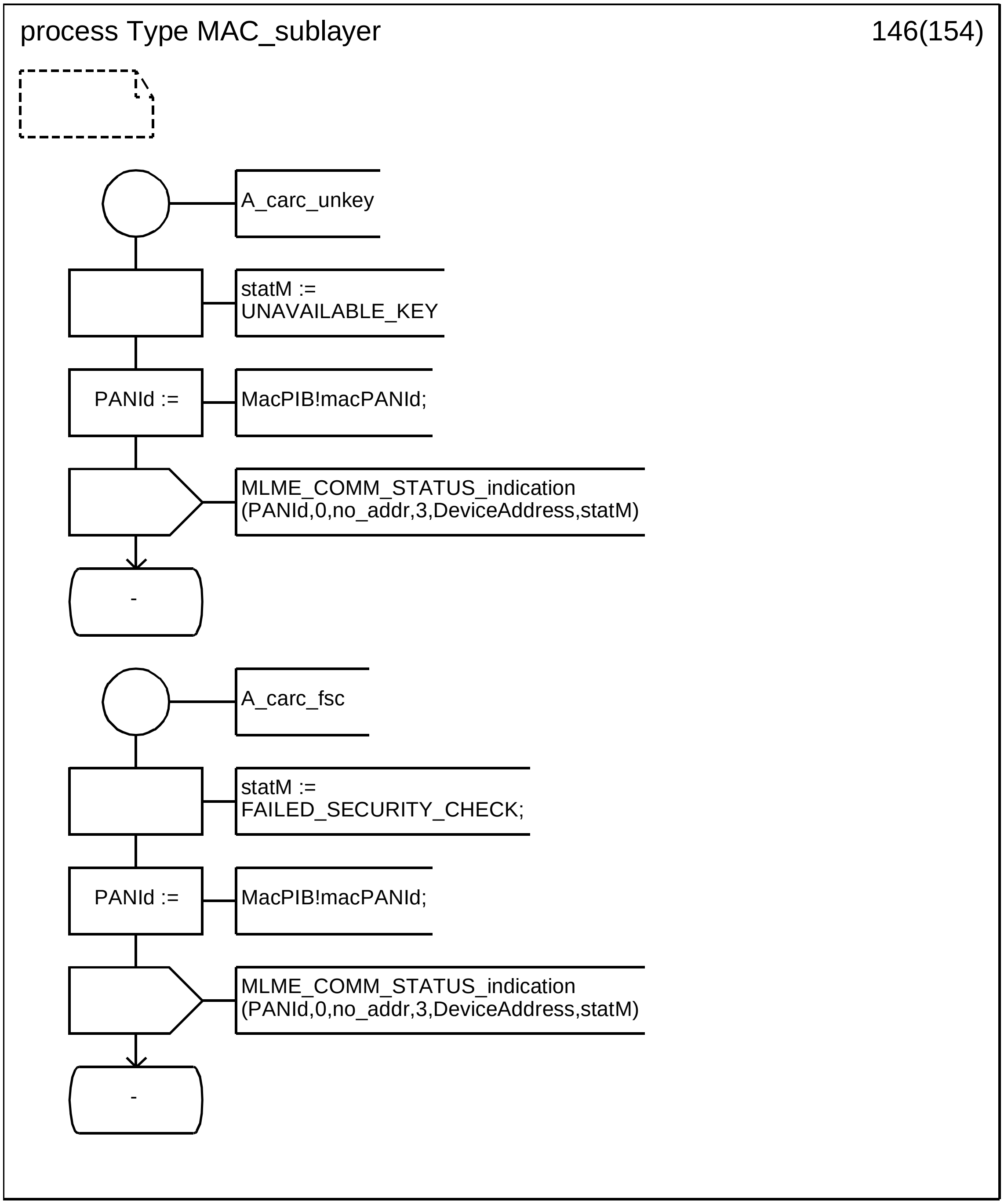
D.3.1.145 Process type MAC_sublayer (145)

.



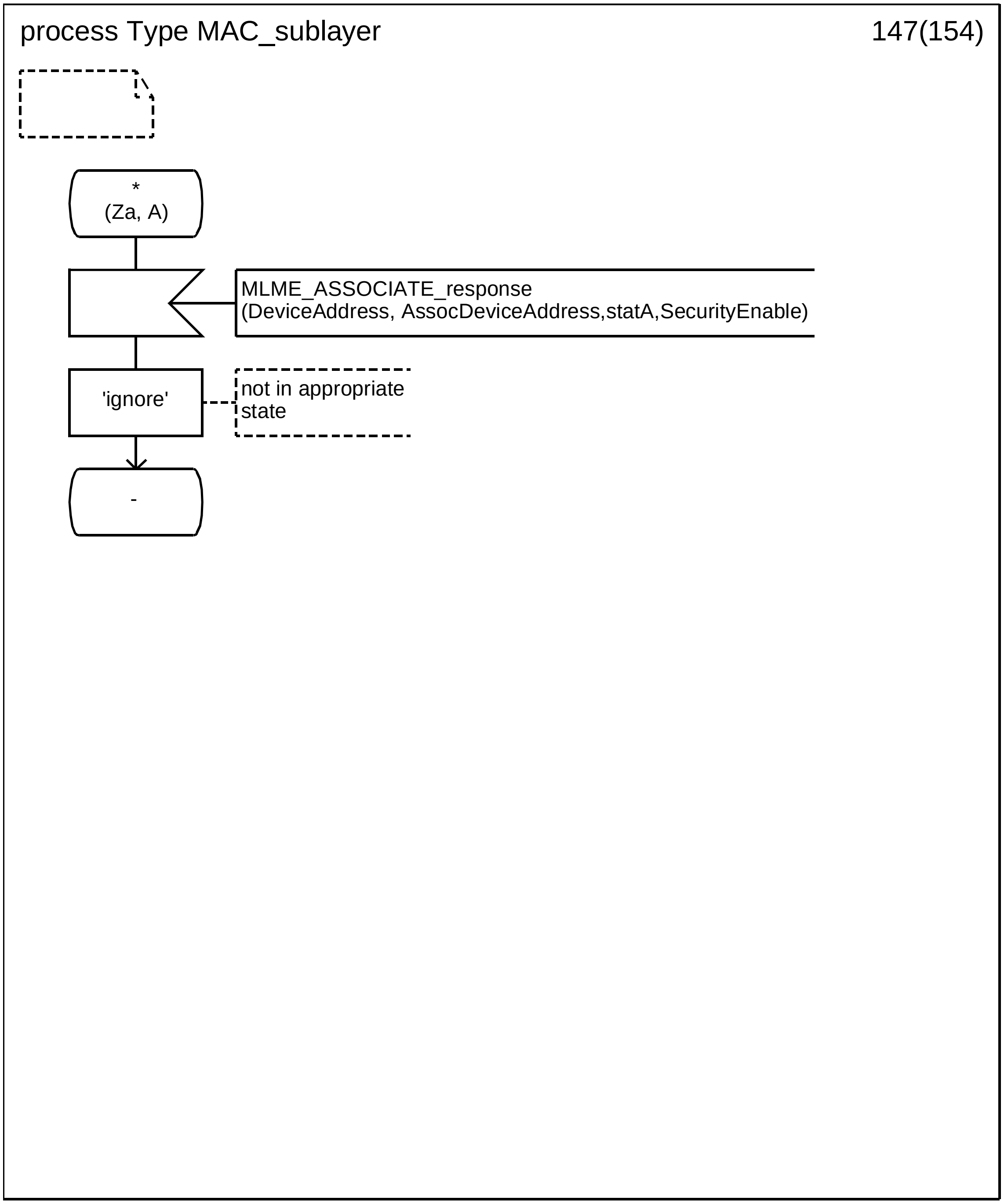
D.3.1.146 Process type MAC_sublayer (146)

.


process Type MAC_sublayer
146(154)
A_carc_unkey
statM :=
UNAVAILABLE_KEY
PANId :=
MacPIB!macPANId;
MLME_COMM_STATUS_indication
(PANId,0,no_addr,3,DeviceAddress,statM)
-
A_carc_fsc
statM :=
FAILED_SECURITY_CHECK;
PANId :=
MacPIB!macPANId;
MLME_COMM_STATUS_indication
(PANId,0,no_addr,3,DeviceAddress,statM)
-

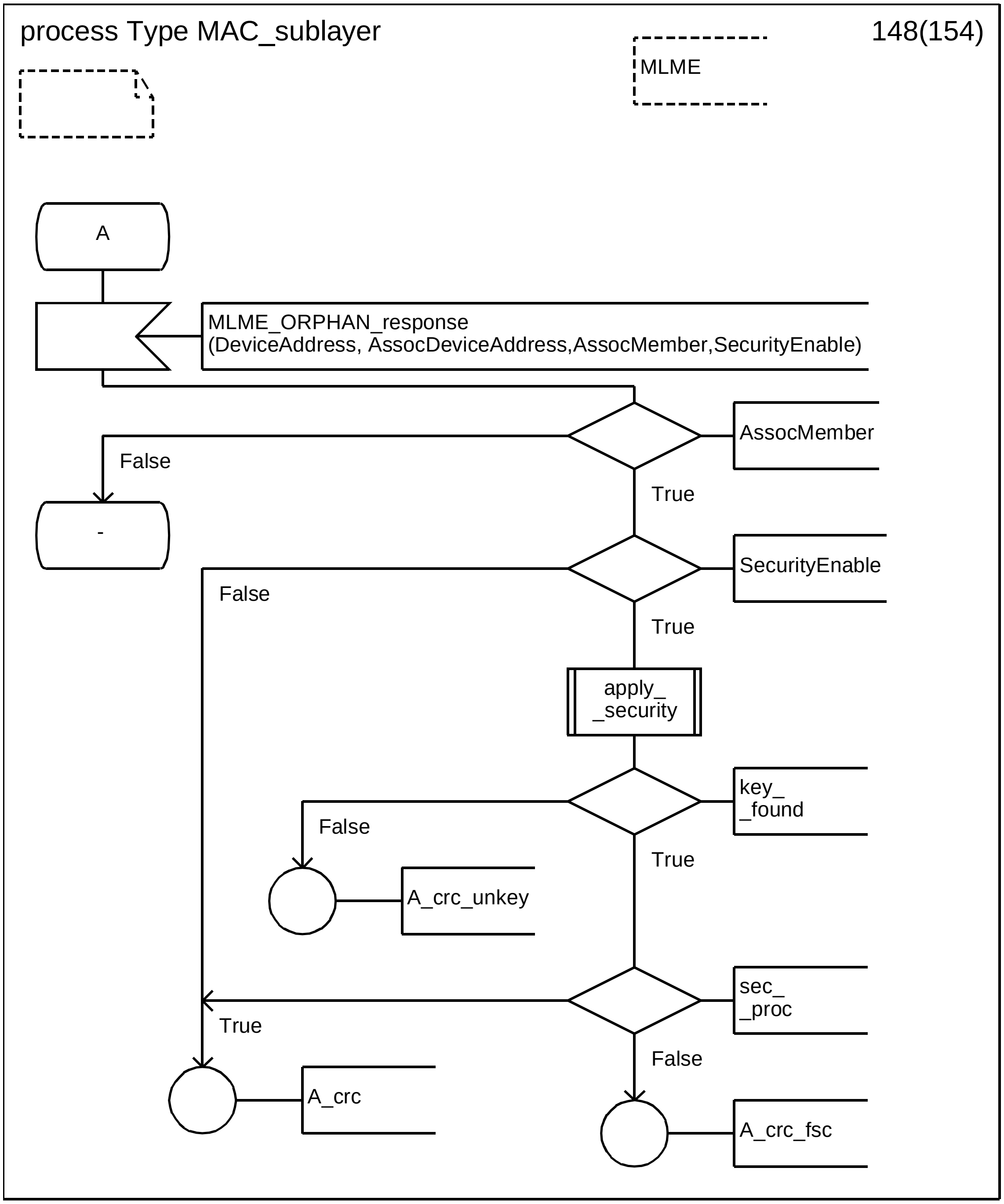
D.3.1.147 Process type MAC_sublayer (147)

.

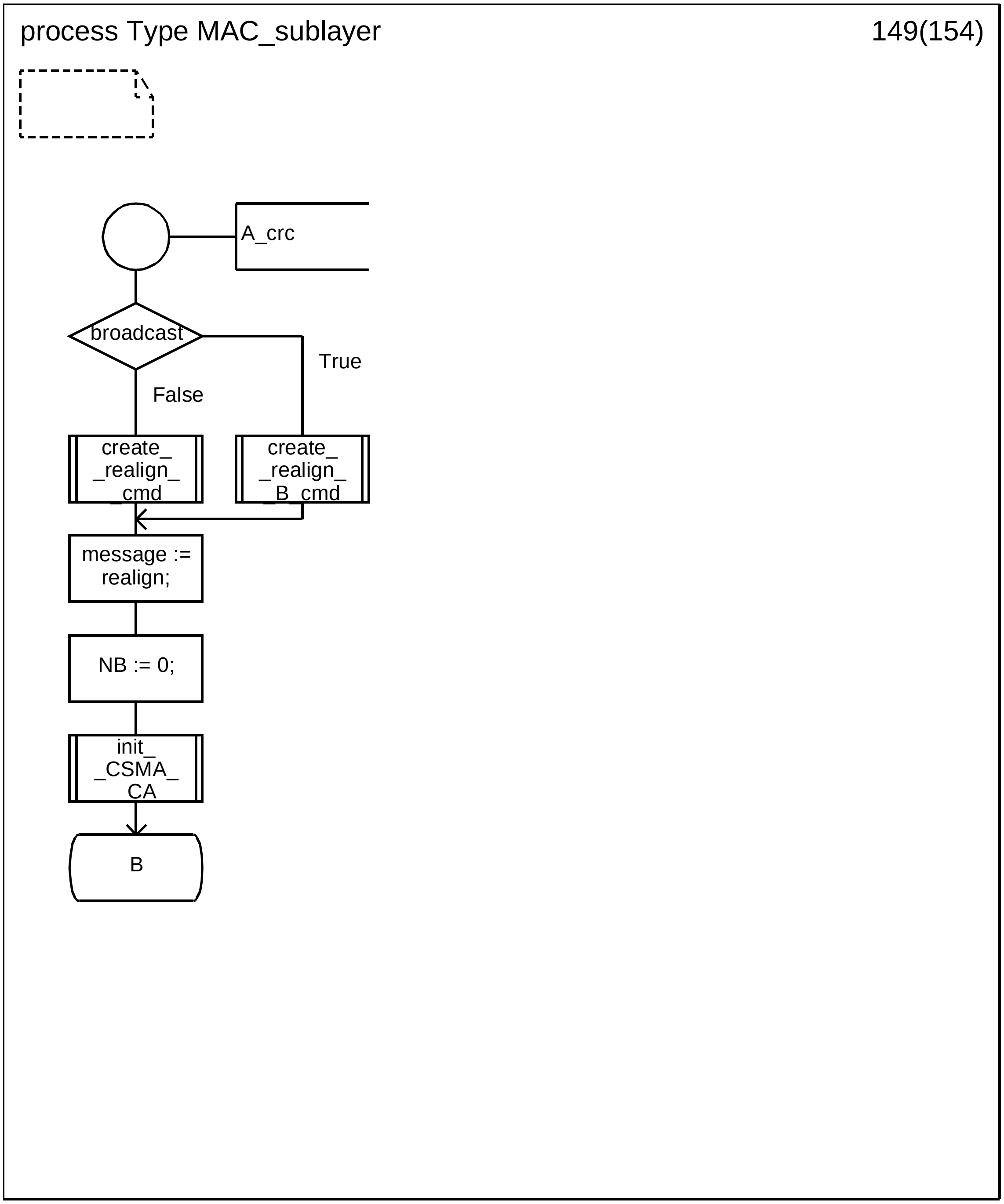

process Type MAC_sublayer
147(154)
*
(Za, A)
MLME_ASSOCIATE_response
(DeviceAddress, AssocDeviceAddress,statA,SecurityEnable)
'ignore'
not in appropriate
state
-

D.3.1.148 Process type MAC_sublayer (148)

.

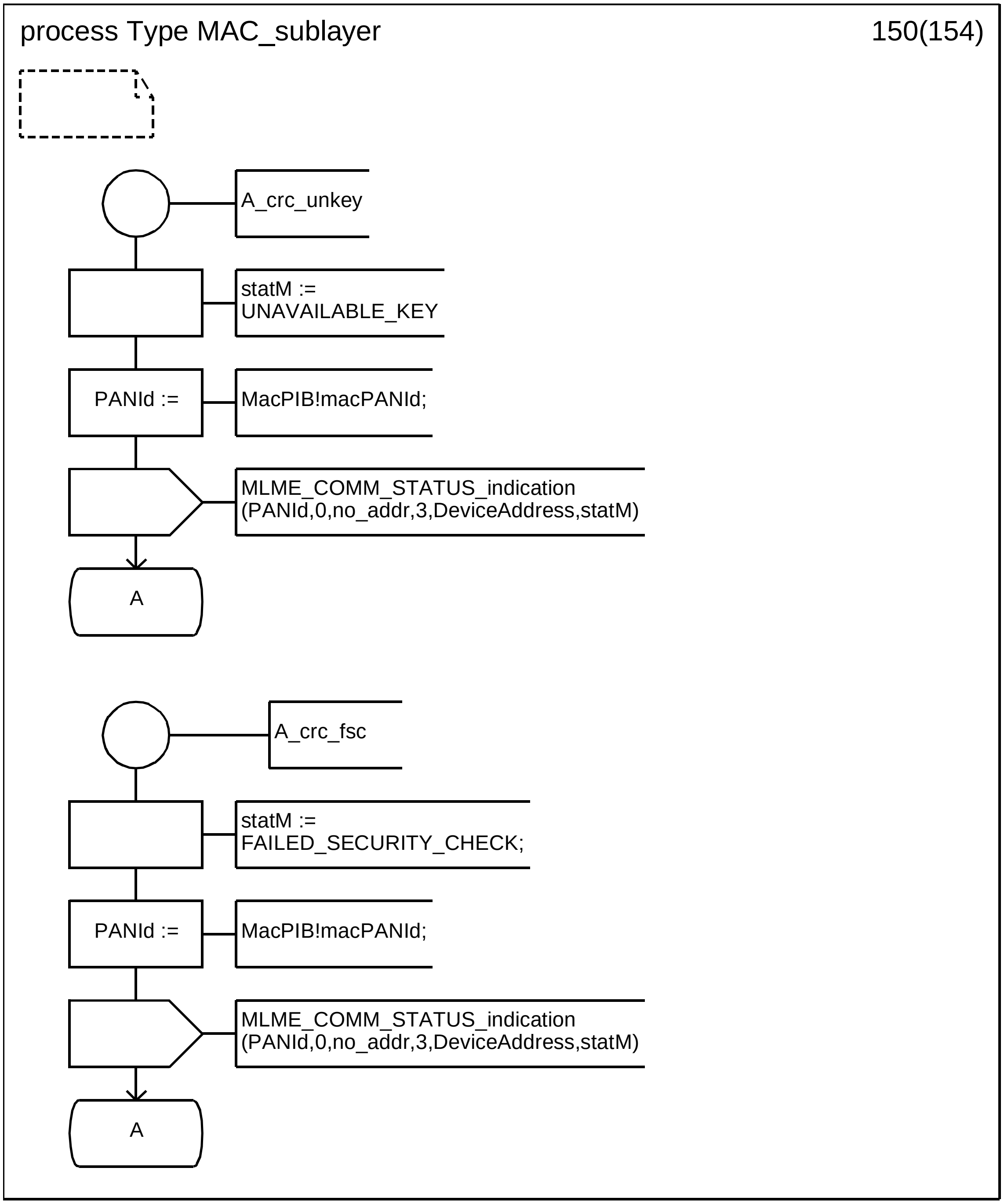


D.3.1.149 Process type MAC_sublayer (149)



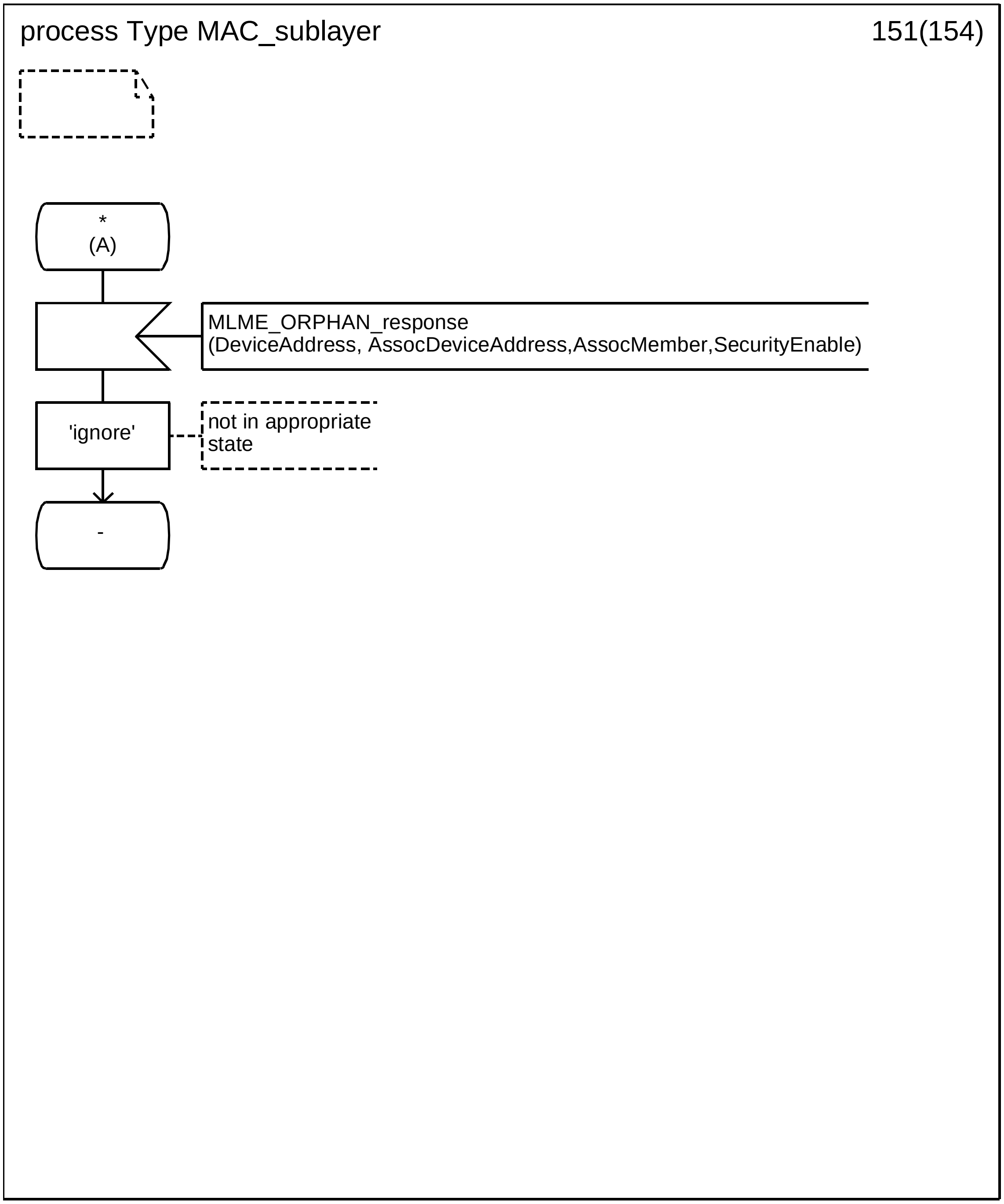
D.3.1.150 Process type MAC_sublayer (150)

.


process Type MAC_sublayer
150(154)
A_crc_unkey
statM :=
UNAVAILABLE_KEY
PANId :=
MacPIB!macPANId;
MLME_COMM_STATUS_indication
(PANId,0,no_addr,3,DeviceAddress,statM)
A
A_crc_fsc
statM :=
FAILED_SECURITY_CHECK;
PANId :=
MacPIB!macPANId;
MLME_COMM_STATUS_indication
(PANId,0,no_addr,3,DeviceAddress,statM)
A

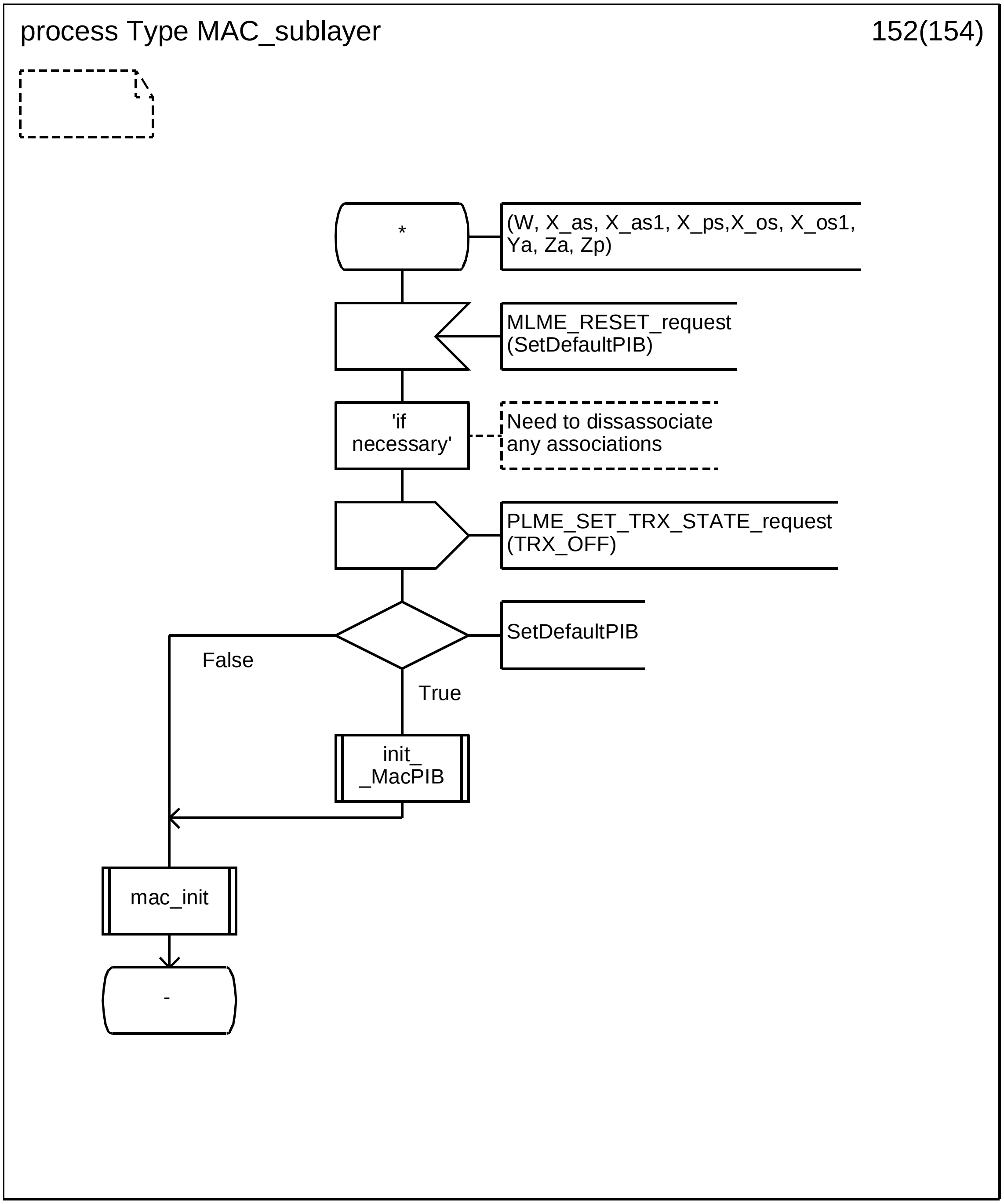
D.3.1.151 Process type MAC_sublayer (151)

.


process Type MAC_sublayer
151(154)
*
(A)
MLME_ORPHAN_response
(DeviceAddress, AssocDeviceAddress,AssocMember,SecurityEnable)
'ignore'
not in appropriate
state
-

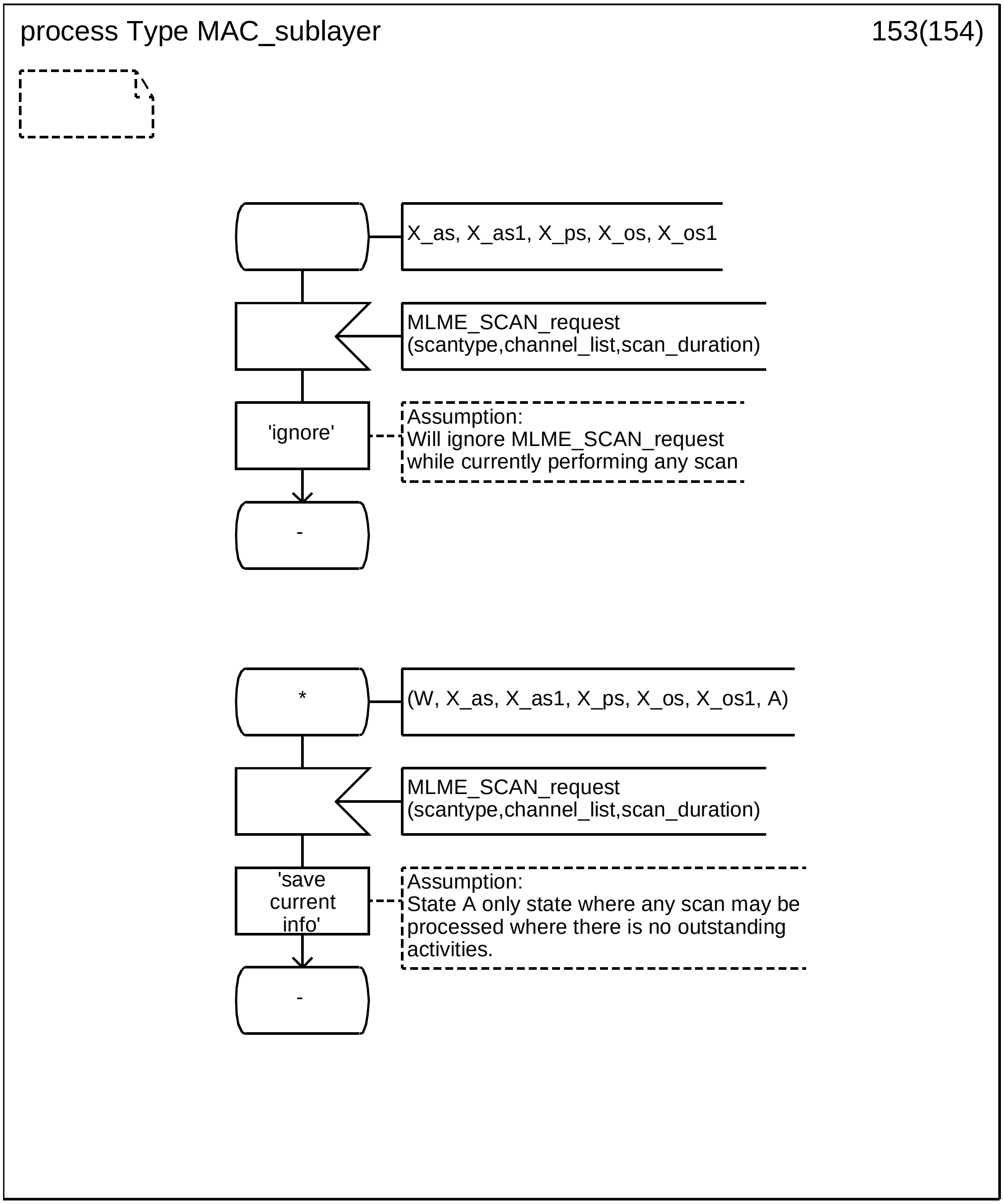
D.3.1.152 Process type MAC_sublayer (152)

.


process Type MAC_sublayer
152(154)
*
(W, X_as, X_as1, X_ps,X_os, X_os1, Ya, Za, Zp)
MLME_RESET_request (SetDefaultPIB)
'if necessary'
Need to dissassociate any associations
PLME_SET_TRX_STATE_request (TRX_OFF)
SetDefaultPIB
False
True
init_ _MacPIB
mac_init
-

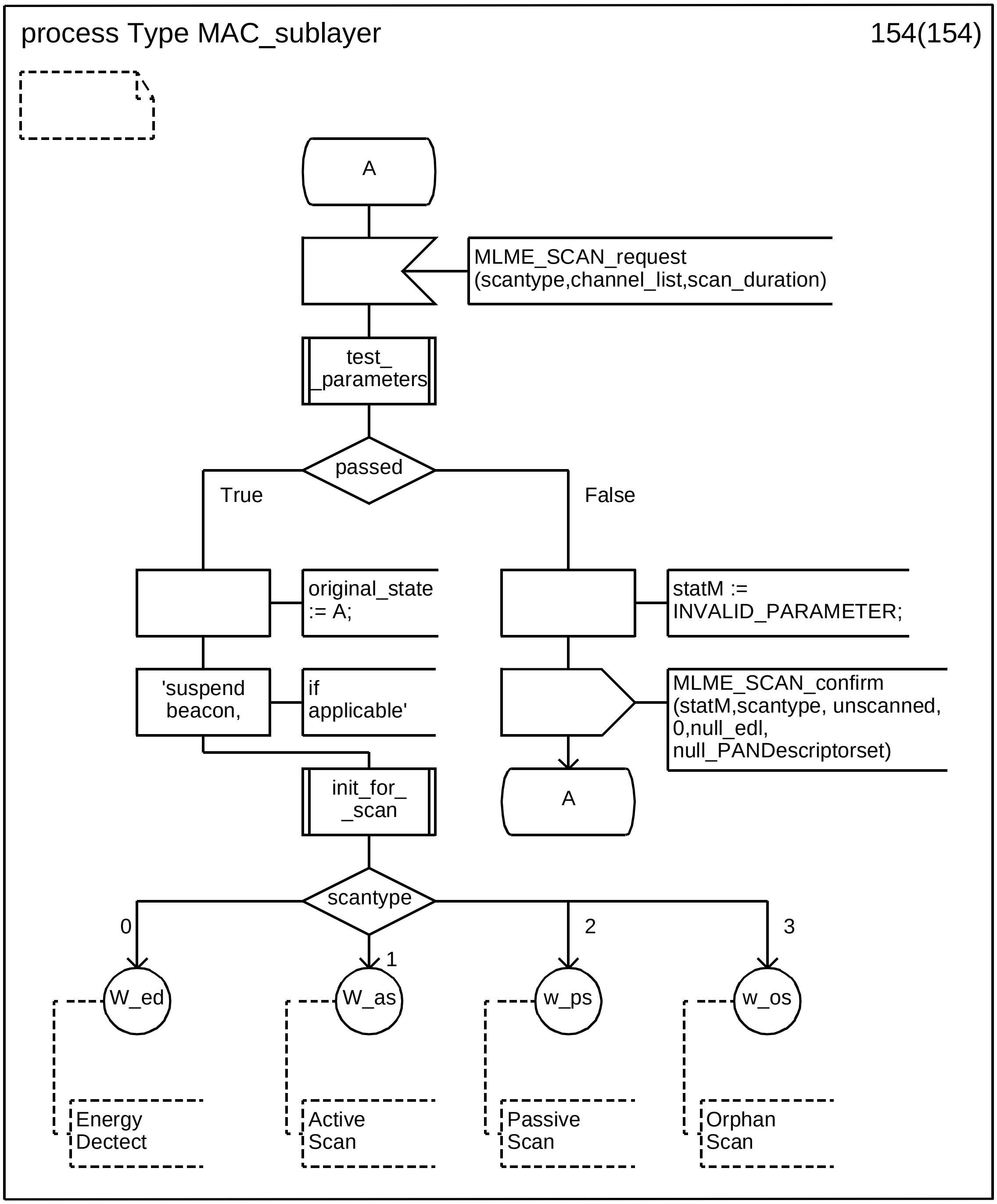
D.3.1.153 Process type MAC_sublayer (153)

.


process Type MAC_sublayer
153(154)
X_as, X_as1, X_ps, X_os, X_os1
MLME_SCAN_request
(scantype,channel_list,scan_duration)
'ignore'
Assumption:
Will ignore MLME_SCAN_request
while currently performing any scan
-
*
(W, X_as, X_as1, X_ps, X_os, X_os1, A)
MLME_SCAN_request
(scantype,channel_list,scan_duration)
'save
current
info'
Assumption:
State A only state where any scan may be
processed where there is no outstanding
activities.
-

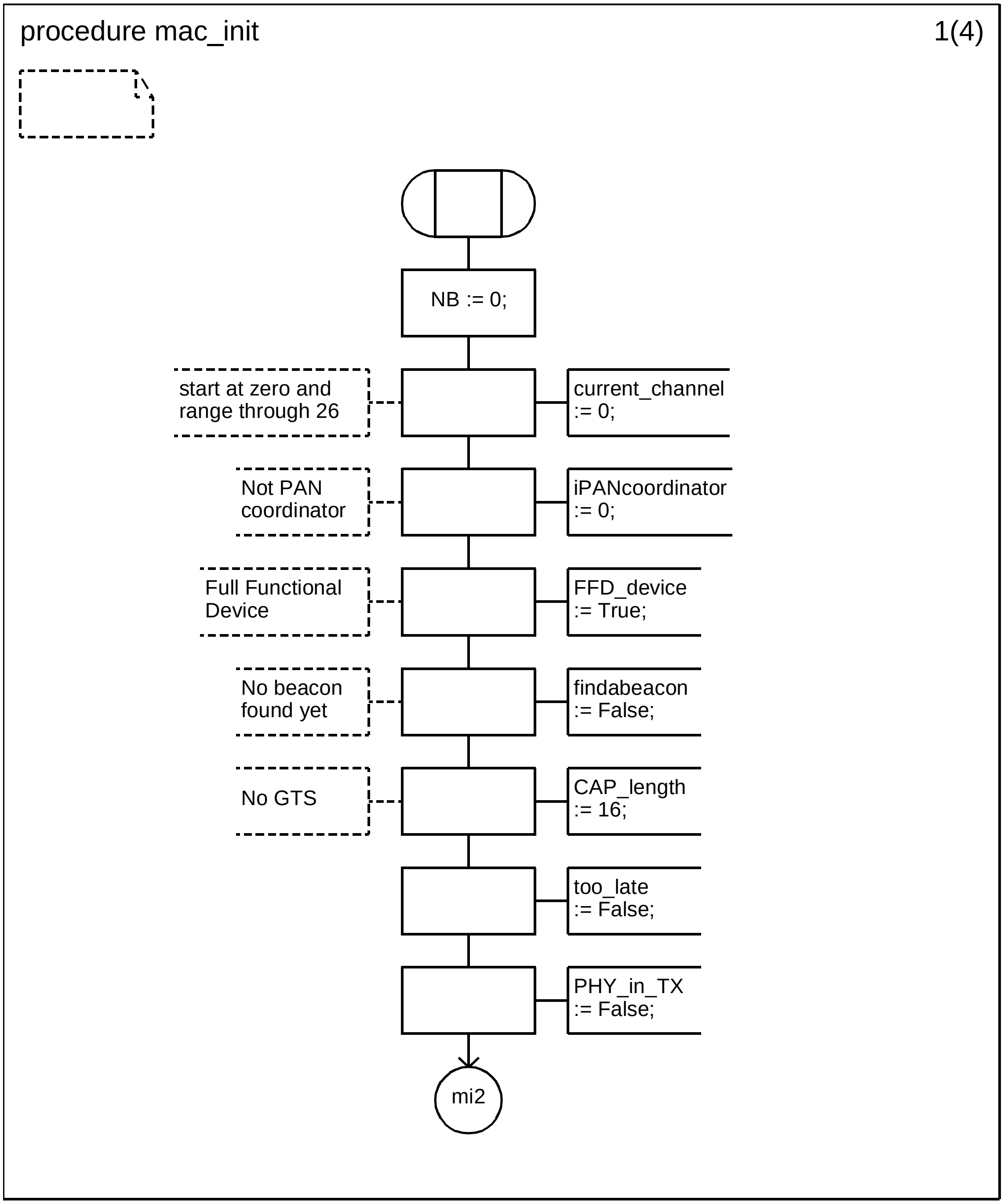
D.3.1.154 Process type MAC_sublayer (154)

.

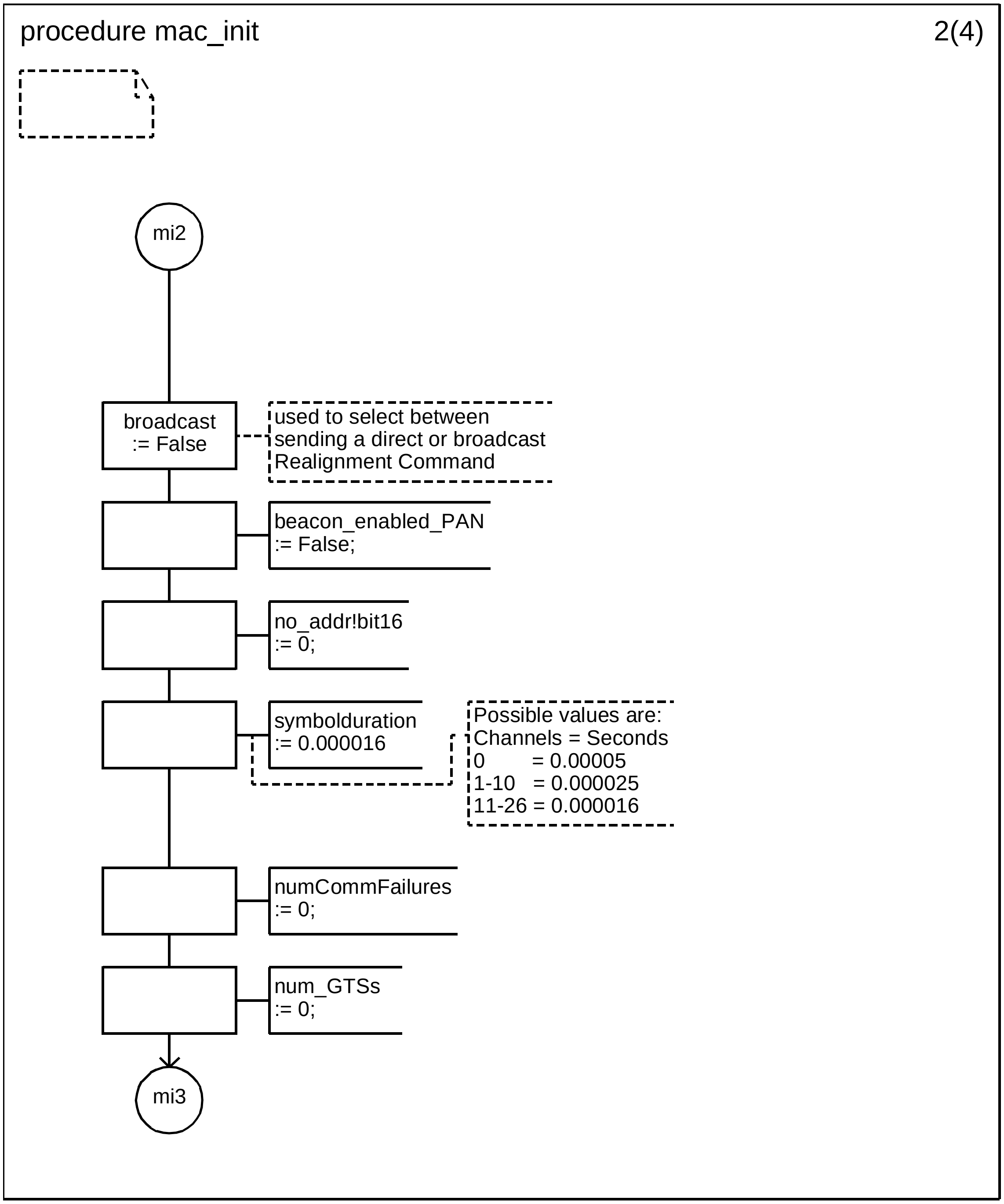


D.3.1.154.1 Procedure mac_init (1)

.

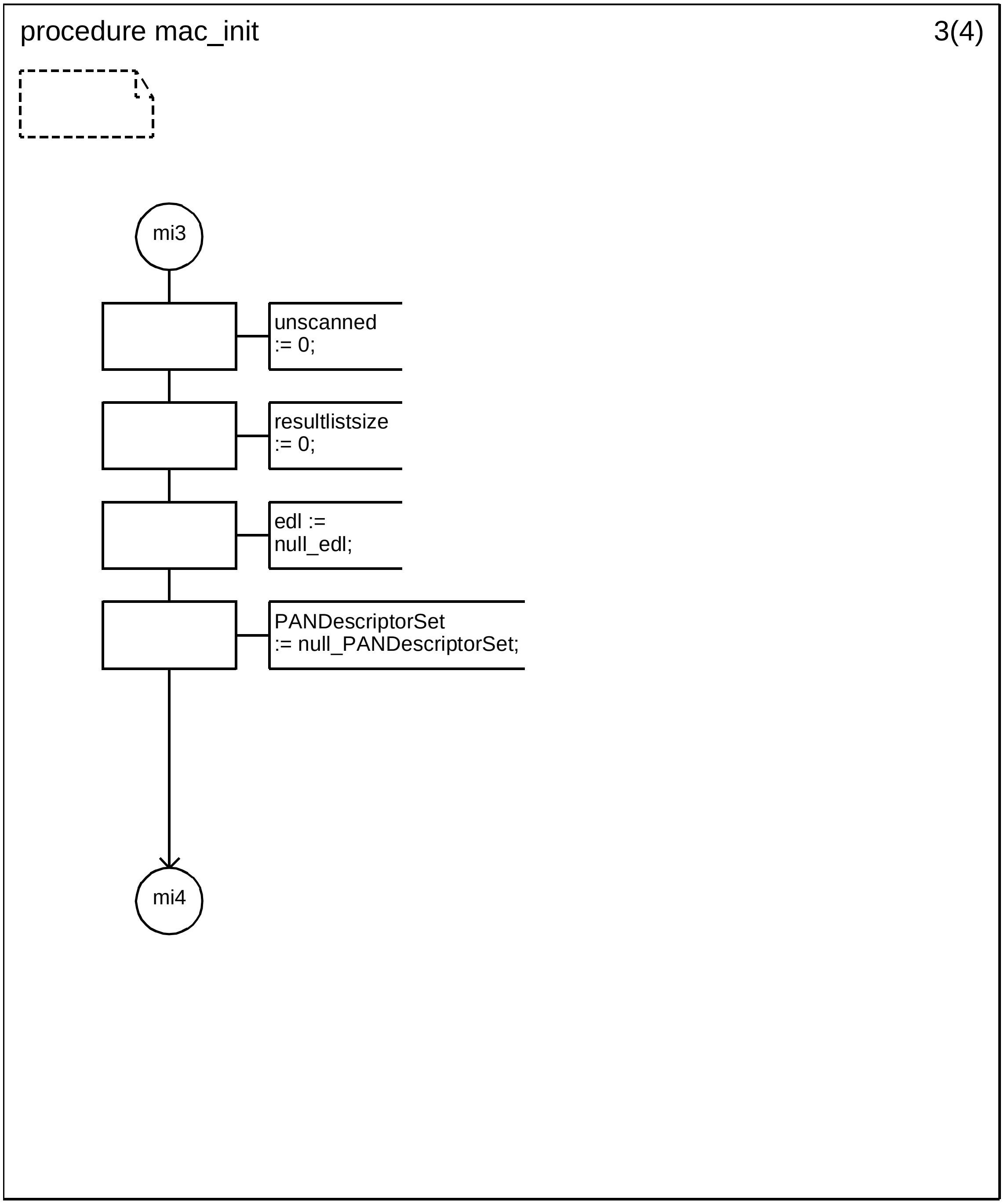


D.3.1.154.2 Procedure mac_init (2)



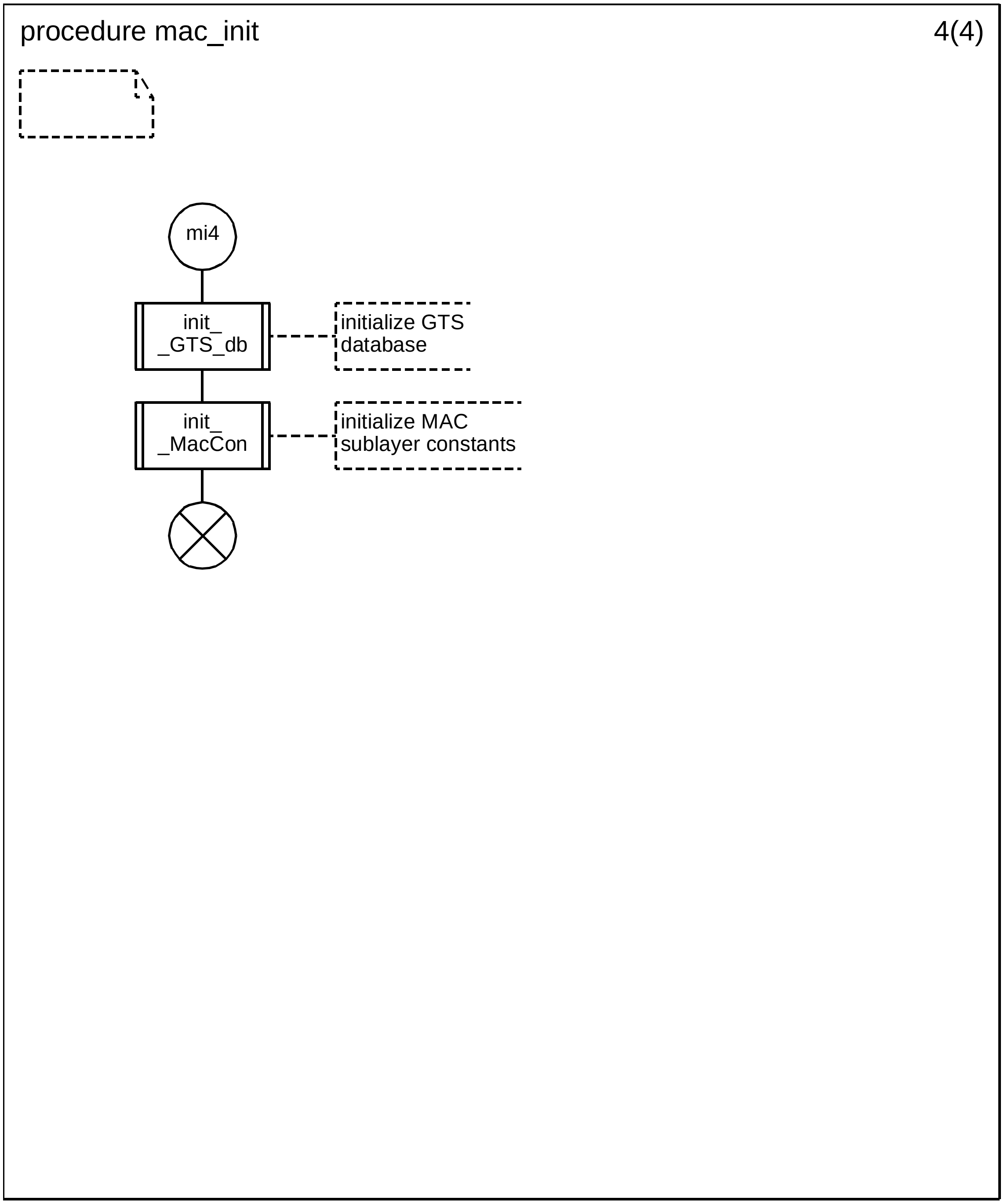
D.3.1.154.3 Procedure mac_init (3)

.



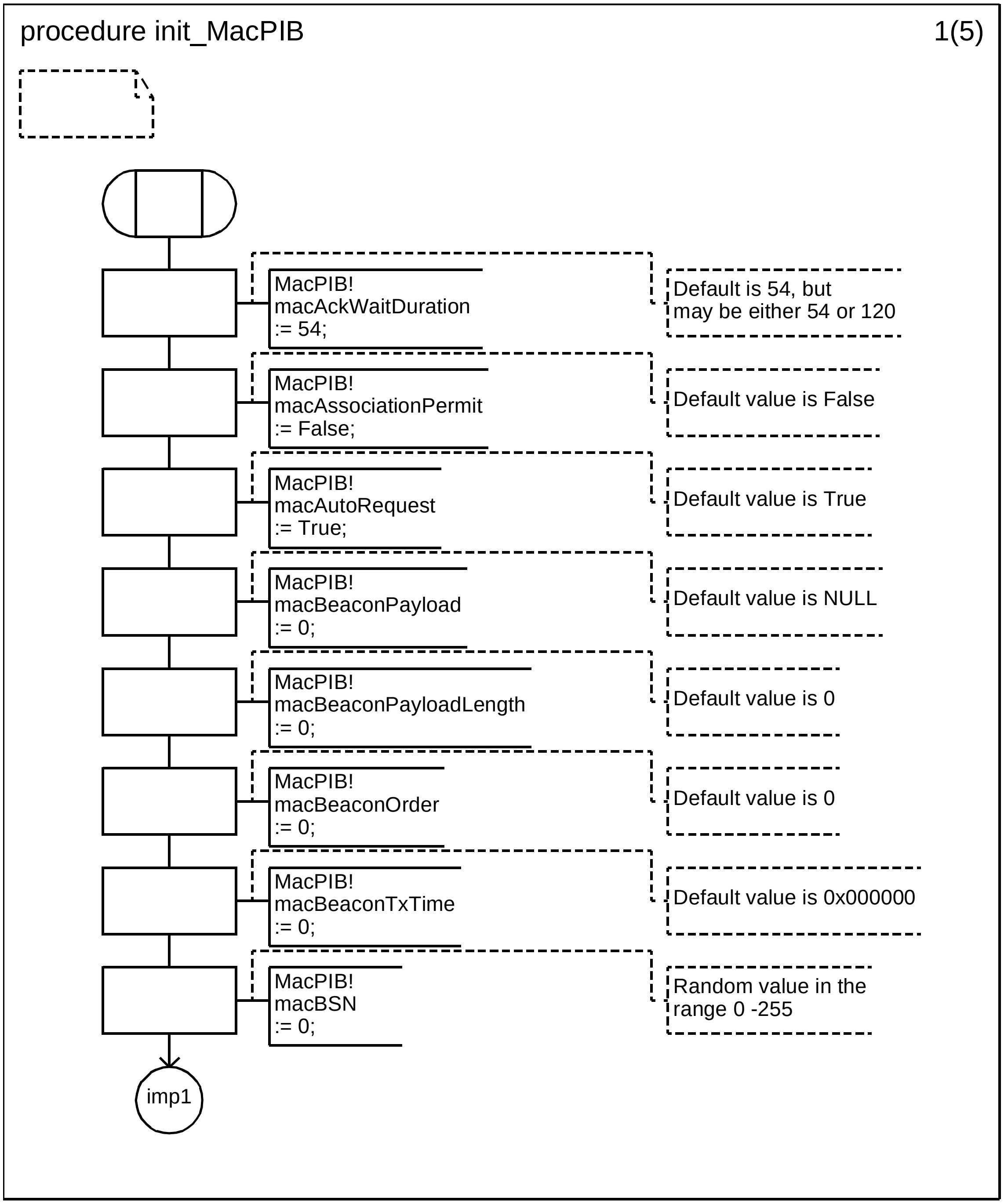
D.3.1.154.4 Procedure mac_init (4)

.



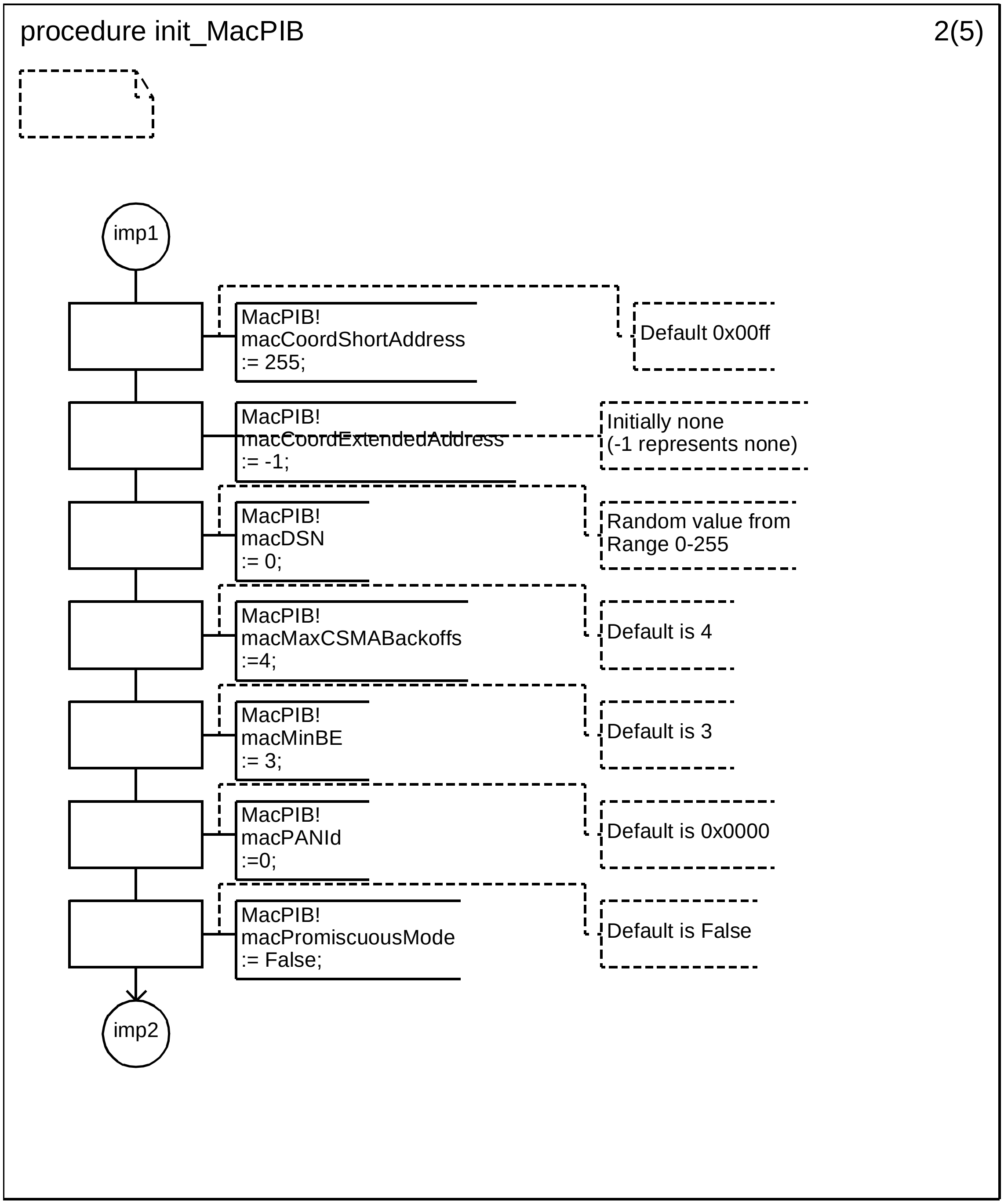
D.3.1.154.5 Procedure init_MacPIB (1)

.


procedure init_MacPIB
1(5)
MacPIB! macAckWaitDuration := 54;
Default is 54, but may be either 54 or 120
MacPIB! macAssociationPermit := False;
Default value is False
MacPIB! macAutoRequest := True;
Default value is True
MacPIB! macBeaconPayload := 0;
Default value is NULL
MacPIB! macBeaconPayloadLength := 0;
Default value is 0
MacPIB! macBeaconOrder := 0;
Default value is 0
MacPIB! macBeaconTxTime := 0;
Default value is 0x000000
MacPIB! macBSN := 0;
Random value in the range 0 -255
imp1

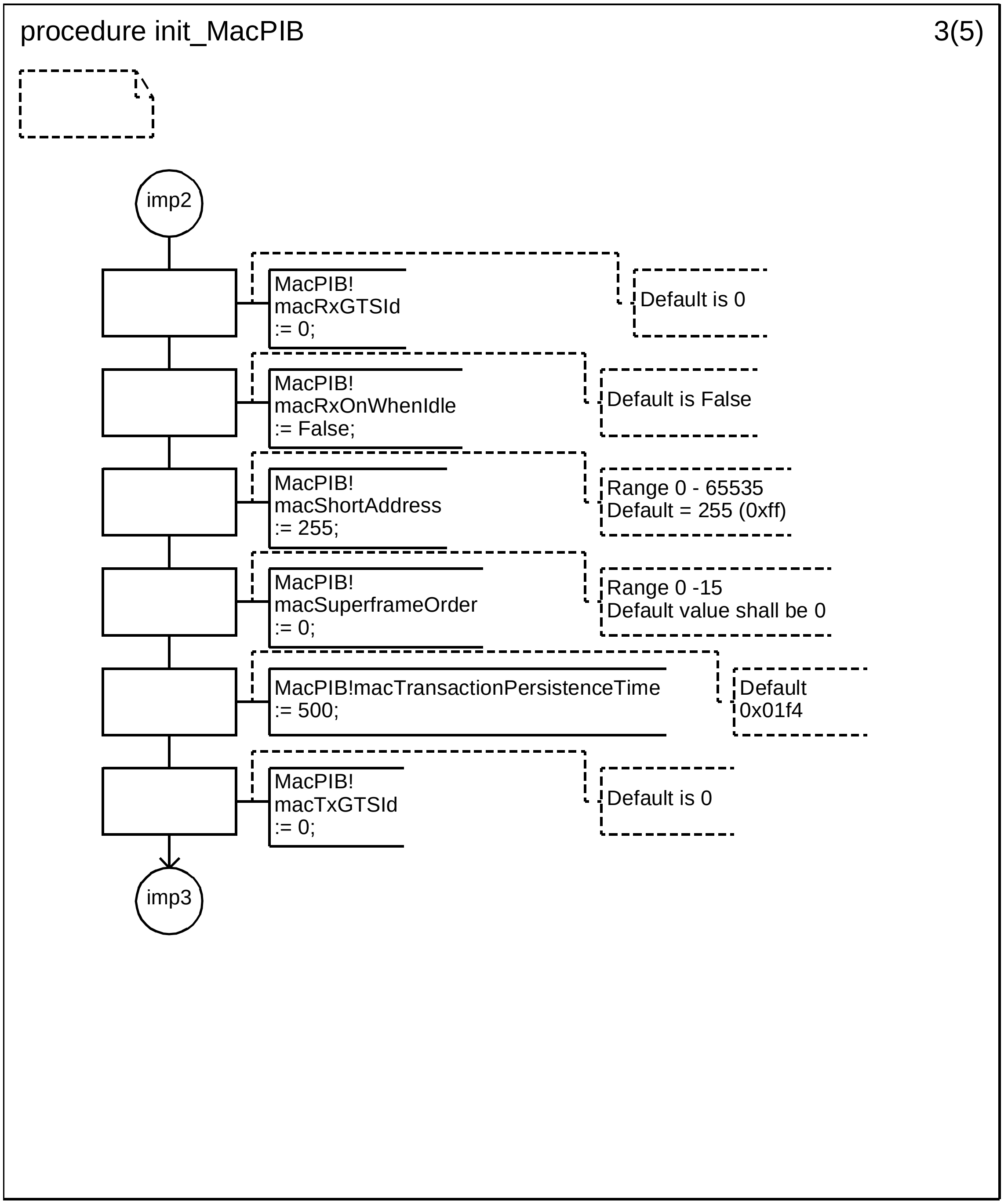
D.3.1.154.6 Procedure init_MacPIB (2)

.



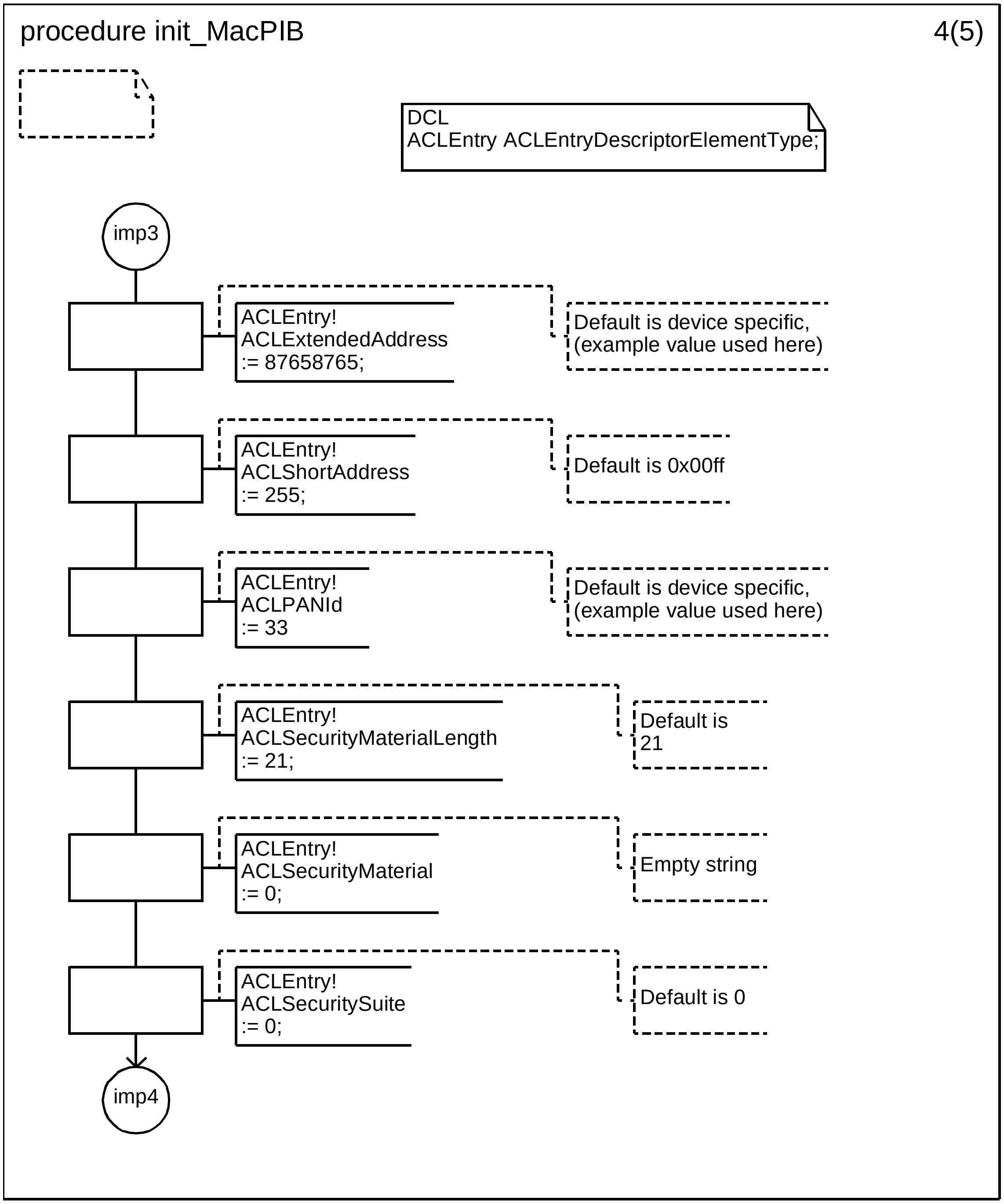
D.3.1.154.7 Procedure init_MacPIB (3)

.



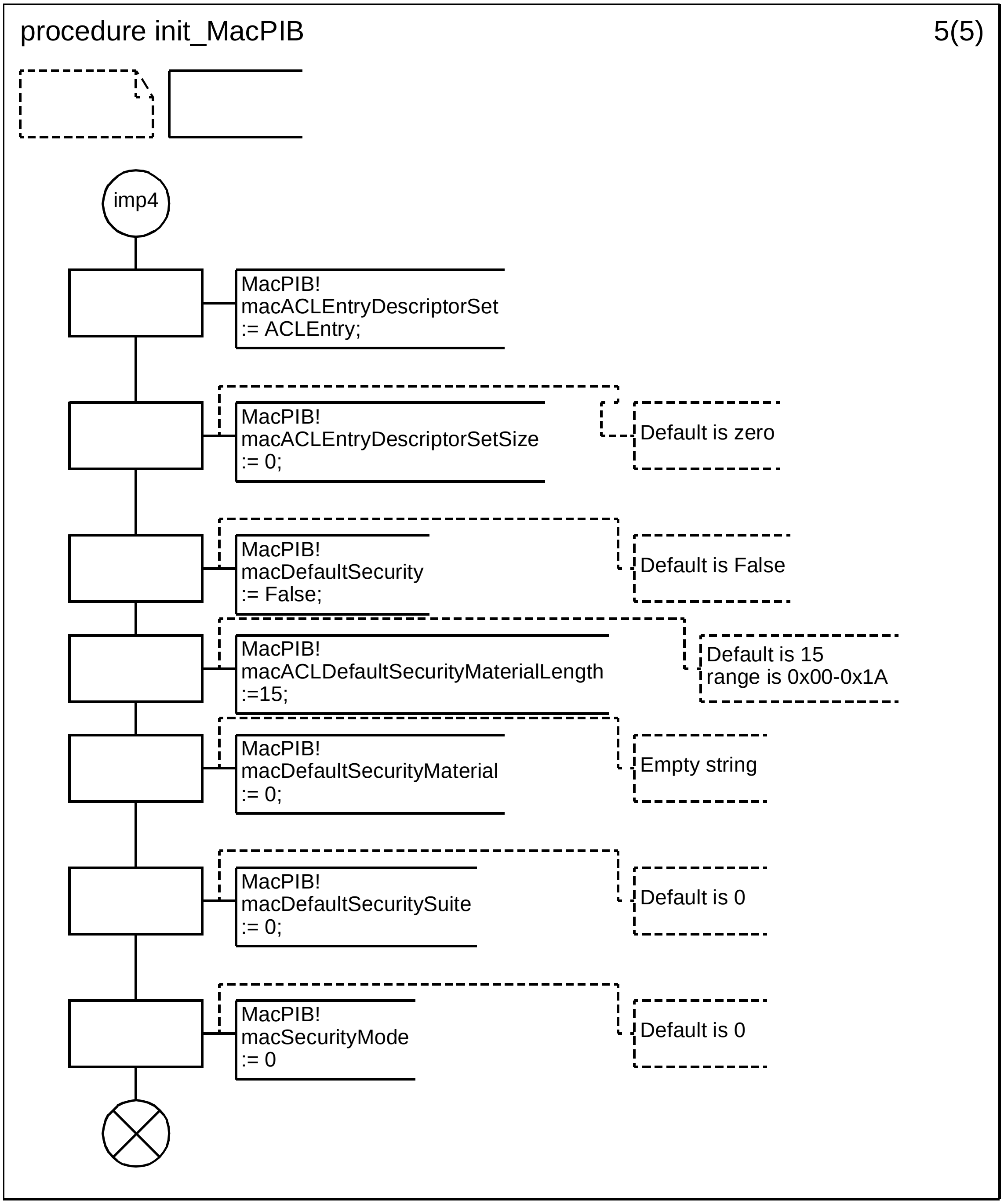
D.3.1.154.8 Procedure init_MacPIB (4)

.



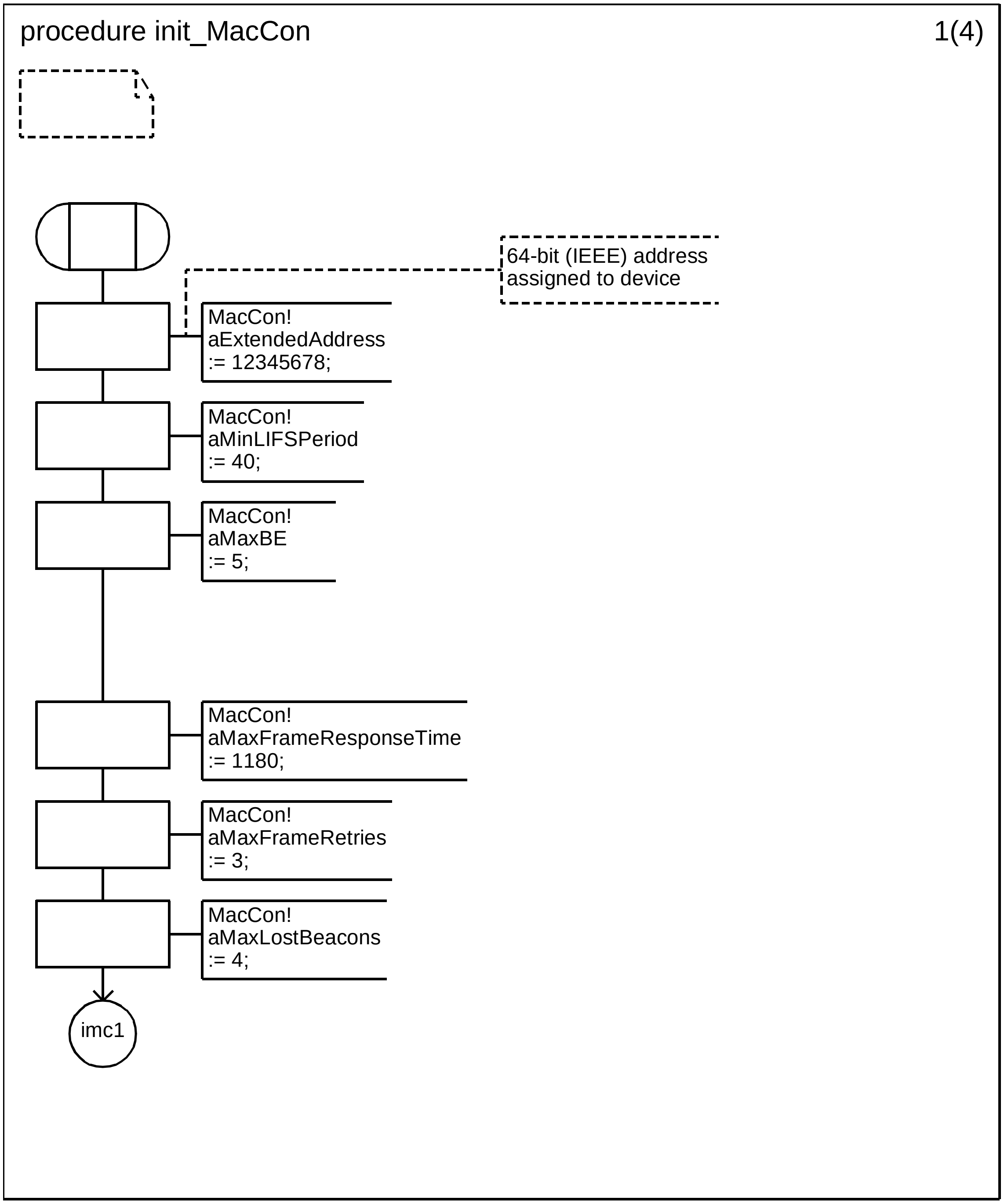
D.3.1.154.9 Procedure init_MacPIB (5)

.



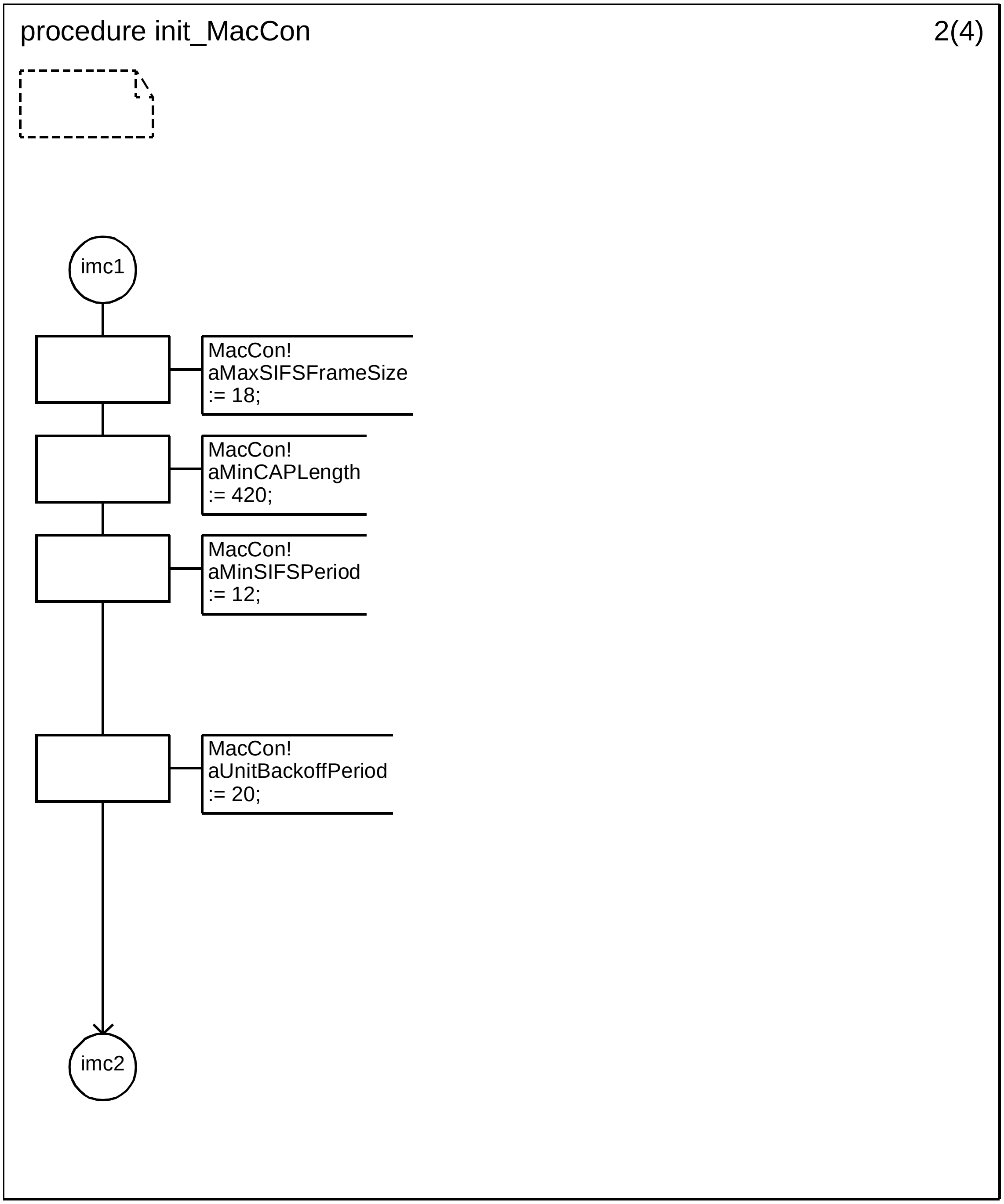
D.3.1.154.10 Procedure init_MACCon (1)

.



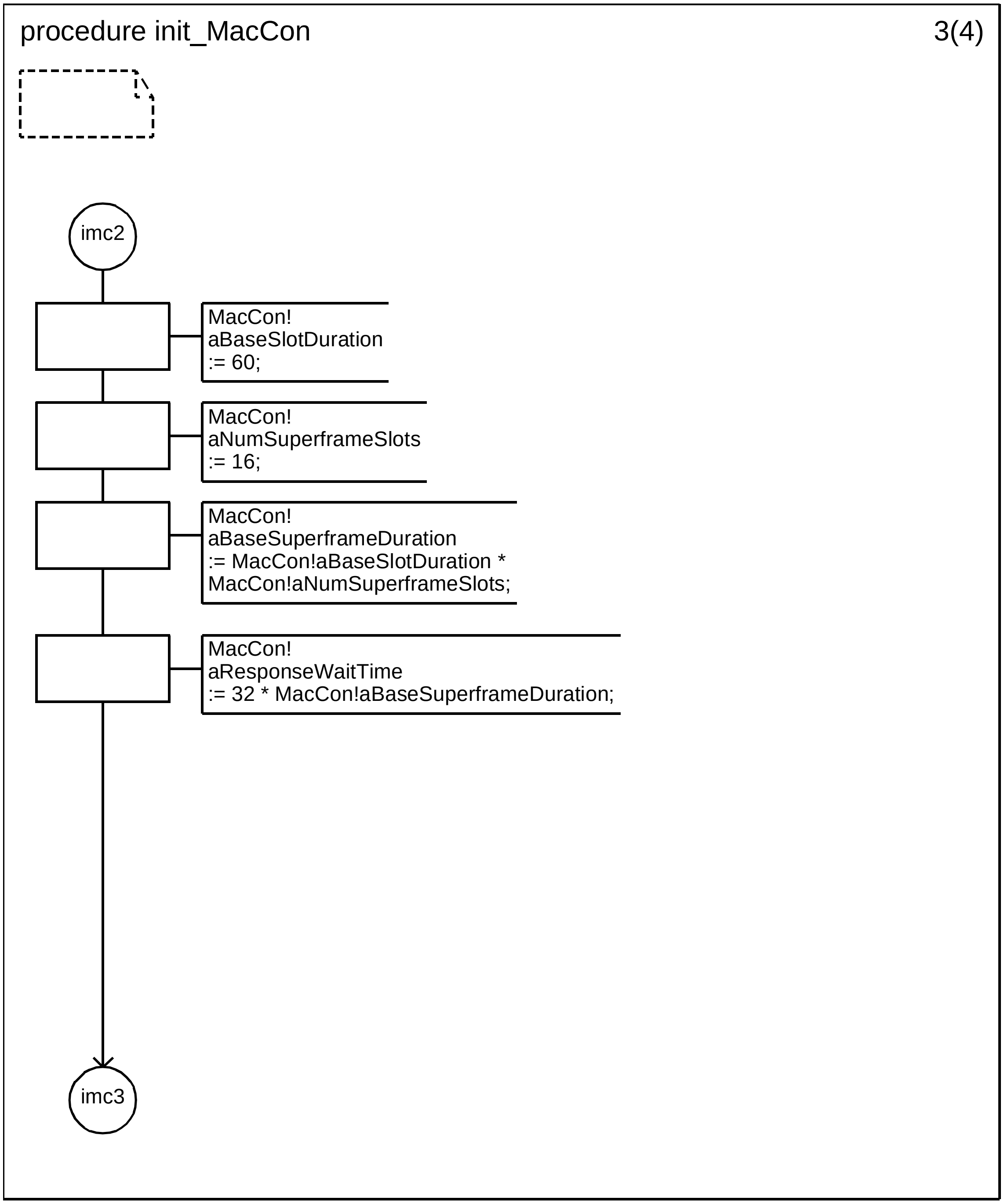
D.3.1.154.11 Procedure init_MACCon (2)

.



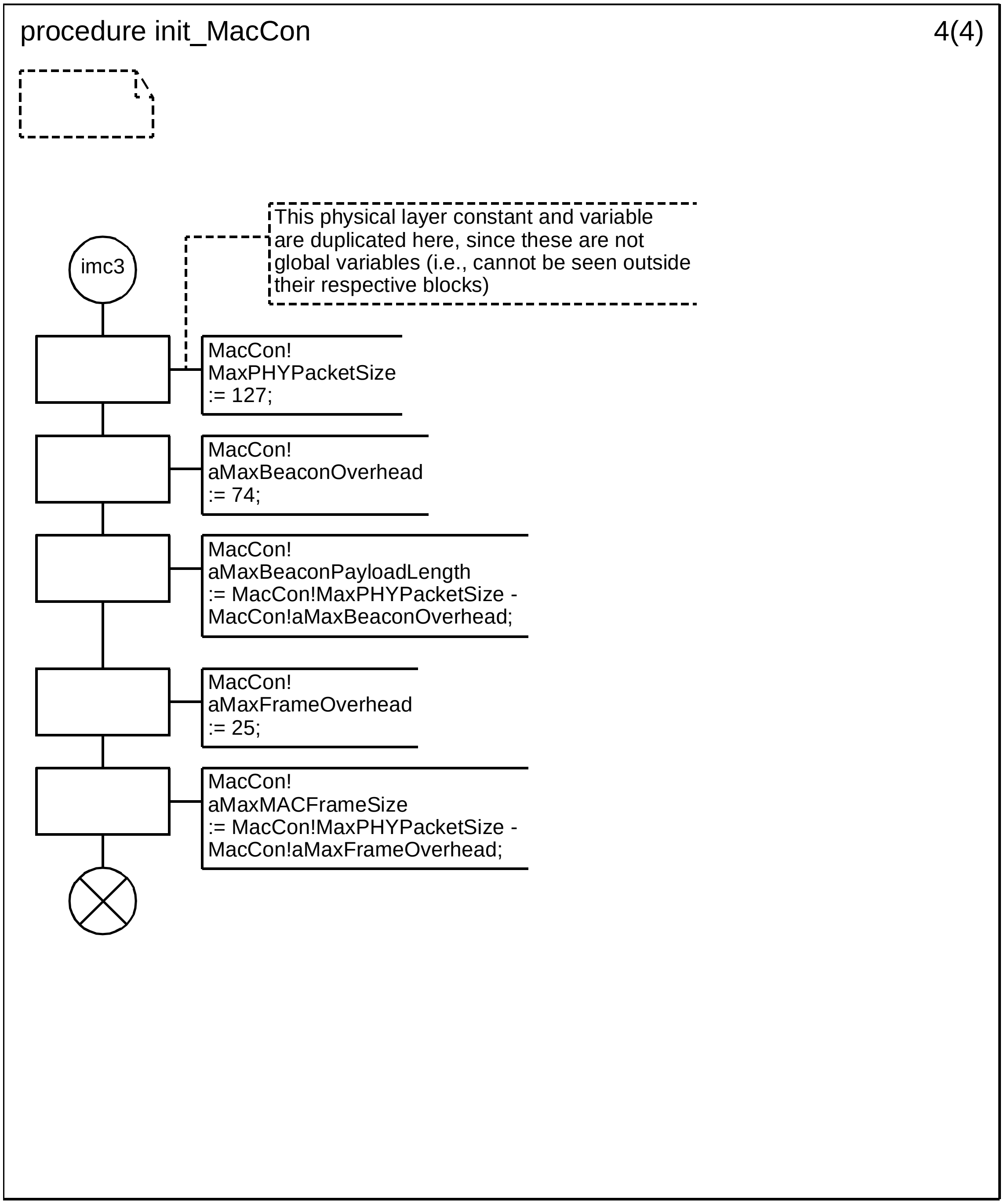
D.3.1.154.12 Procedure init_MACCon (3)

.



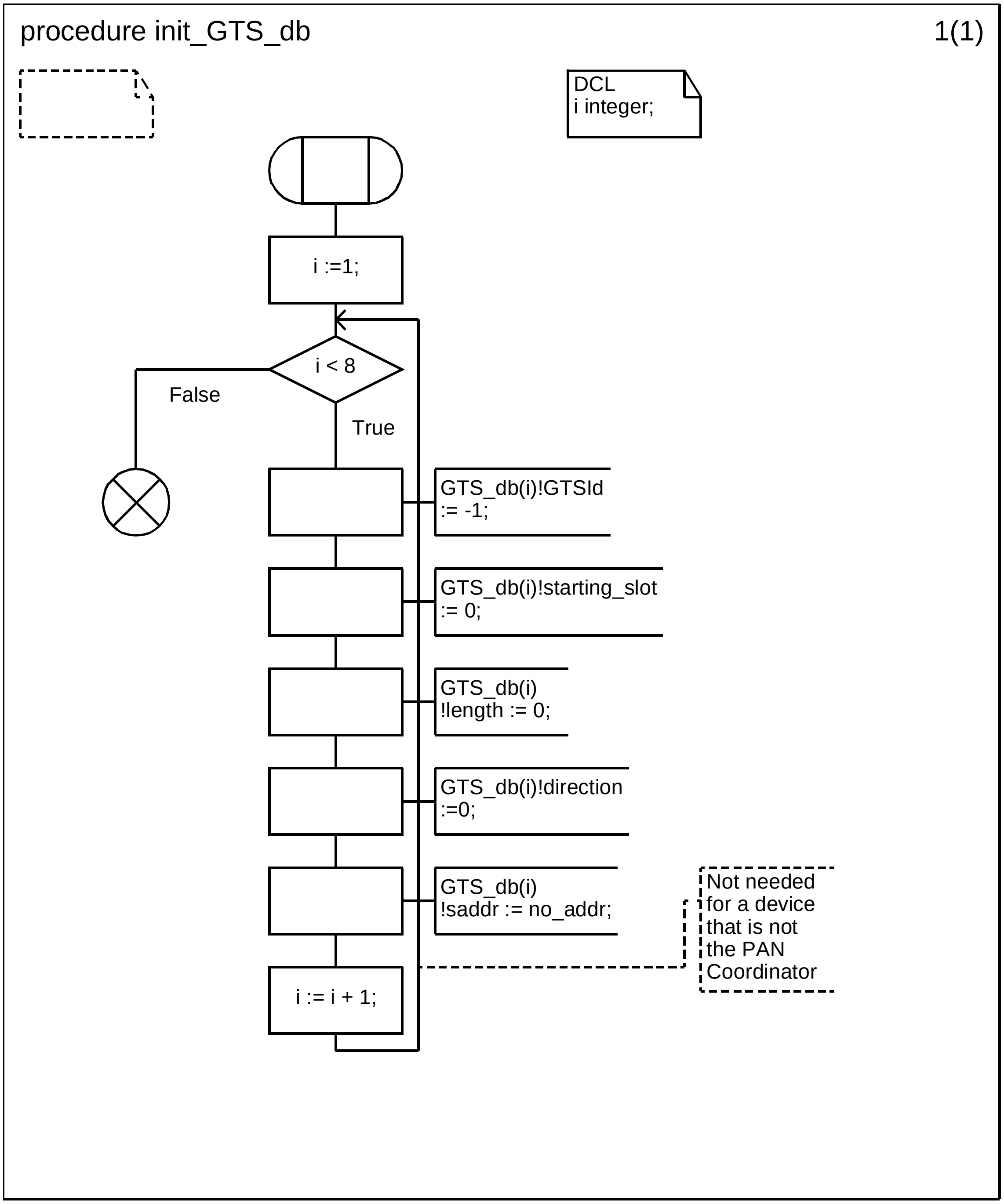
D.3.1.154.13 Procedure init_MACCon (4)

.


procedure init_MacCon
4(4)
This physical layer constant and variable are duplicated here, since these are not global variables (i.e., cannot be seen outside their respective blocks)
imc3
MacCon!
MaxPHYPacketSize
:= 127;
MacCon!
aMaxBeaconOverhead
:= 74;
MacCon!
aMaxBeaconPayloadLength
:= MacCon!MaxPHYPacketSize -
MacCon!aMaxBeaconOverhead;
MacCon!
aMaxFrameOverhead
:= 25;
MacCon!
aMaxMACFrameSize
:= MacCon!MaxPHYPacketSize -
MacCon!aMaxFrameOverhead;

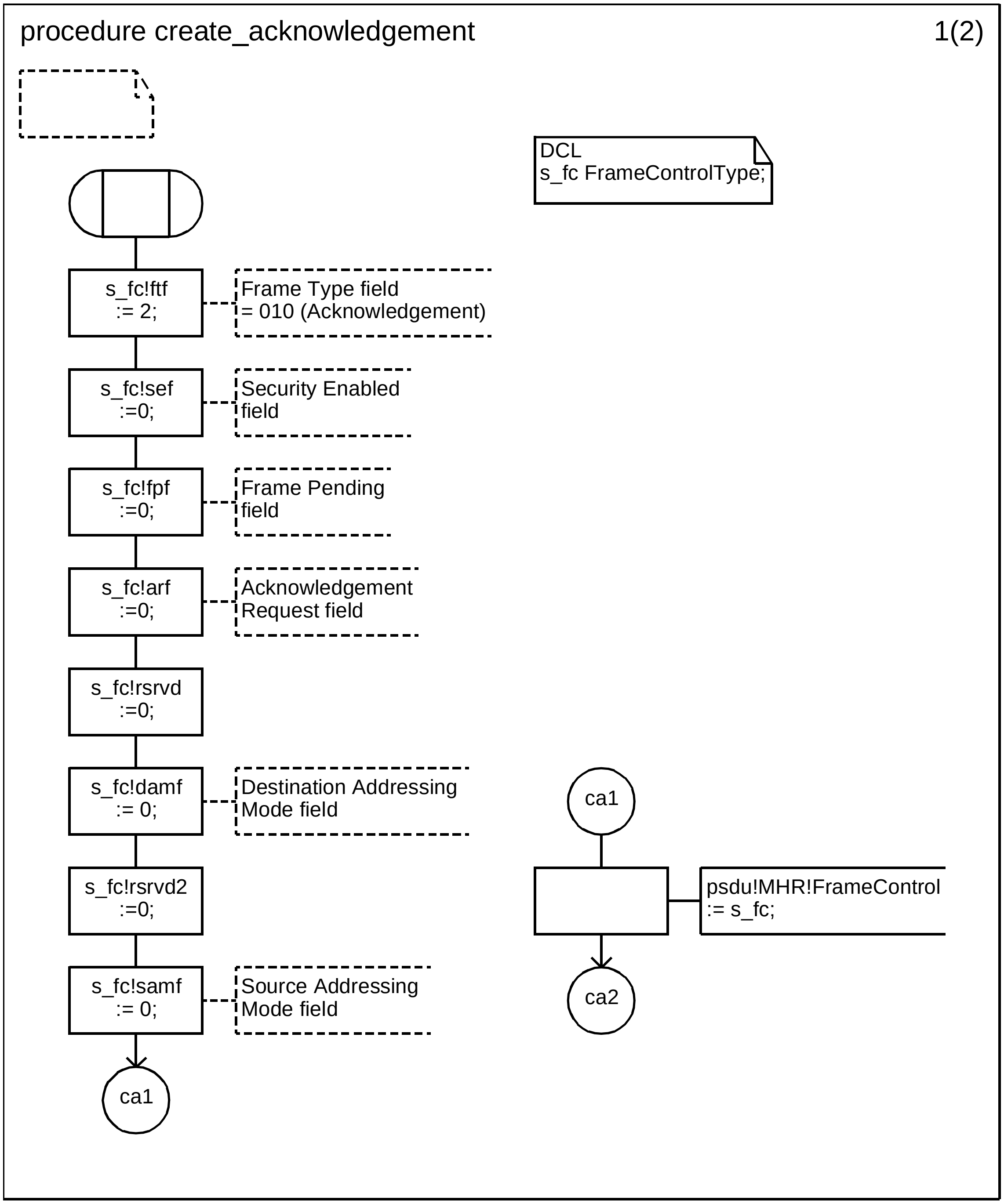
D.3.1.154.14 Procedure init_GTS_db

.



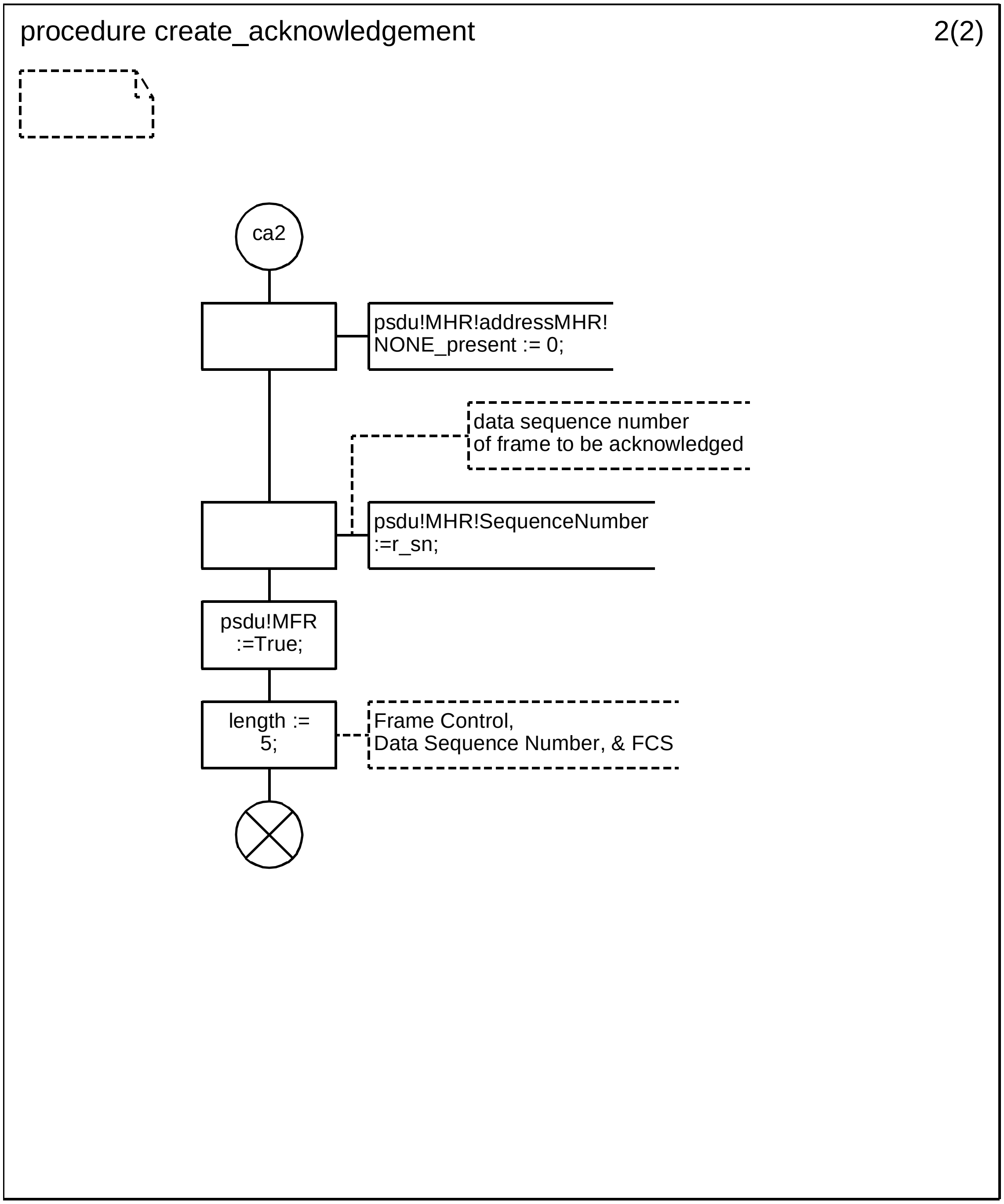
D.3.1.154.15 Procedure create_Acknowledgement (1)

.



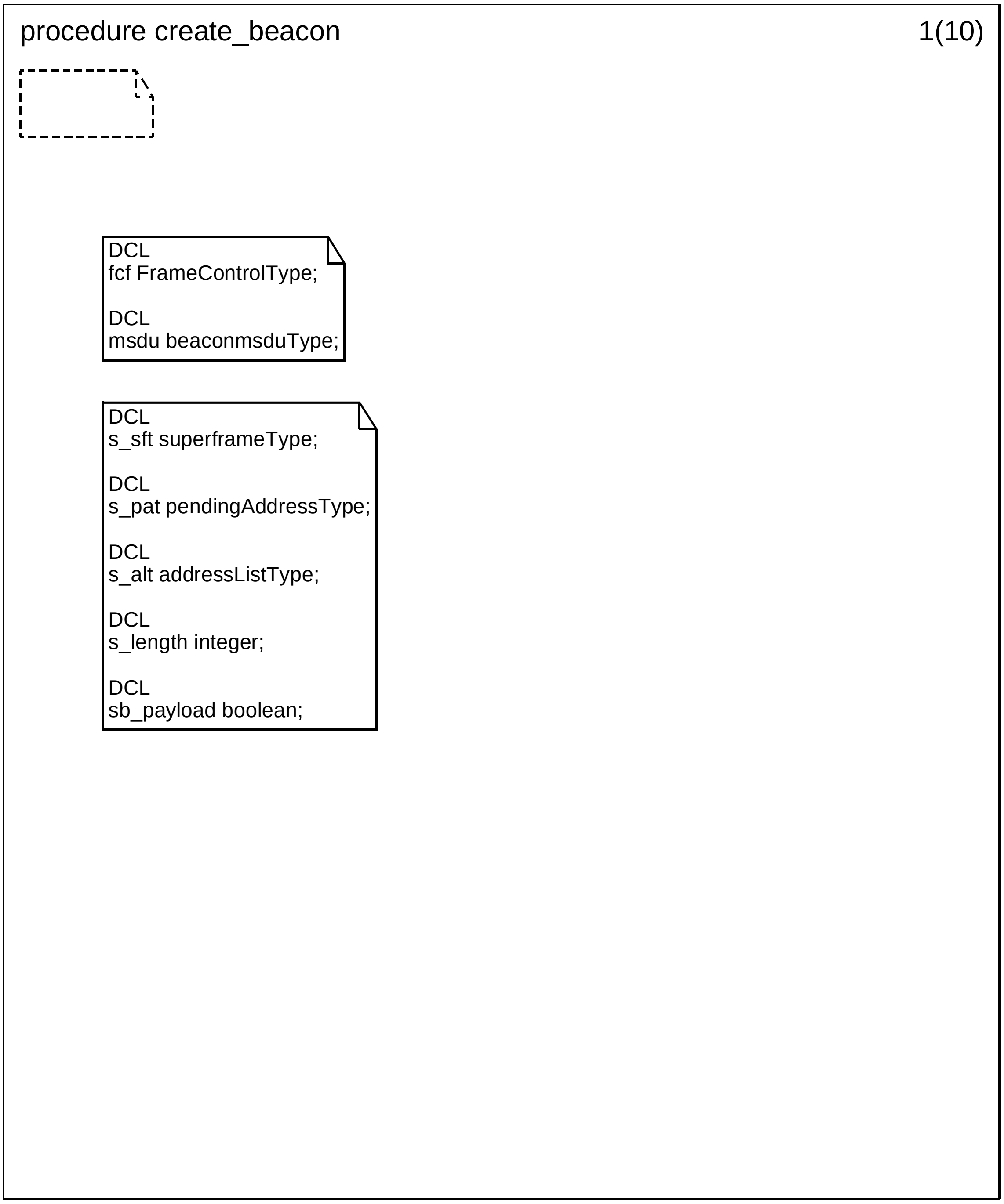
D.3.1.154.16 Procedure create_Acknowledgement (2)

.



D.3.1.154.17 Procedure create_beacon (1)

.

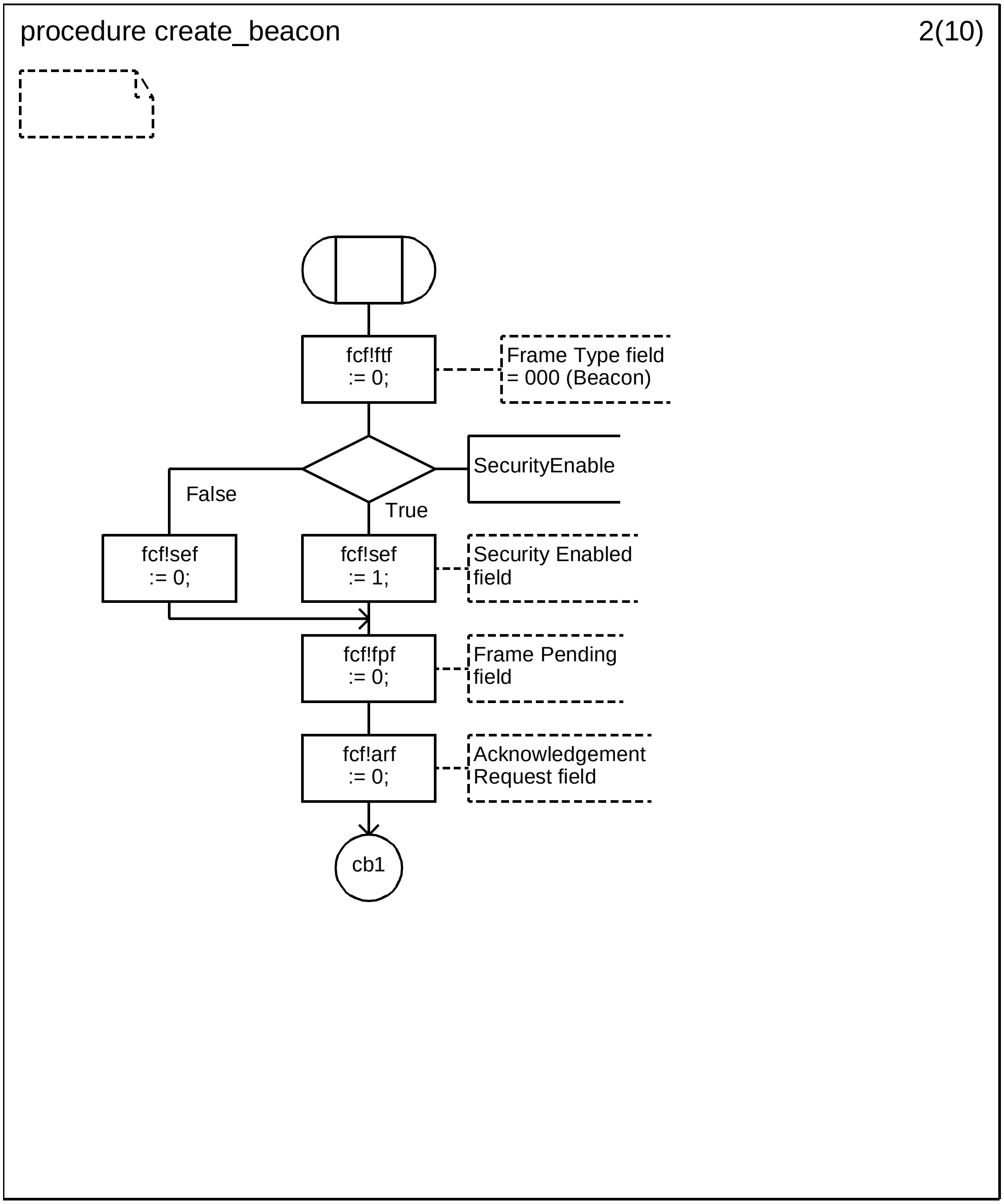


```
DCL
fcf FrameControlType;

DCL
msdu beaconmsduType;
```

```
DCL
s_sft superframeType;

DCL
s_pat pendingAddressType;

DCL
s_alt addressListType;

DCL
s_length integer;

DCL
sb_payload boolean;
```

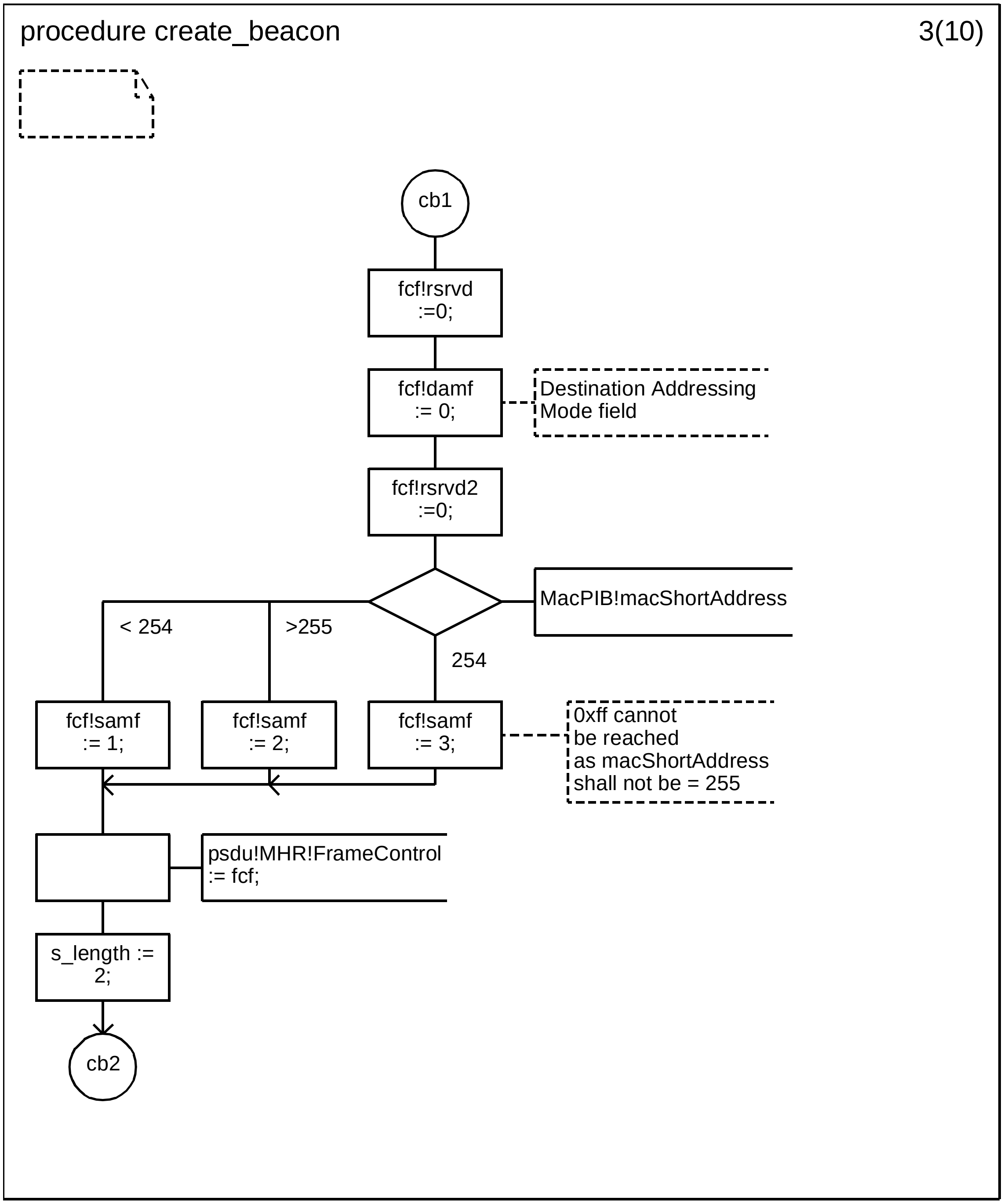
D.3.1.154.18 Procedure create_beacon (2)

.



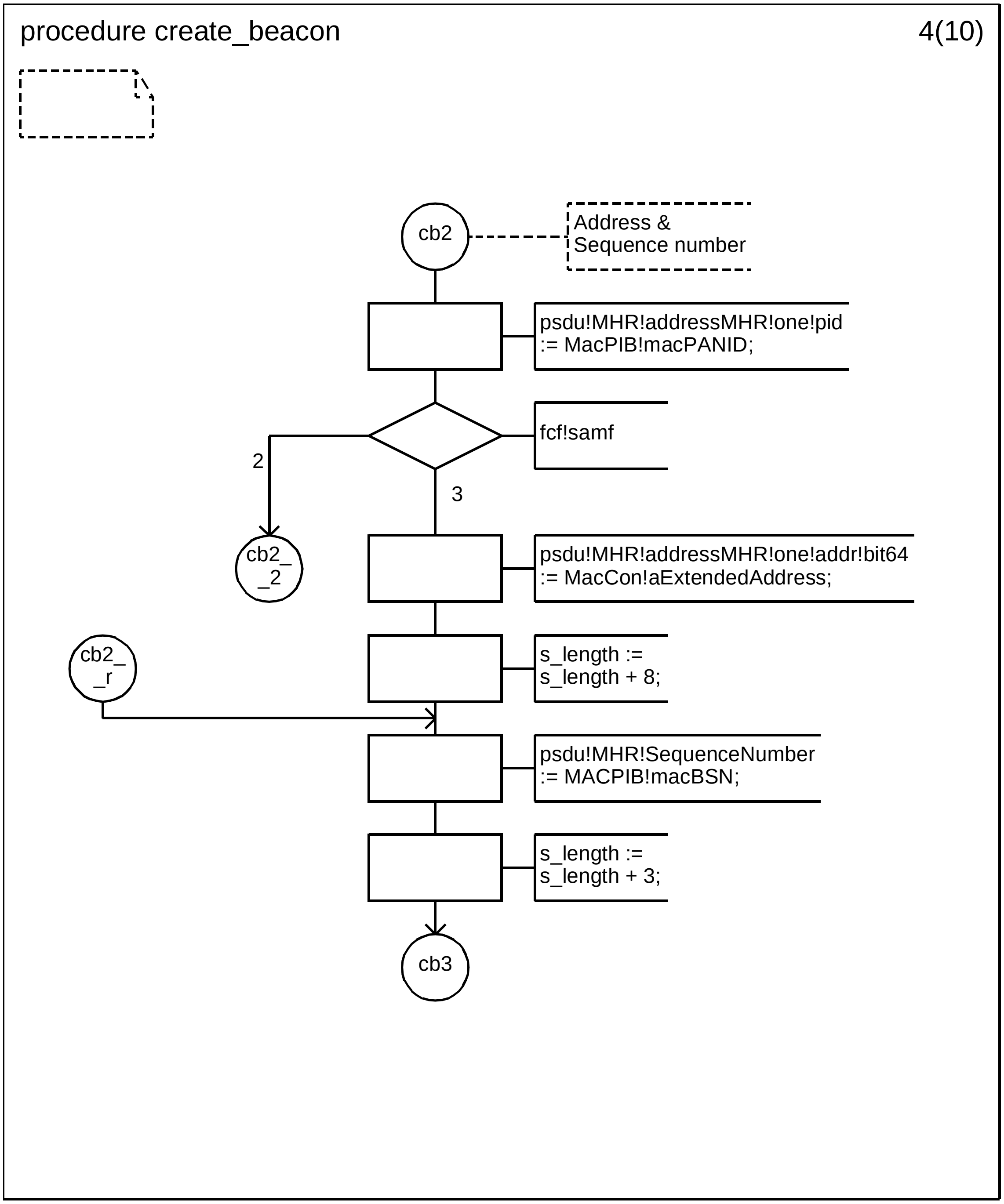
D.3.1.154.19 Procedure create_beacon (3)

.



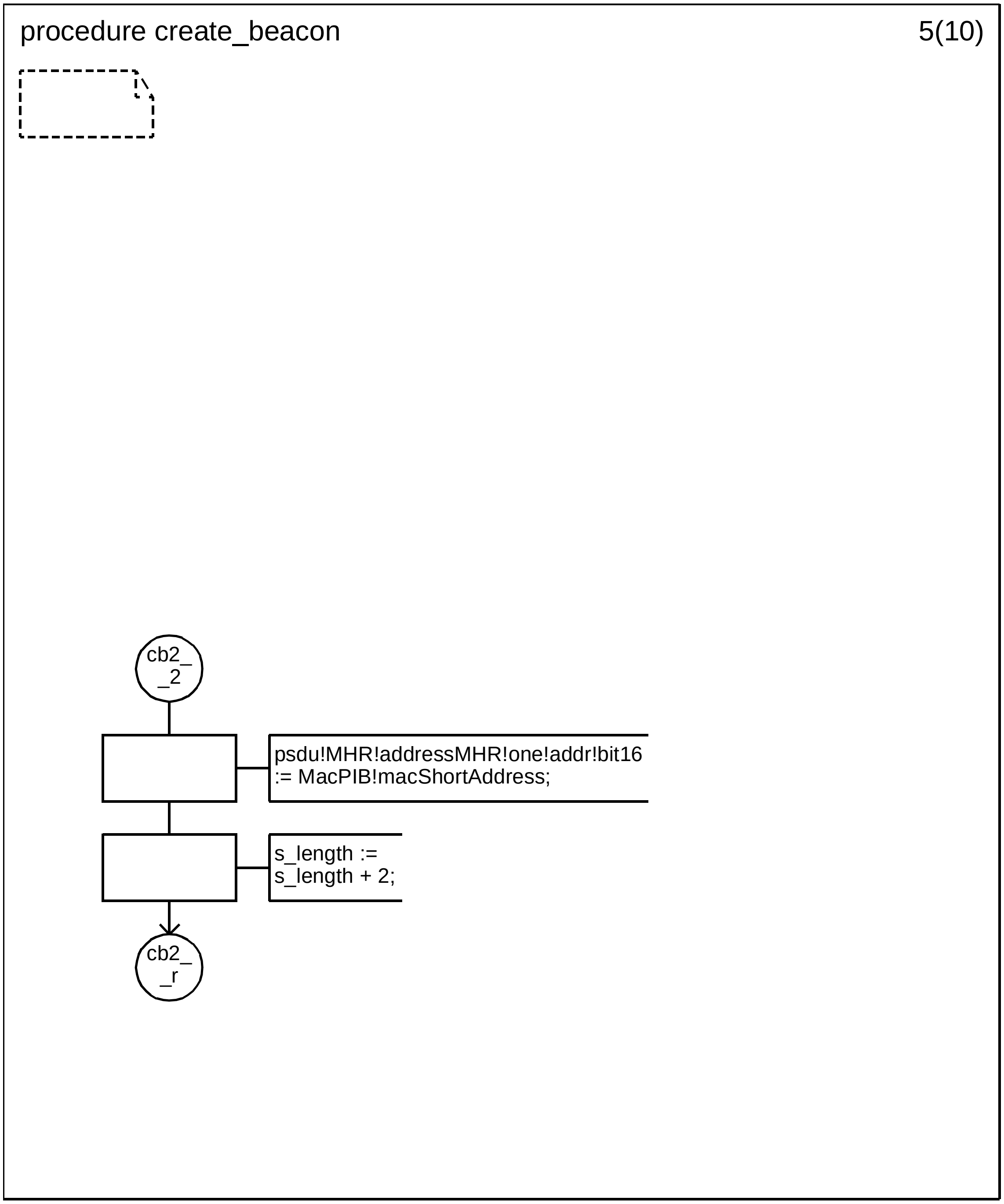
D.3.1.154.20 Procedure create_beacon (4)

.



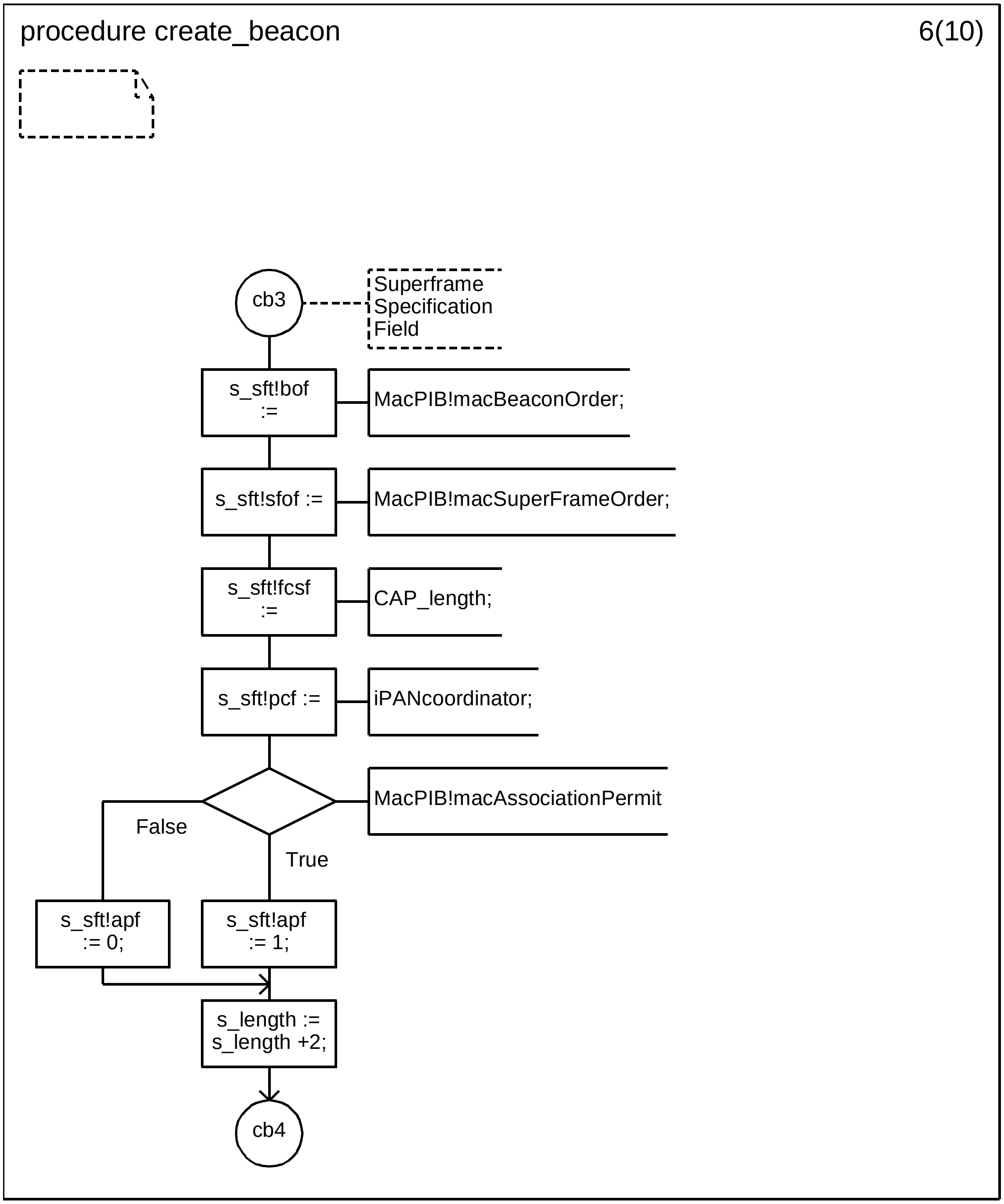
D.3.1.154.21 Procedure create_beacon (5)

.

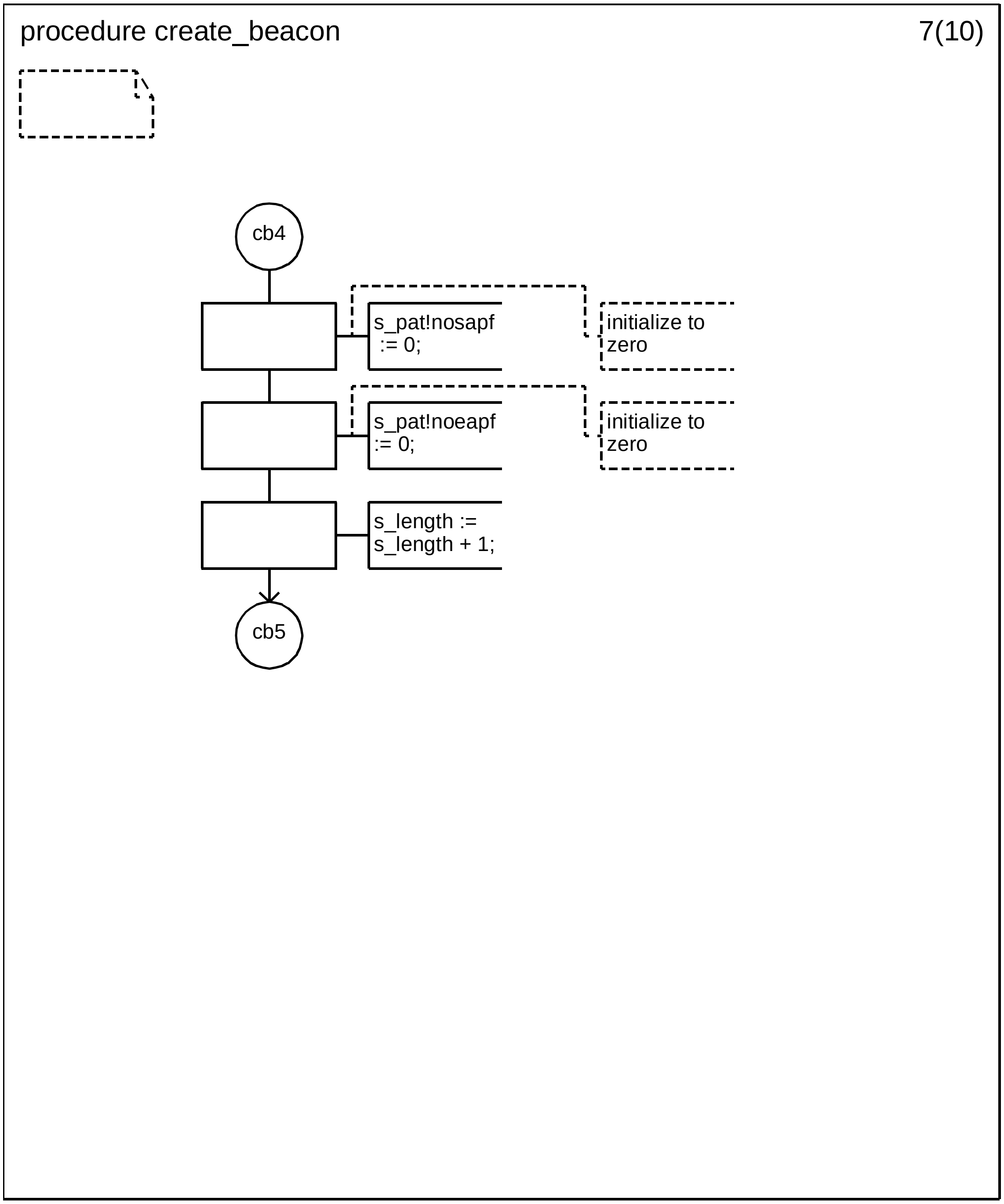


D.3.1.154.22 Procedure create_beacon (6)

.

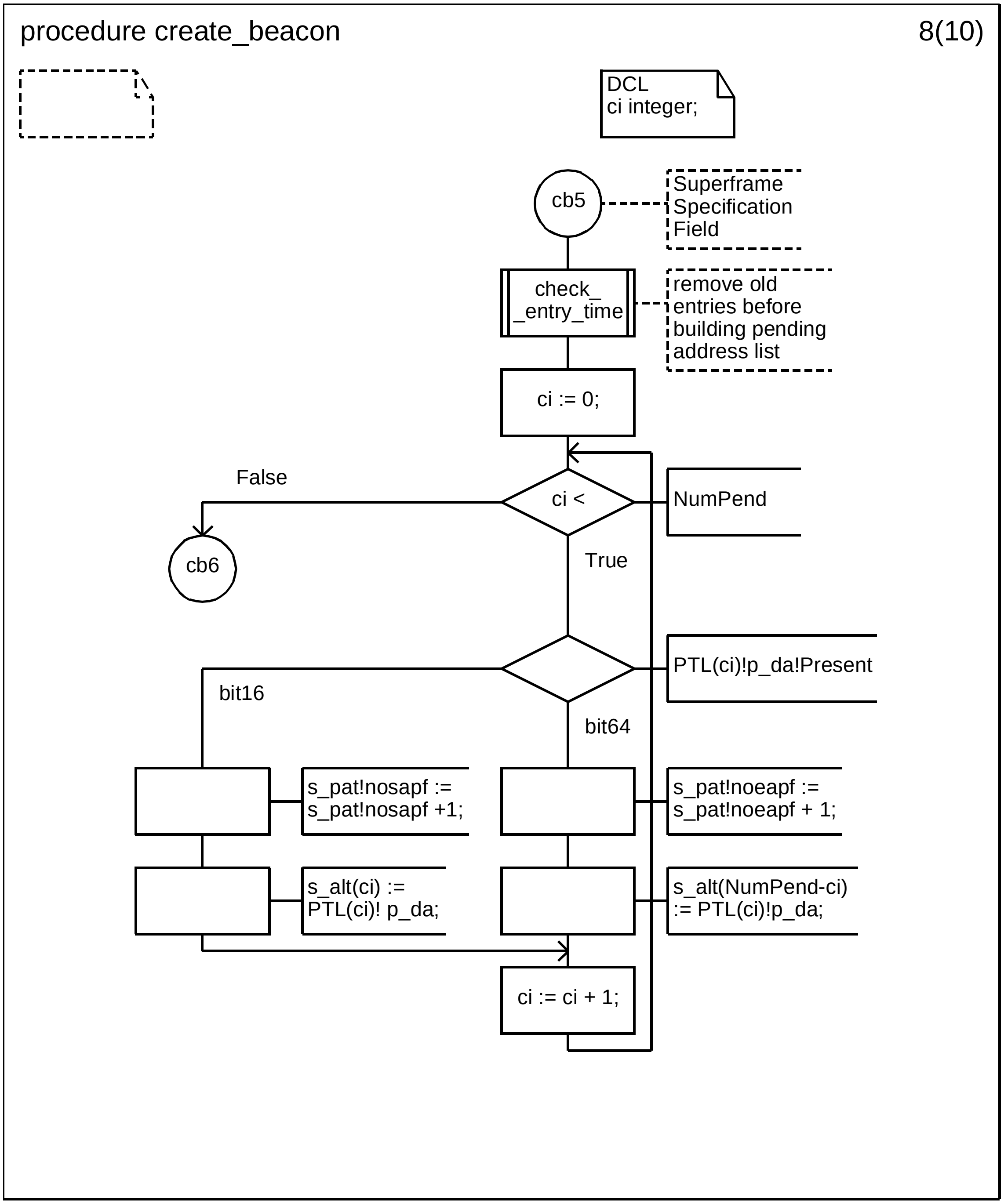


D.3.1.154.23 Procedure create_beacon (7)

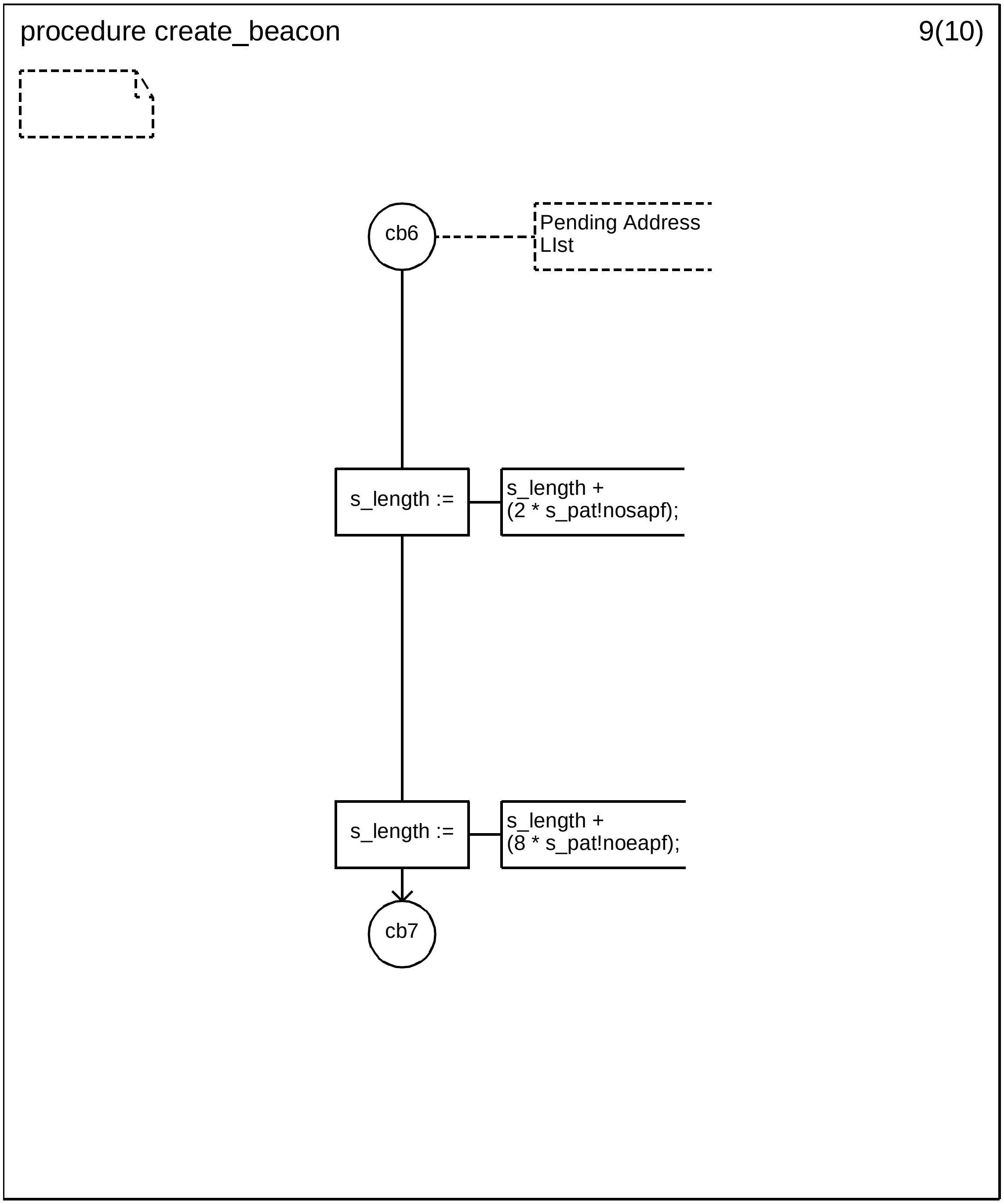


D.3.1.154.24 Procedure create_beacon (8)

.

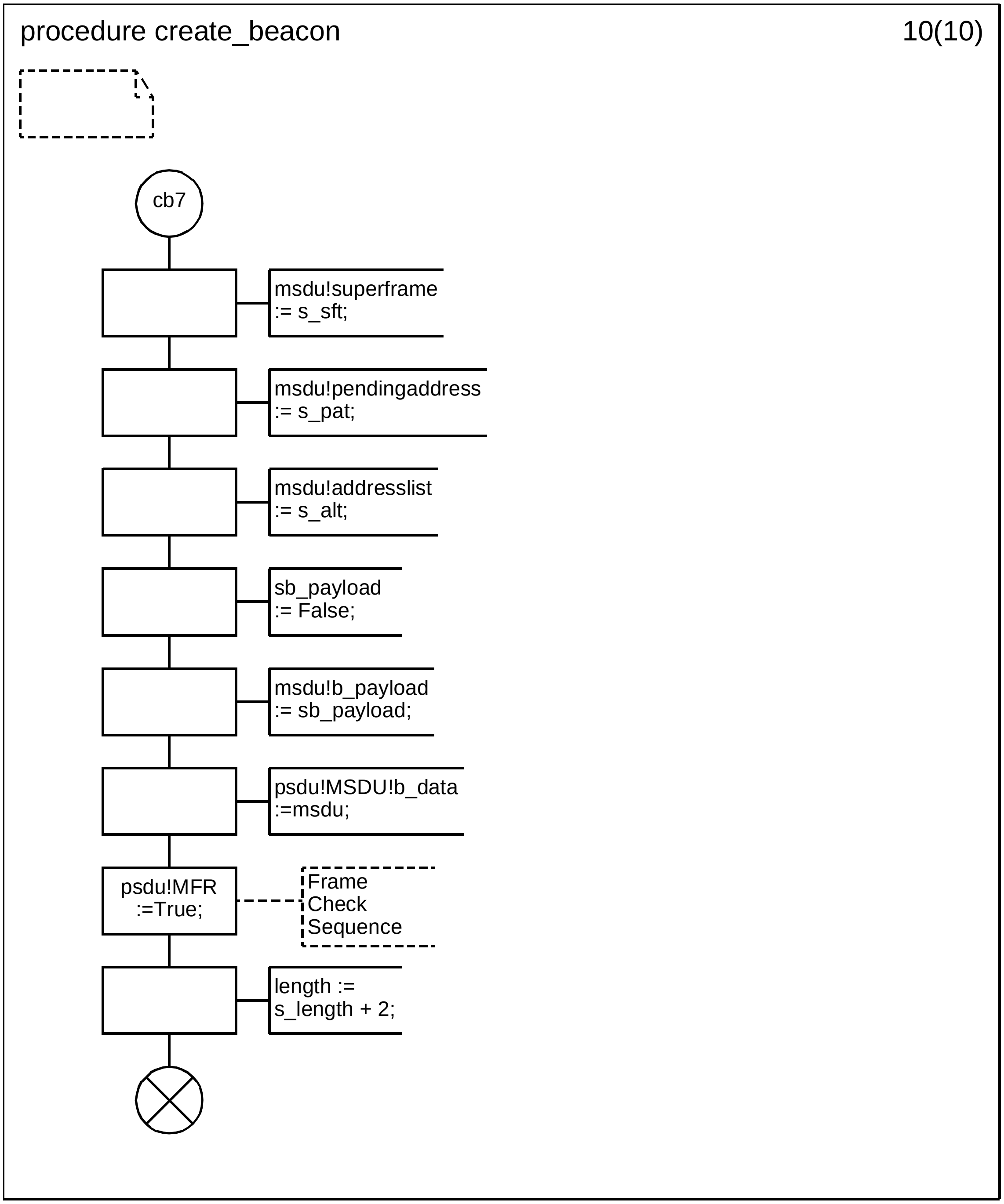


D.3.1.154.25 Procedure create_beacon (9)



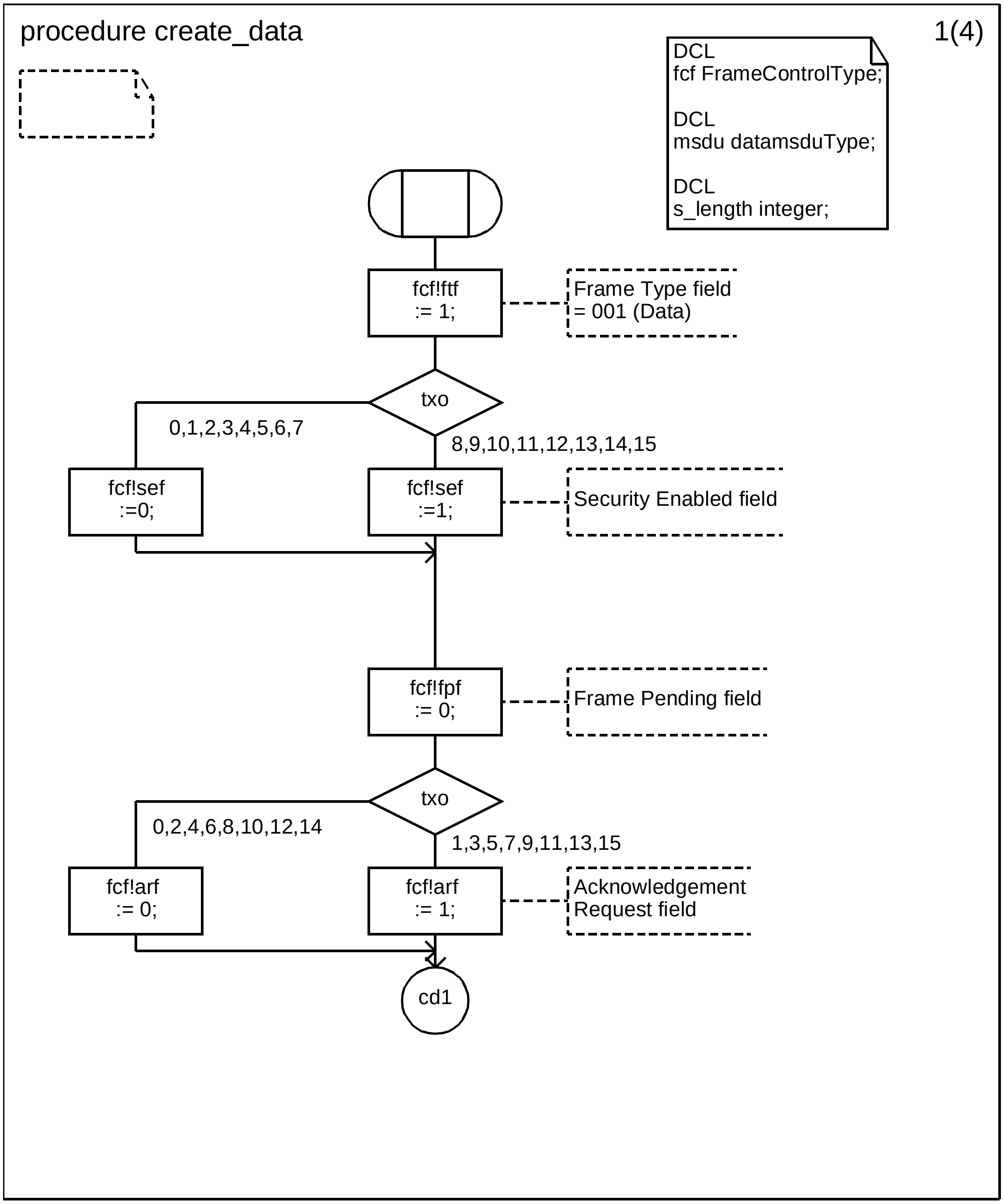
D.3.1.154.26 Procedure create_beacon (10)

.



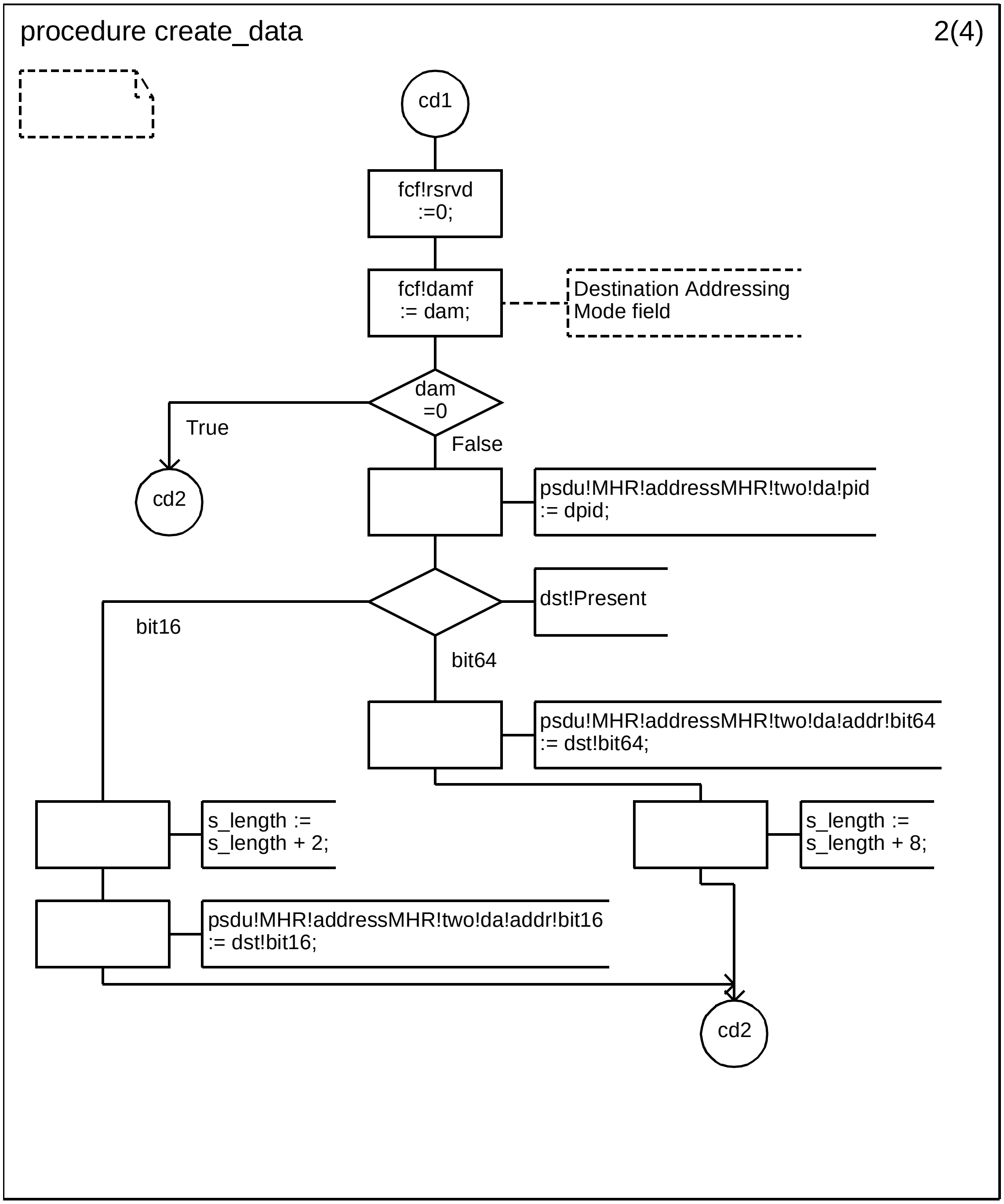
D.3.1.154.27 Procedure create_data (1)

.



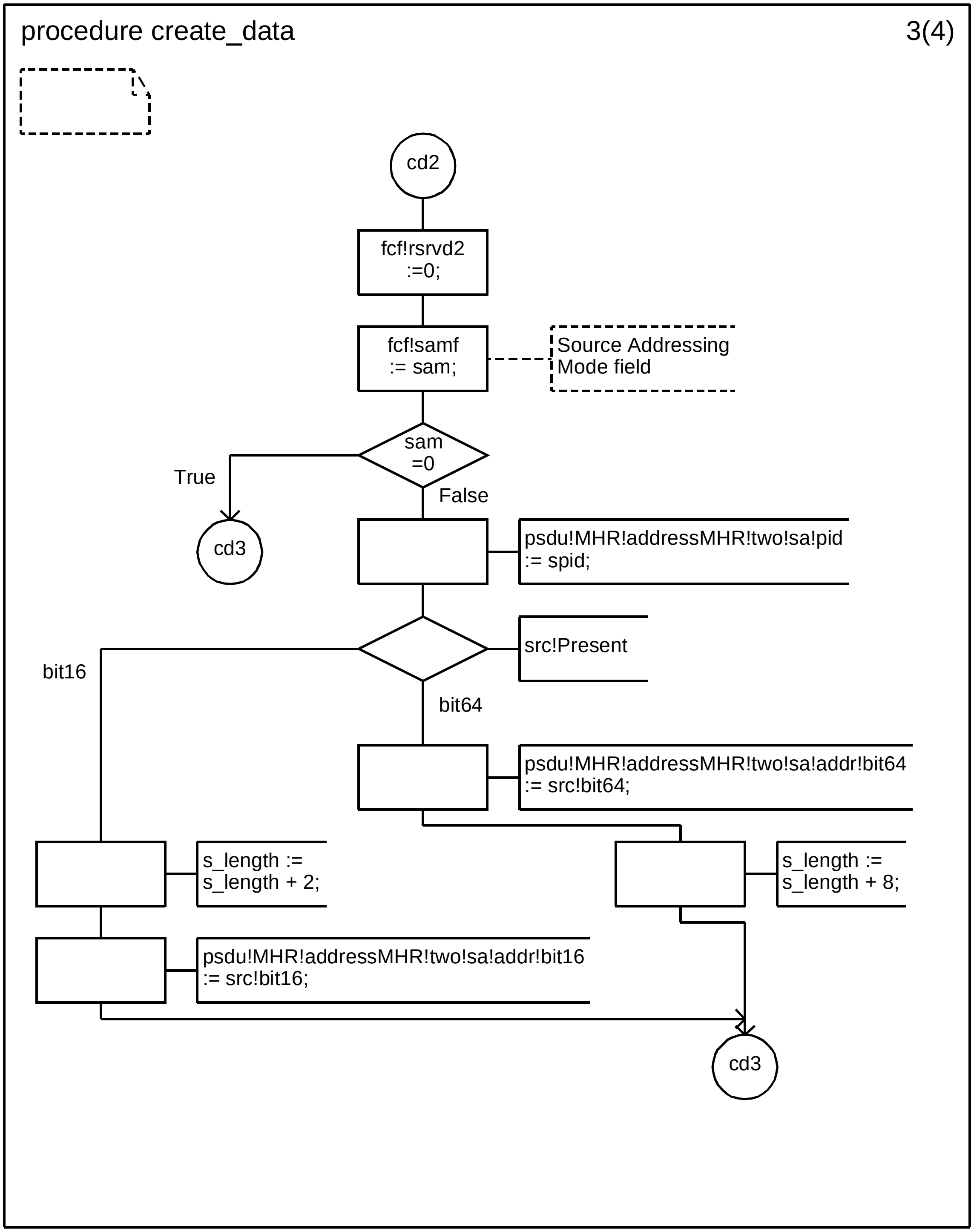
D.3.1.154.28 Procedure create_data (2)

.



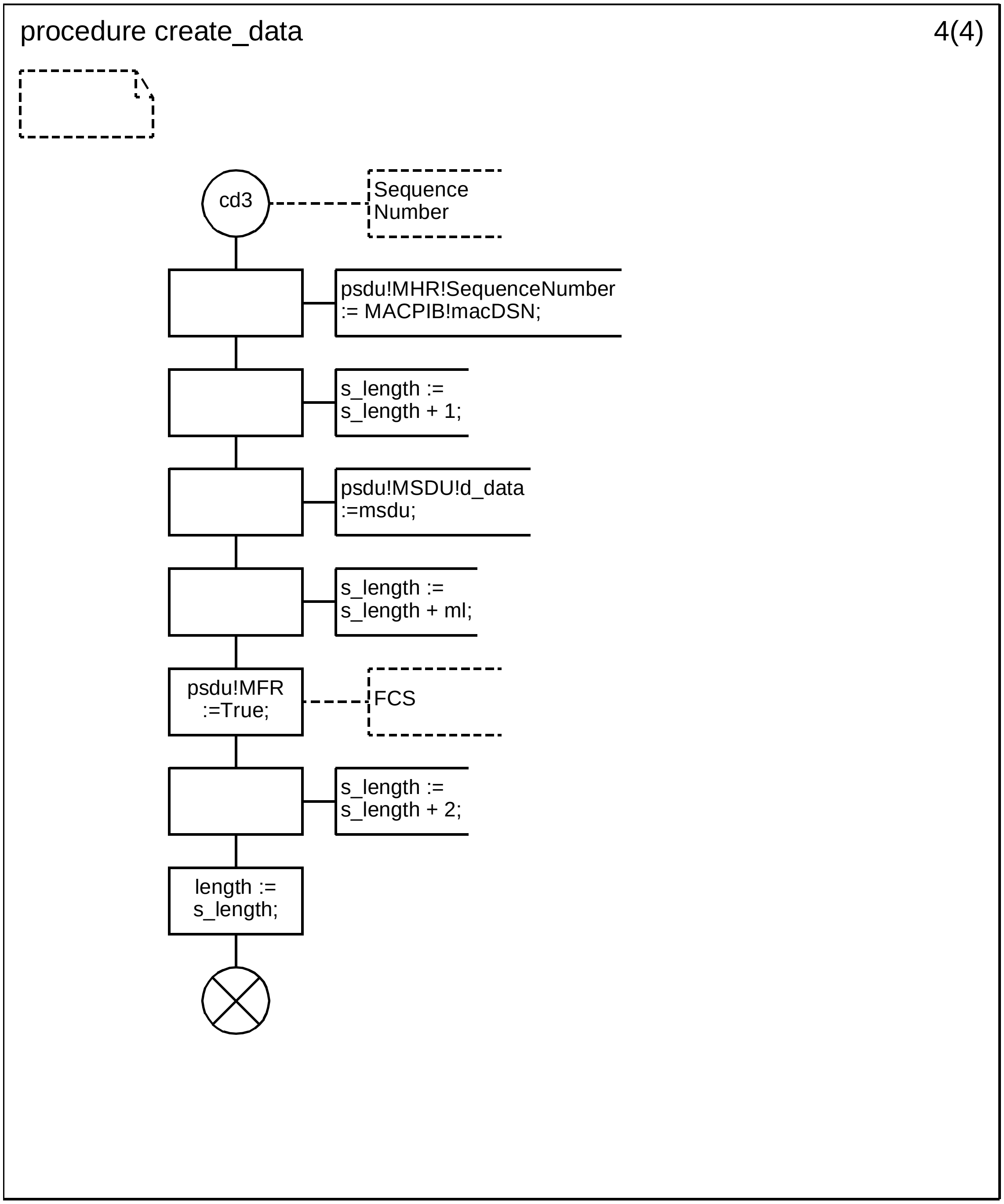
D.3.1.154.29 Procedure create_data (3)

.



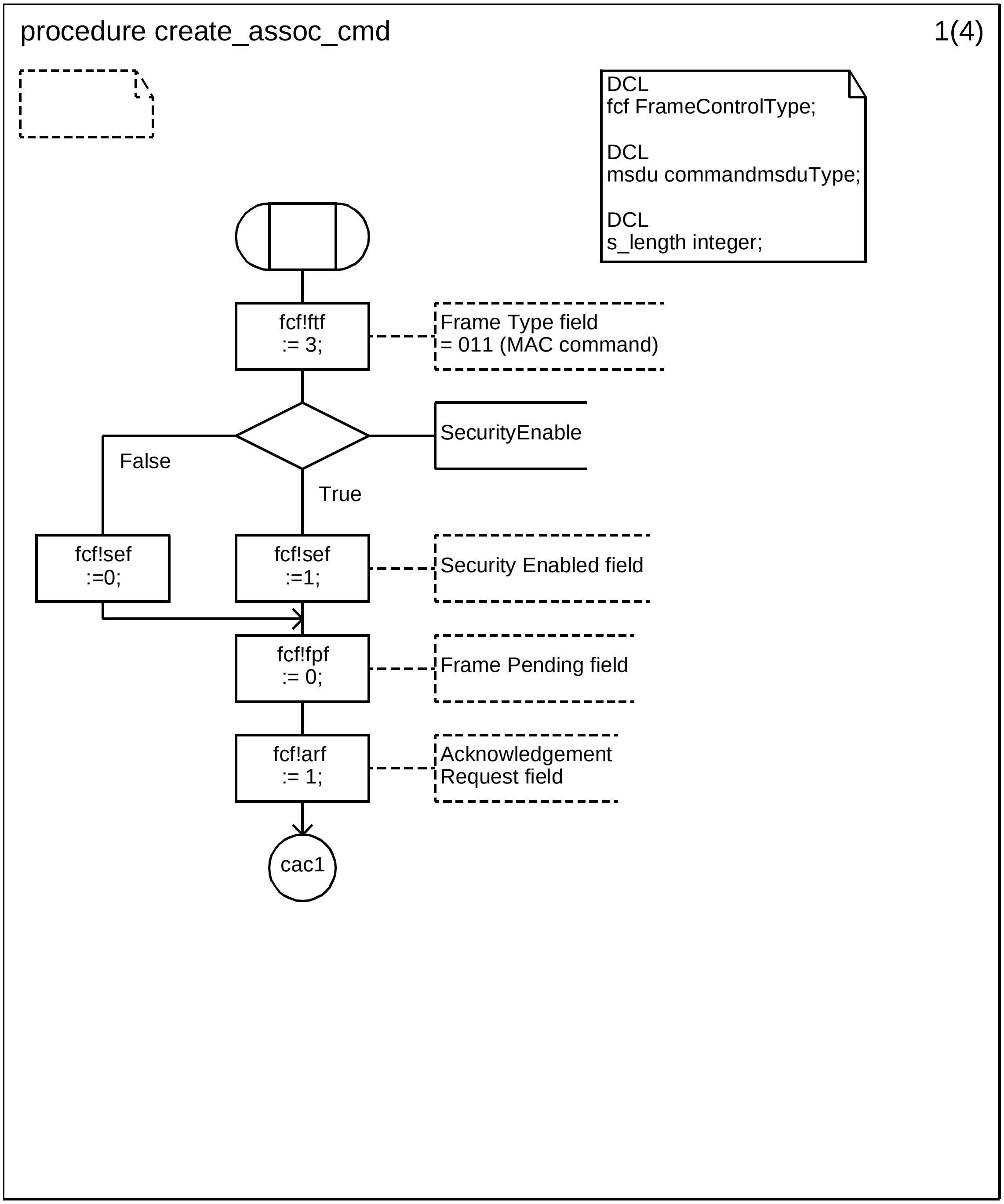
D.3.1.154.30 Procedure create_data (4)

.



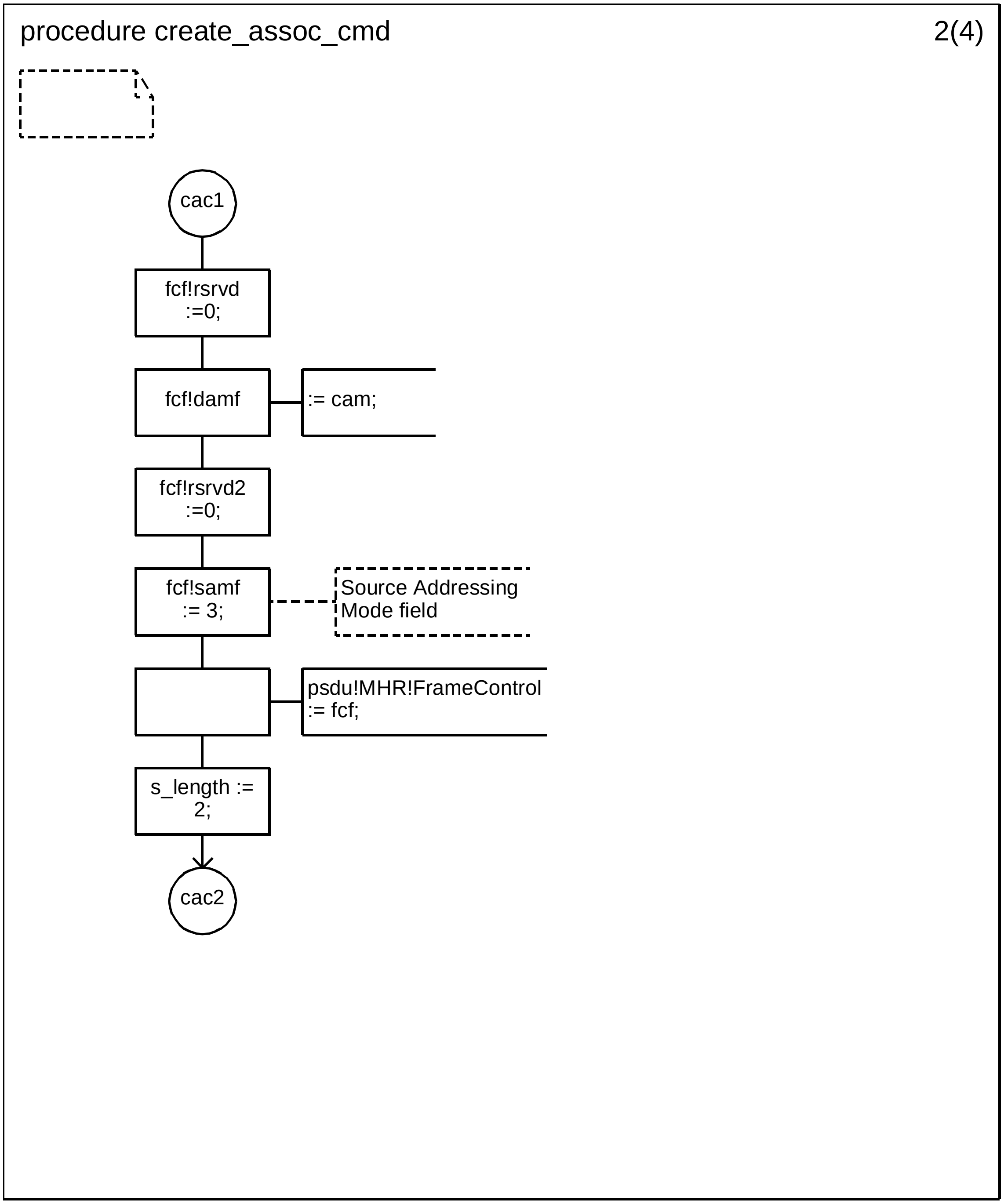
D.3.1.154.31 Procedure create_assoc_cmd (1)

.



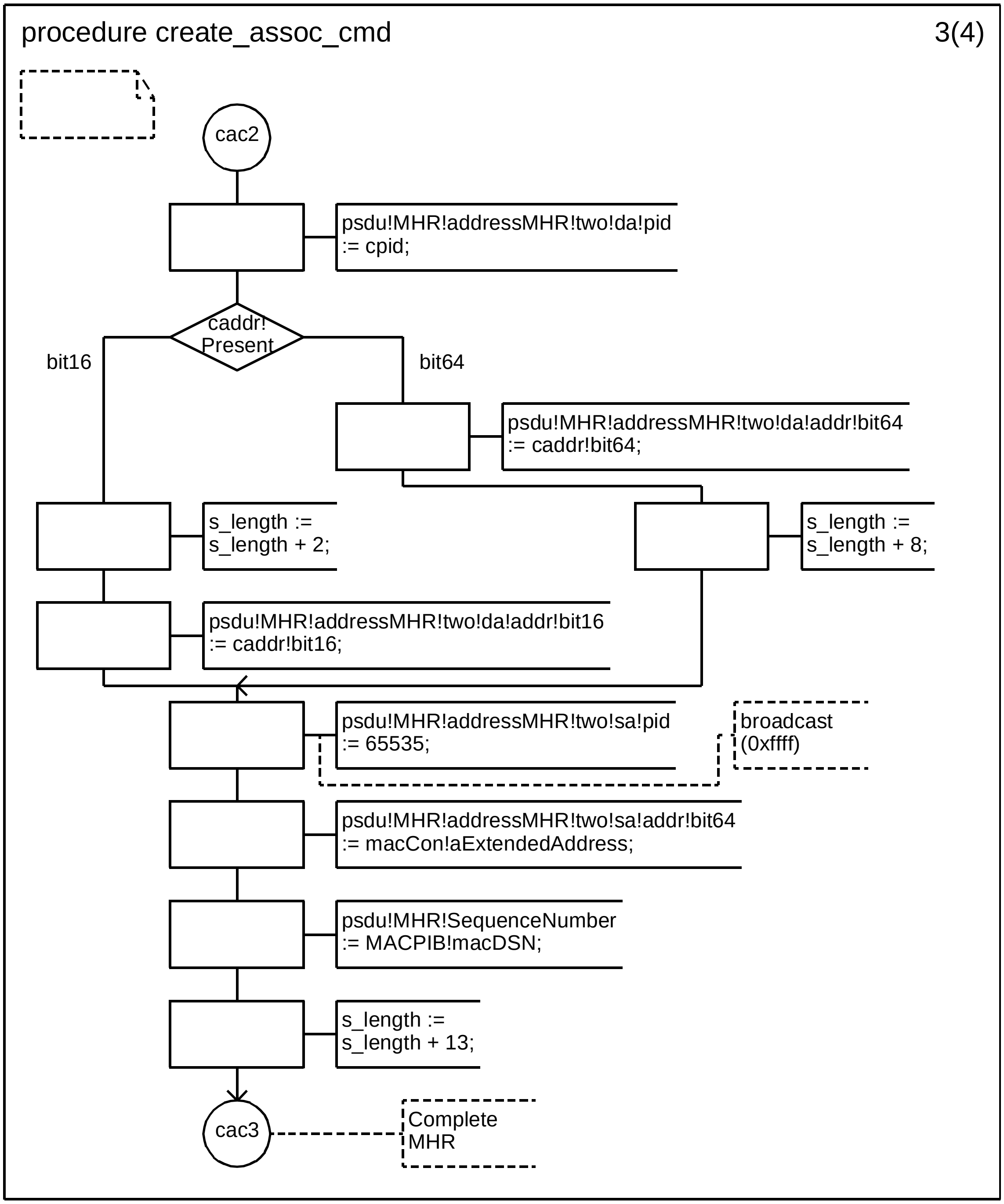
D.3.1.154.32 Procedure create_assoc_cmd (2)

.



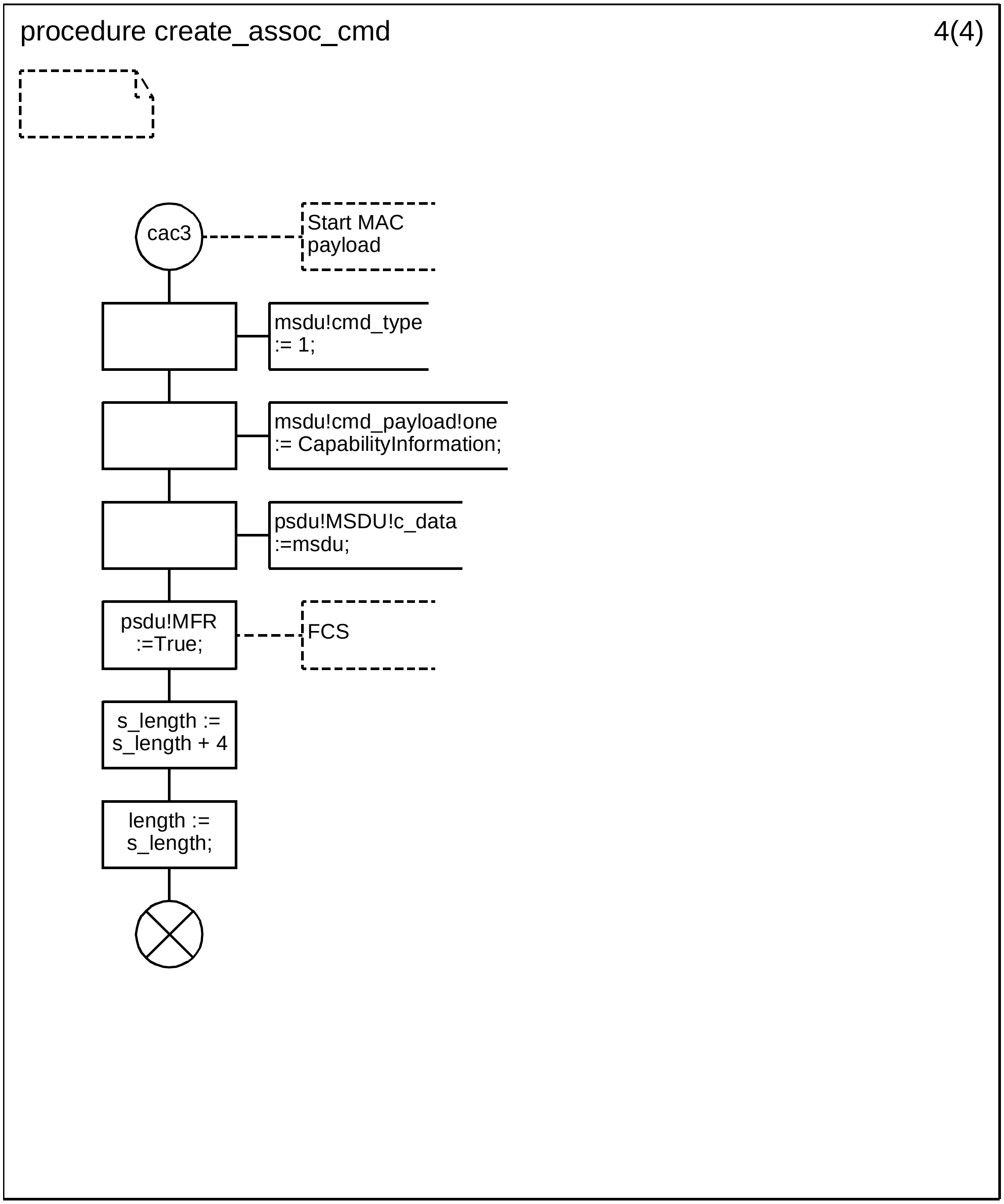
D.3.1.154.33 Procedure create_assoc_cmd (3)

.


procedure create_assoc_cmd
3(4)
cac2
psdu!MHR!addressMHR!two!da!pid
:= cpid;
caddr!
Present
bit16
bit64
psdu!MHR!addressMHR!two!da!addr!bit64
:= caddr!bit64;
s_length :=
s_length + 2;
s_length :=
s_length + 8;
psdu!MHR!addressMHR!two!da!addr!bit16
:= caddr!bit16;
psdu!MHR!addressMHR!two!sa!pid
:= 65535;
broadcast
(0xffff)
psdu!MHR!addressMHR!two!sa!addr!bit64
:= macCon!aExtendedAddress;
psdu!MHR!SequenceNumber
:= MACPIB!macDSN;
s_length :=
s_length + 13;
cac3
Complete
MHR

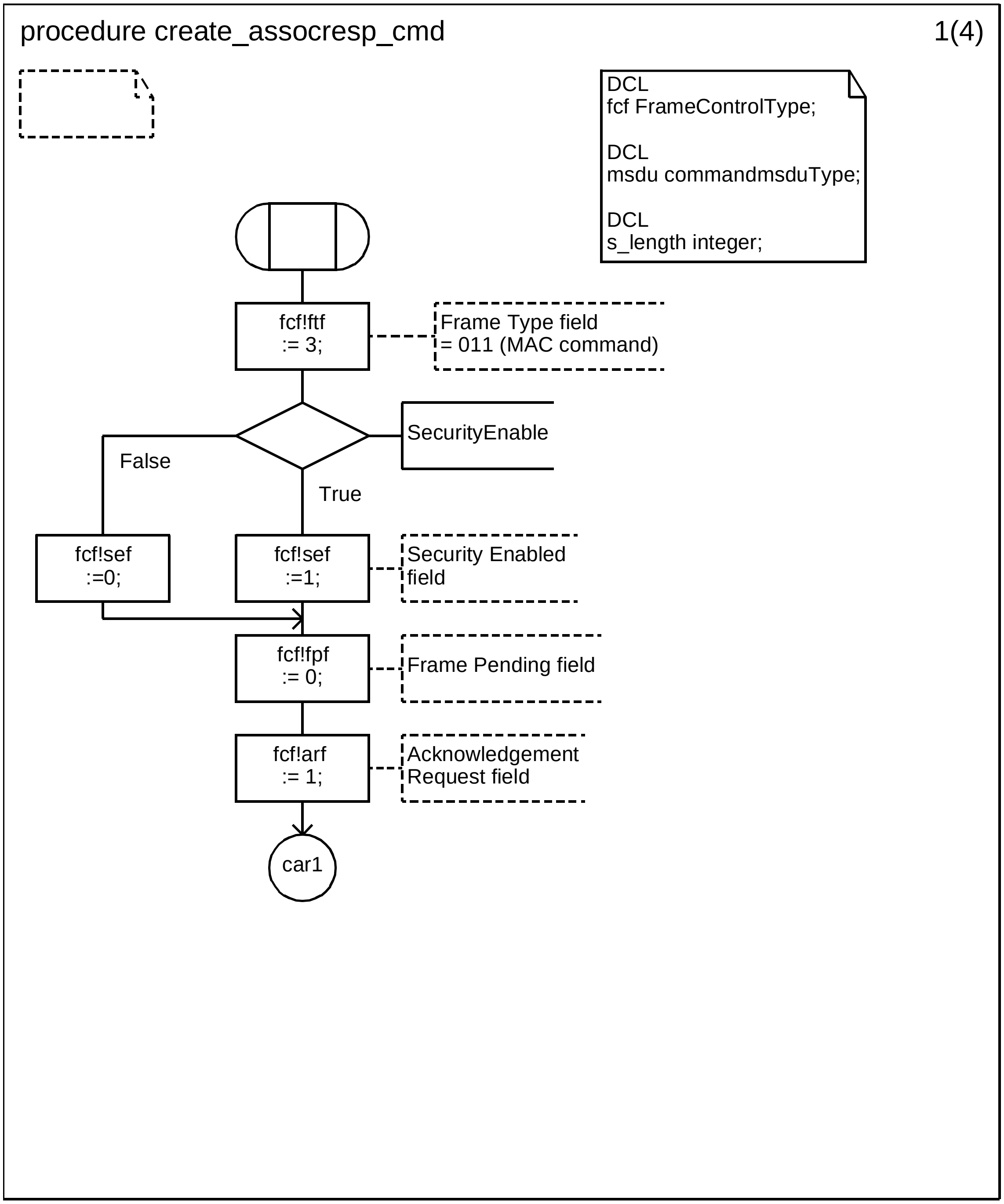
D.3.1.154.34 Procedure create_assoc_cmd (4)

.



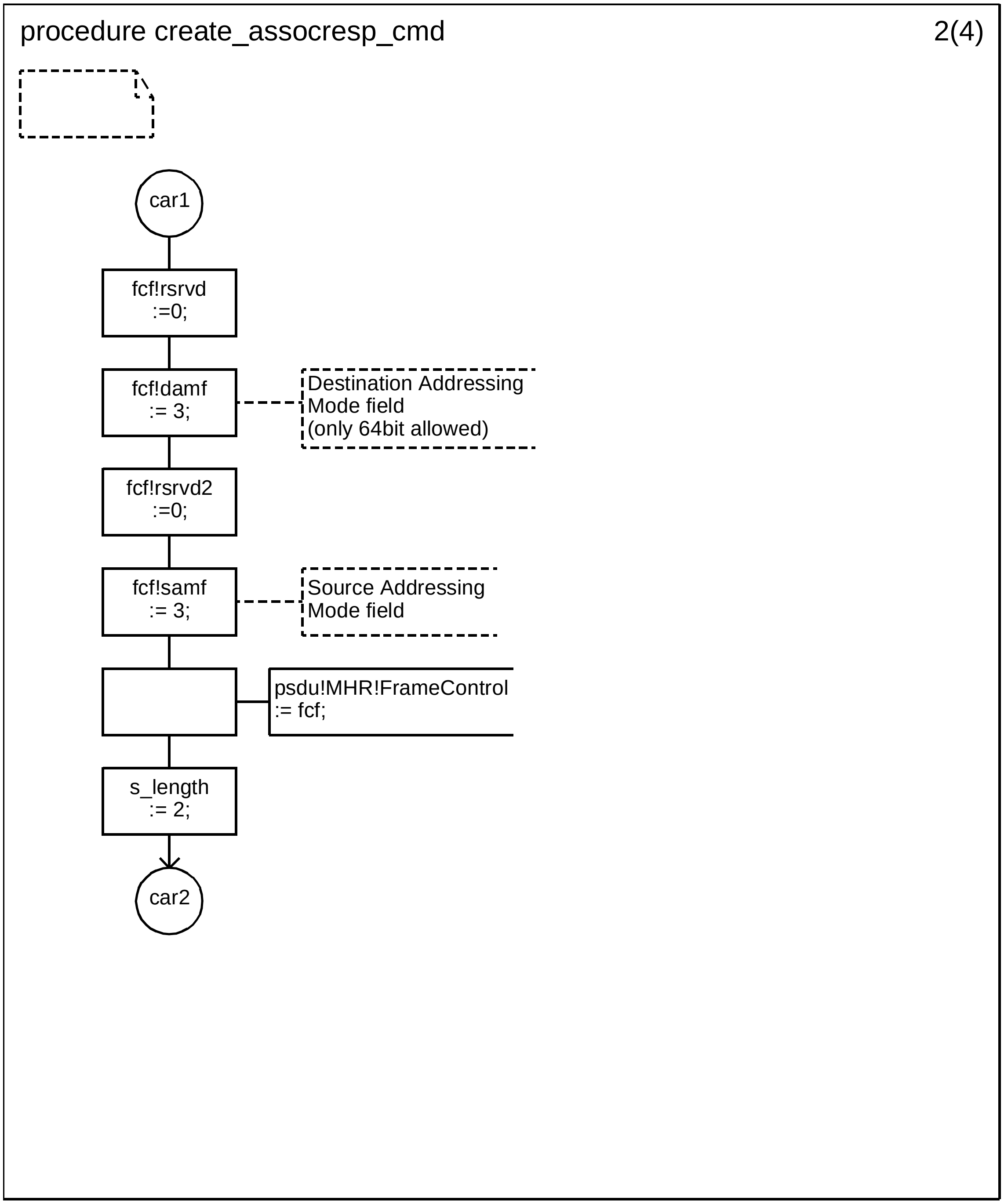
D.3.1.154.35 Procedure create_assocresp_cmd (1)

.



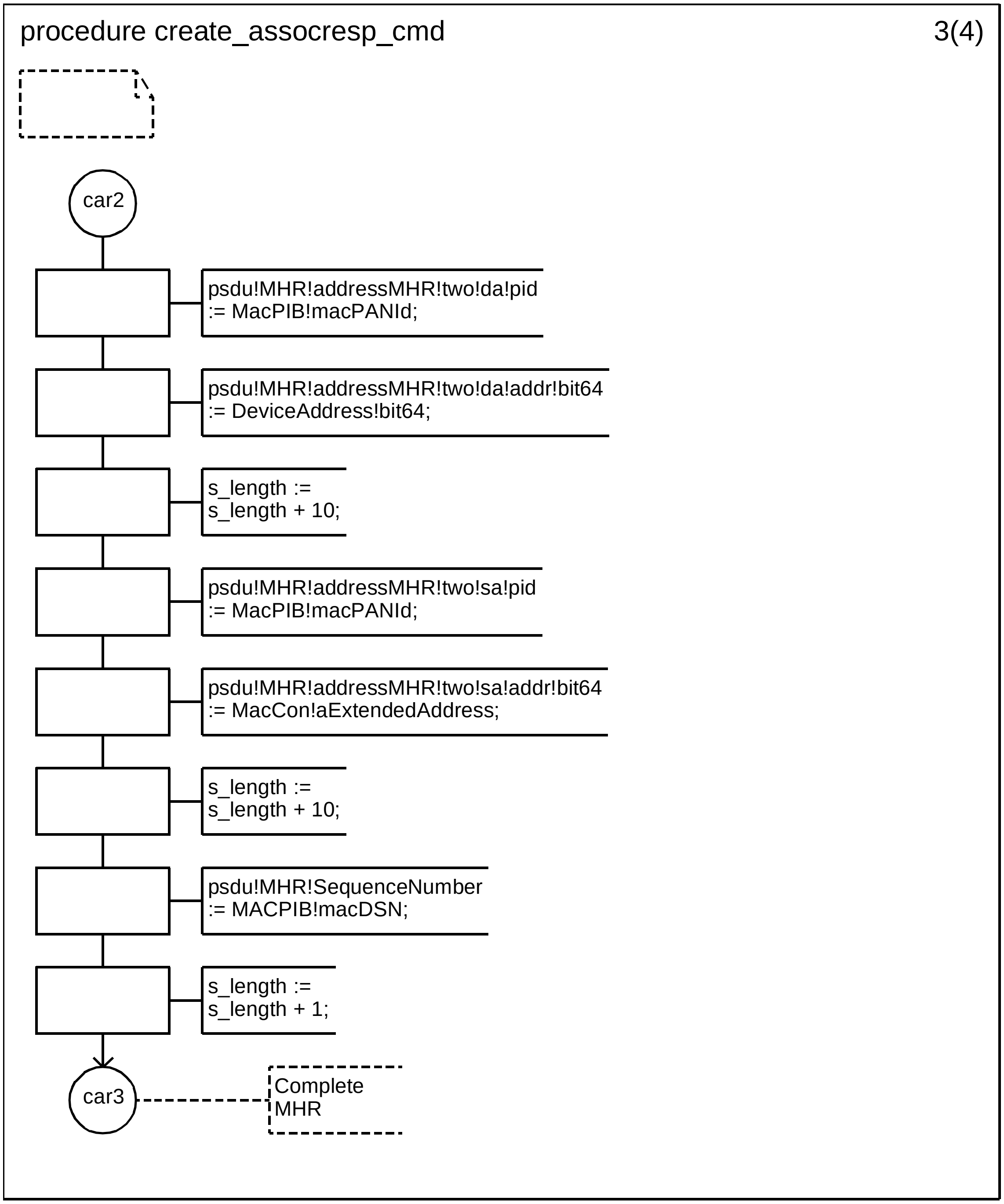
D.3.1.154.36 Procedure create_assocresp_cmd (2)

.



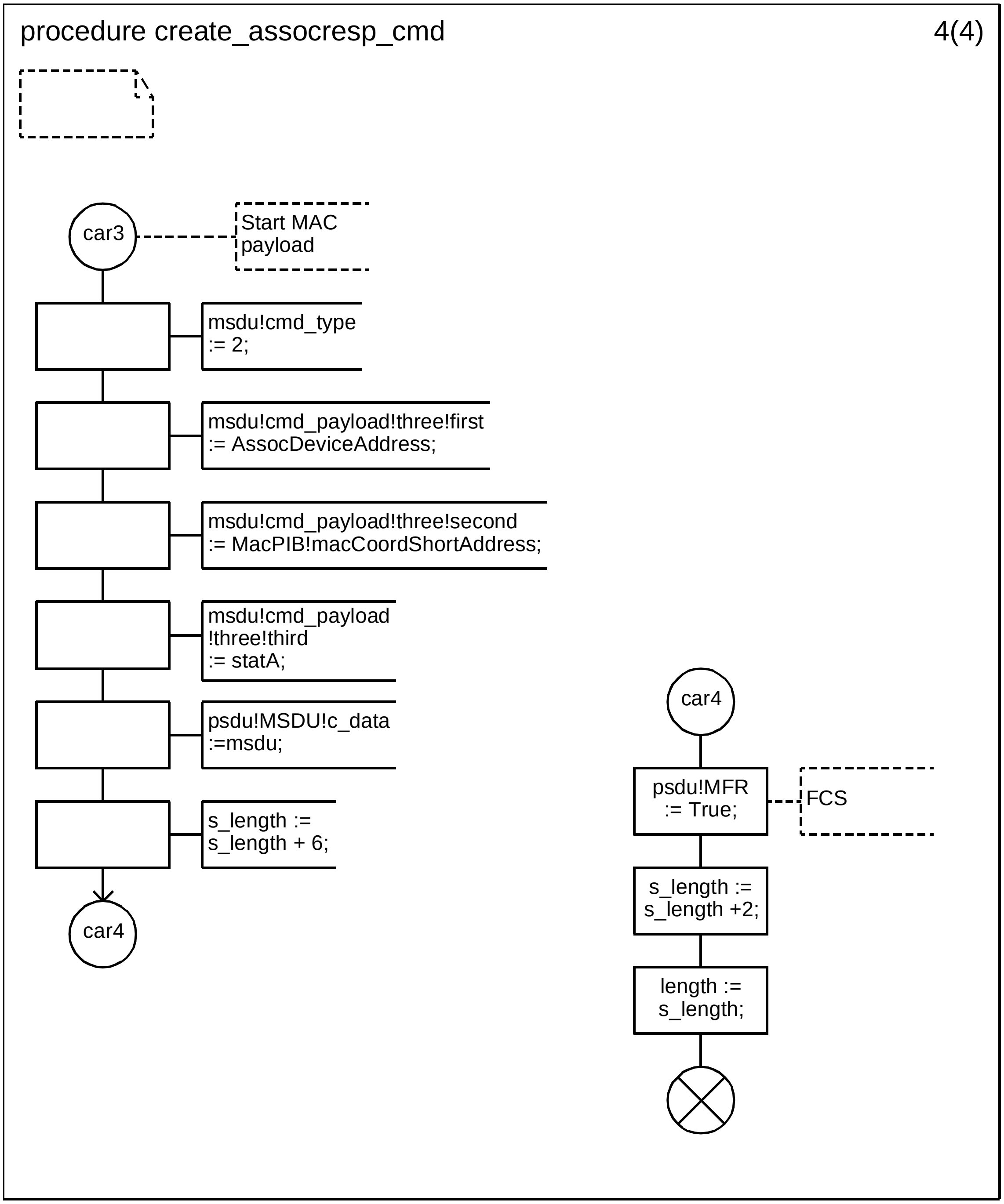
D.3.1.154.37 Procedure create_assocresp_cmd (3)

.


procedure create_assocresp_cmd
3(4)
car2
psdu!MHR!addressMHR!two!da!pid := MacPIB!macPANId;
psdu!MHR!addressMHR!two!da!addr!bit64 := DeviceAddress!bit64;
s_length := s_length + 10;
psdu!MHR!addressMHR!two!sa!pid := MacPIB!macPANId;
psdu!MHR!addressMHR!two!sa!addr!bit64 := MacCon!aExtendedAddress;
s_length := s_length + 10;
psdu!MHR!SequenceNumber := MACPIB!macDSN;
s_length := s_length + 1;
car3
Complete MHR

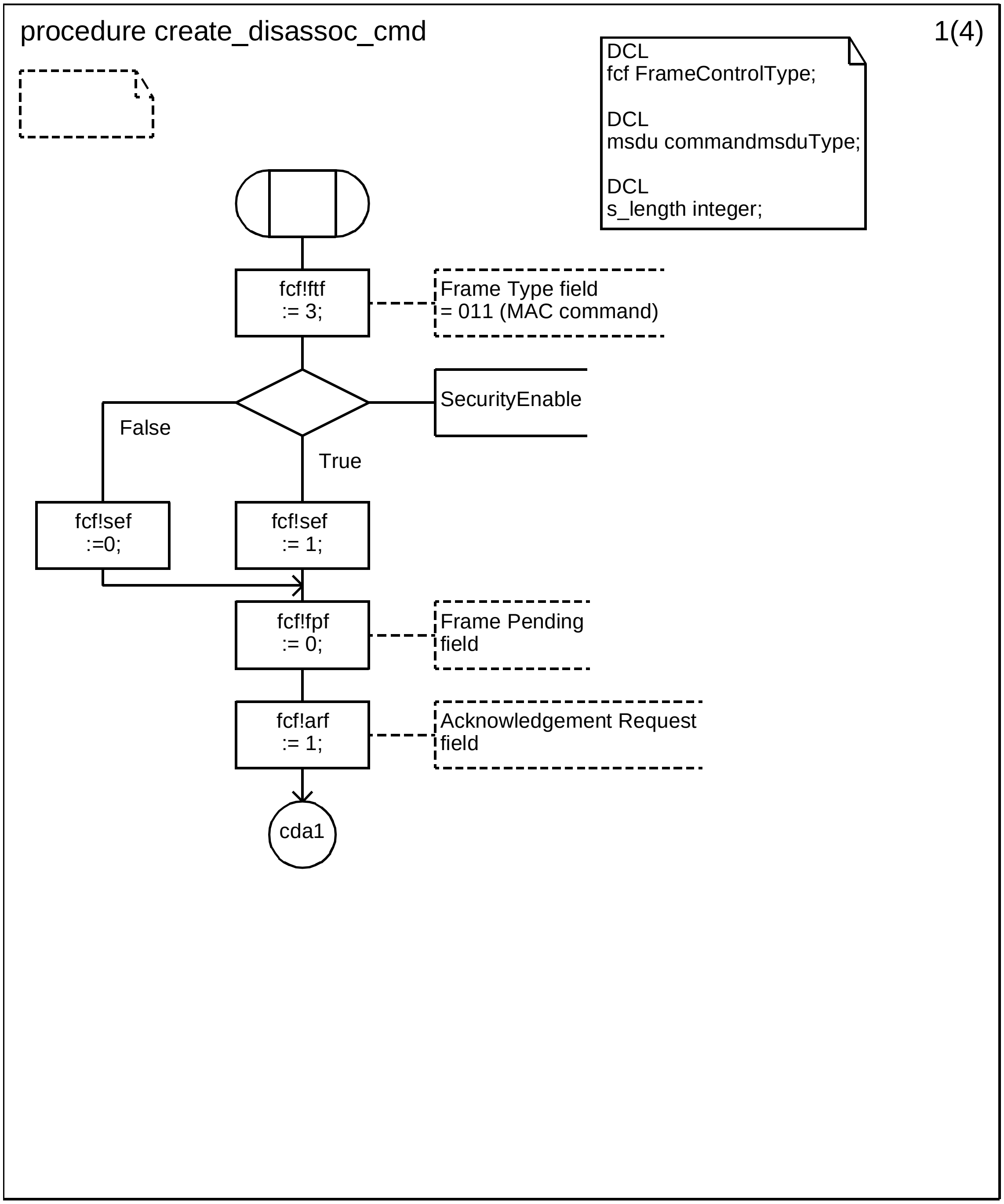
D.3.1.154.38 Procedure create_assocresp_cmd (4)

.


procedure create_assocresp_cmd
4(4)
car3
Start MAC payload
msdu!cmd_type := 2;
msdu!cmd_payload!three!first := AssocDeviceAddress;
msdu!cmd_payload!three!second := MacPIB!macCoordShortAddress;
msdu!cmd_payload !three!third := statA;
psdu!MSDU!c_data :=msdu;
s_length := s_length + 6;
car4
car4
psdu!MFR := True;
FCS
s_length := s_length +2;
length := s_length;

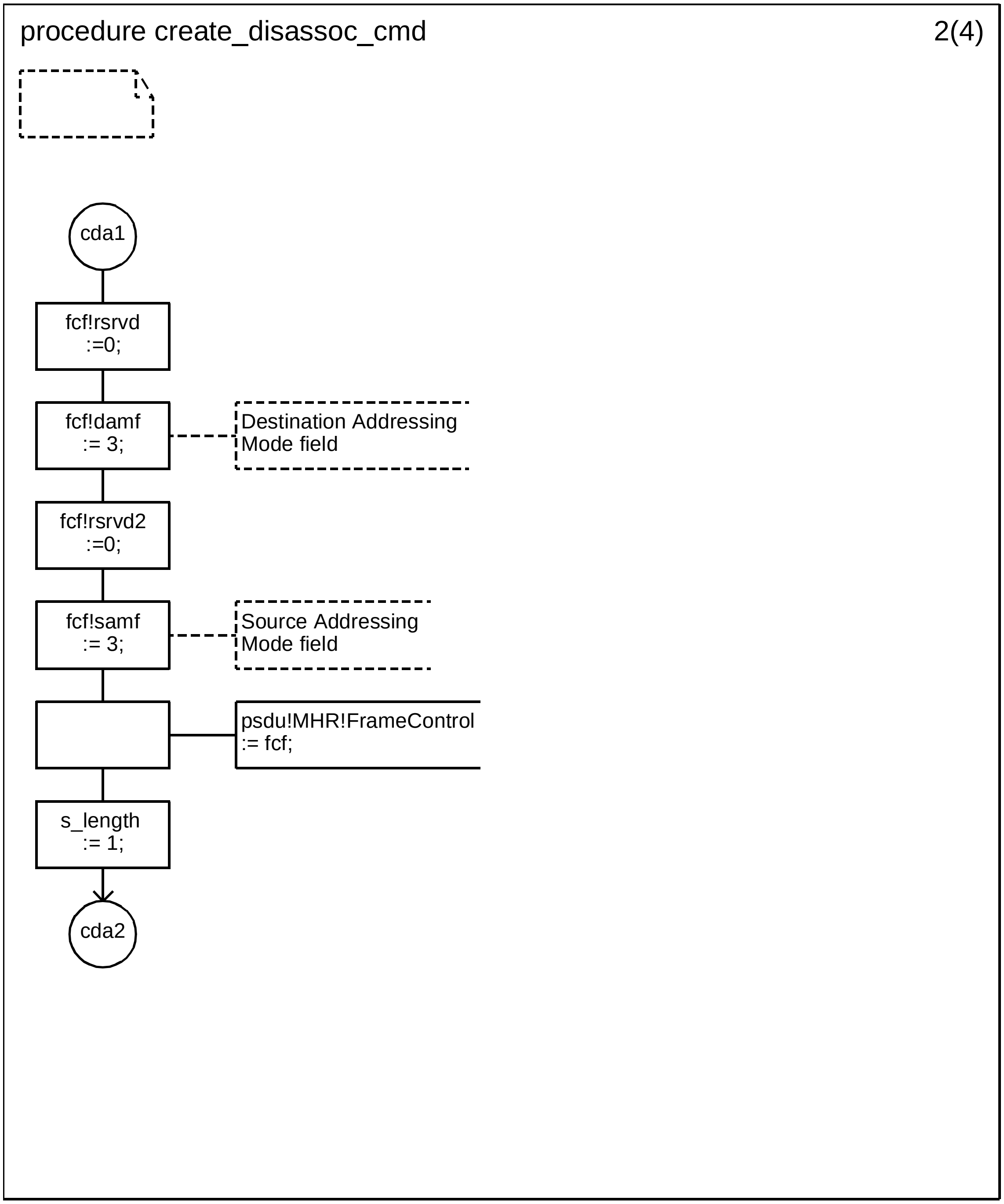
D.3.1.154.39 Procedure create_disassoc_cmd (1)

.



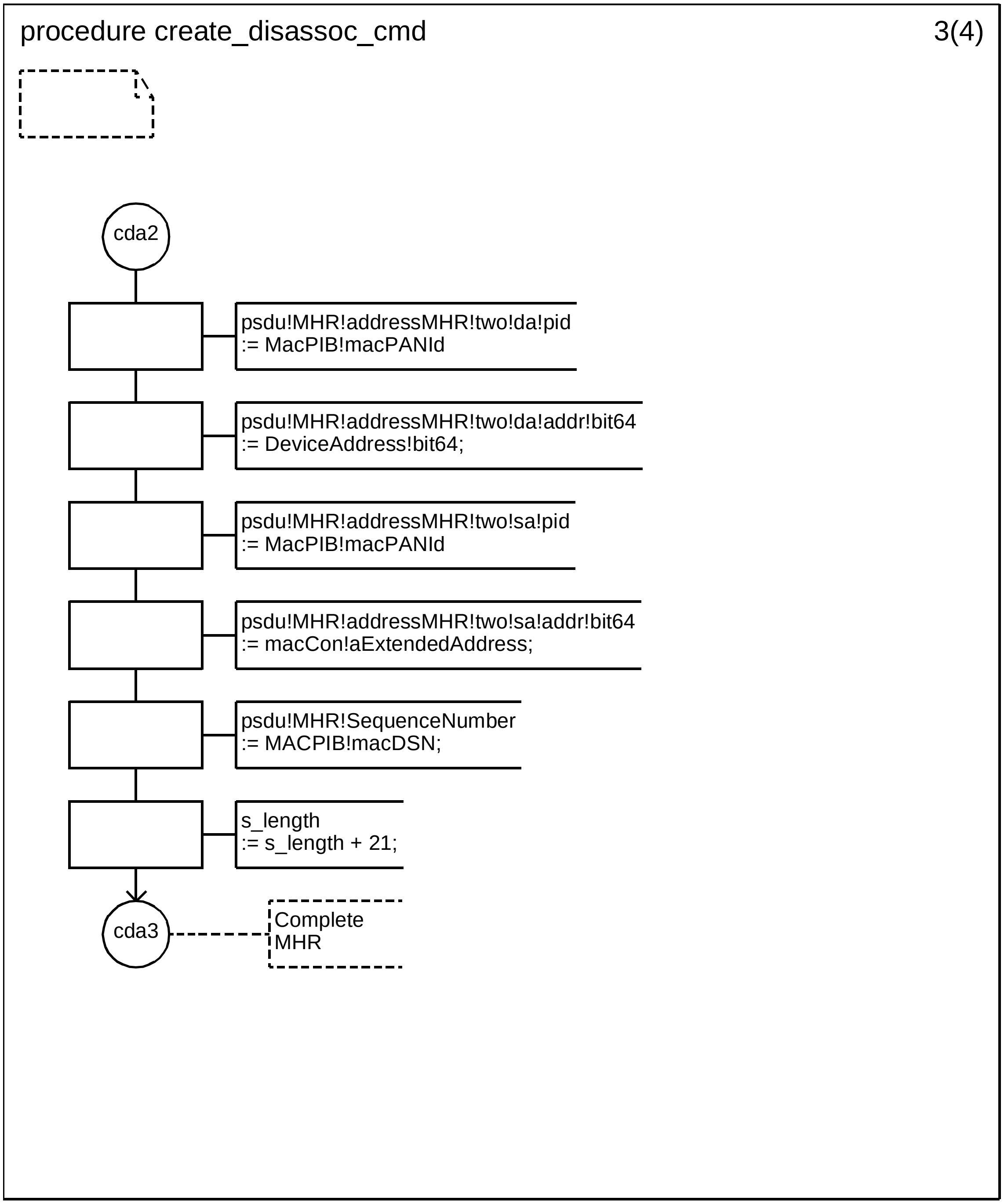
D.3.1.154.40 Procedure create_disassoc_cmd (2)

.


procedure create_disassoc_cmd
2(4)
cda1
fcf!rsrvd :=0;
fcf!damf := 3;
Destination Addressing Mode field
fcf!rsrvd2 :=0;
fcf!samf := 3;
Source Addressing Mode field
psdu!MHR!FrameControl := fcf;
s_length := 1;
cda2

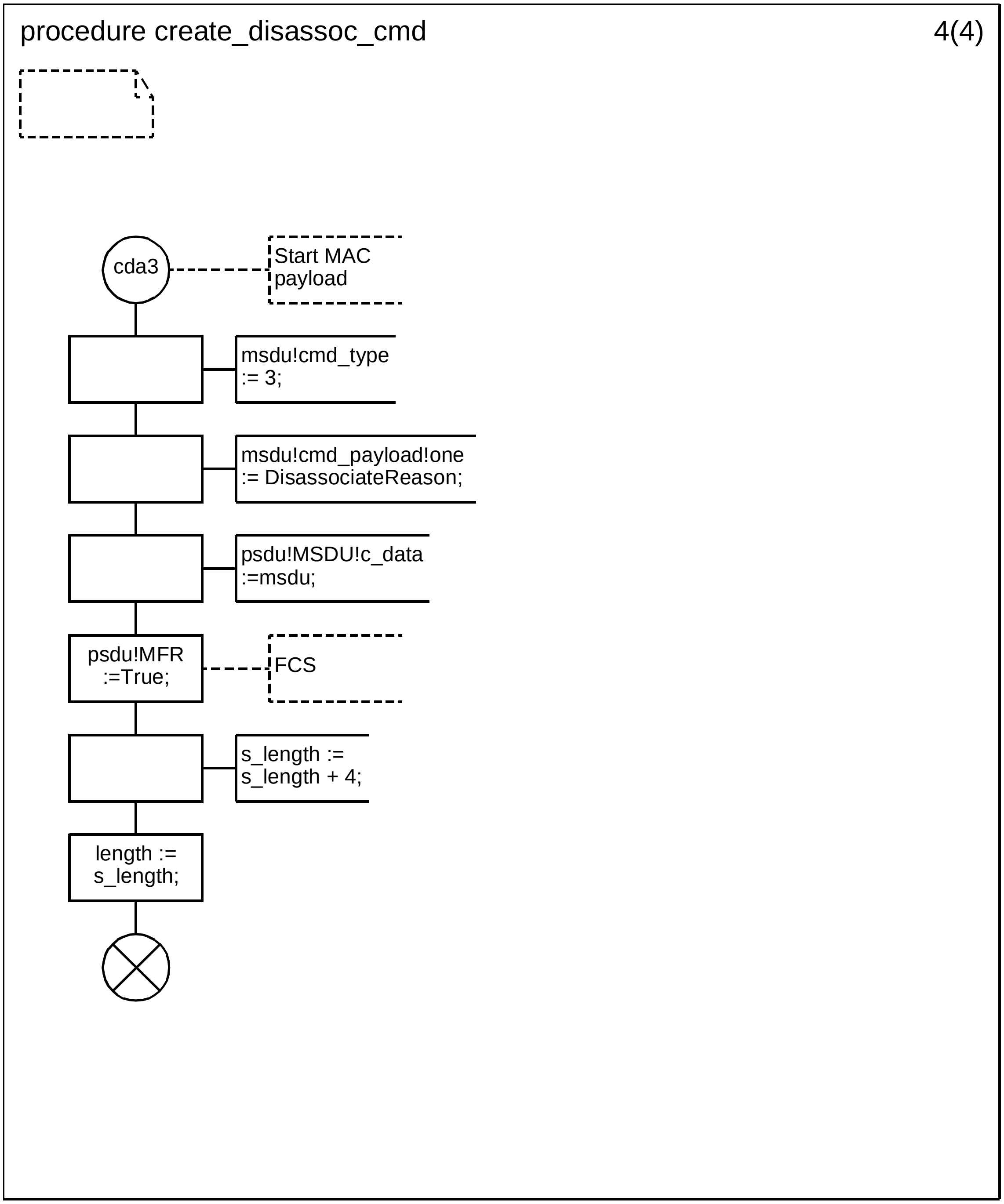
D.3.1.154.41 Procedure create_disassoc_cmd (3)

.


procedure create_disassoc_cmd
3(4)
cda2
psdu!MHR!addressMHR!two!da!pid
:= MacPIB!macPANId
psdu!MHR!addressMHR!two!da!addr!bit64
:= DeviceAddress!bit64;
psdu!MHR!addressMHR!two!sa!pid
:= MacPIB!macPANId
psdu!MHR!addressMHR!two!sa!addr!bit64
:= macCon!aExtendedAddress;
psdu!MHR!SequenceNumber
:= MACPIB!macDSN;
s_length
:= s_length + 21;
cda3
Complete
MHR

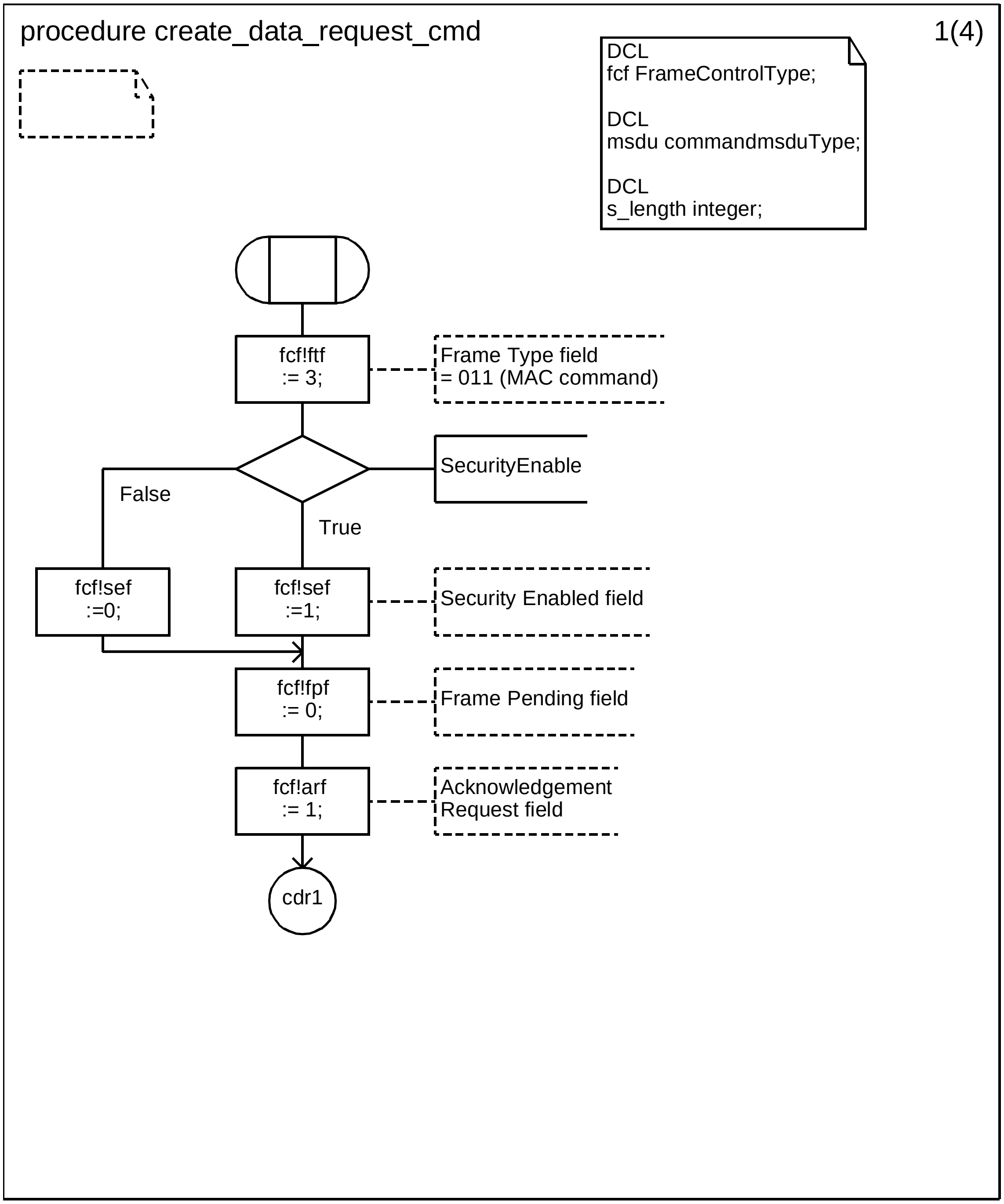
D.3.1.154.42 Procedure create_disassoc_cmd (4)

.



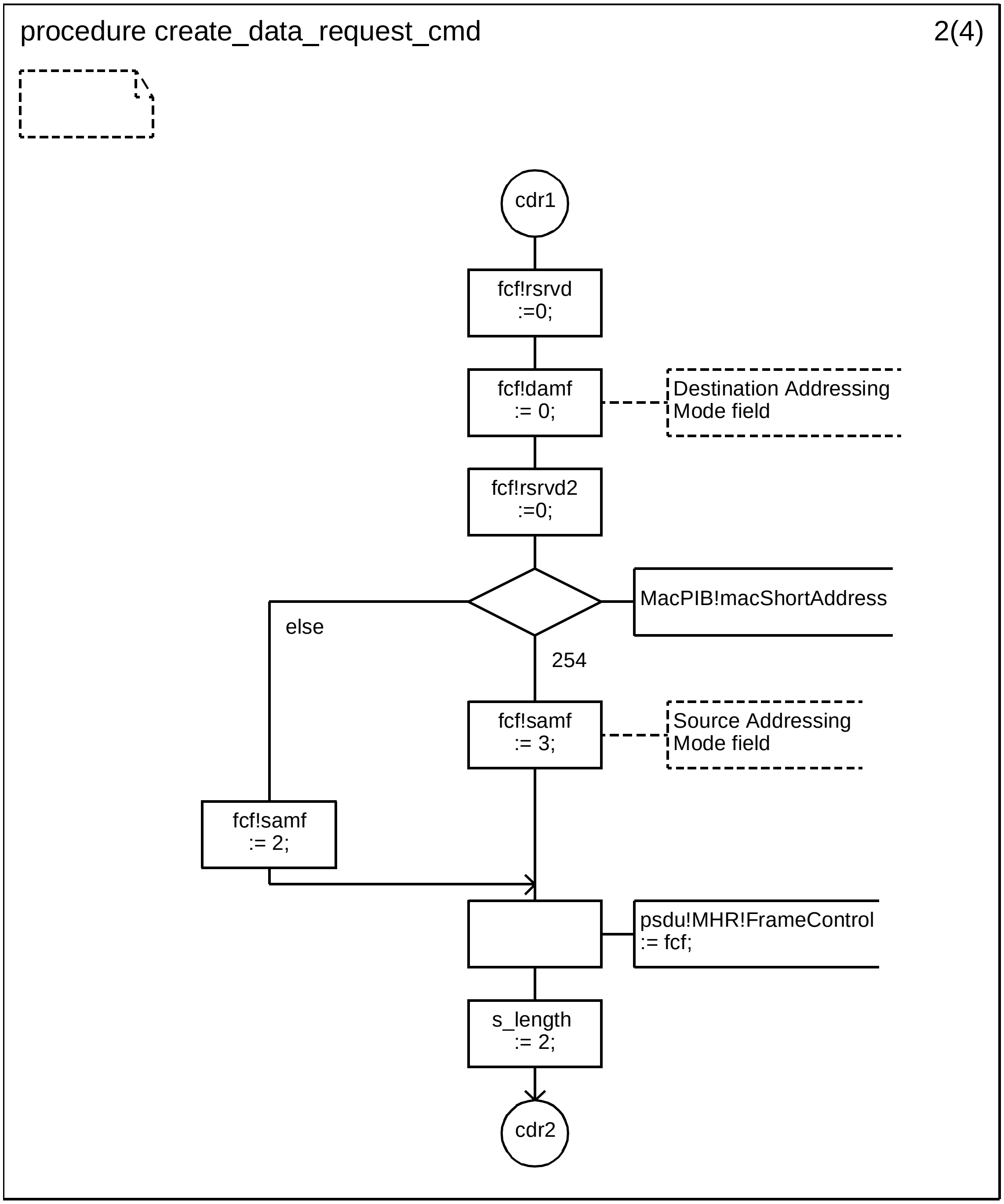
D.3.1.154.43 Procedure create_data_request_cmd (1)

.



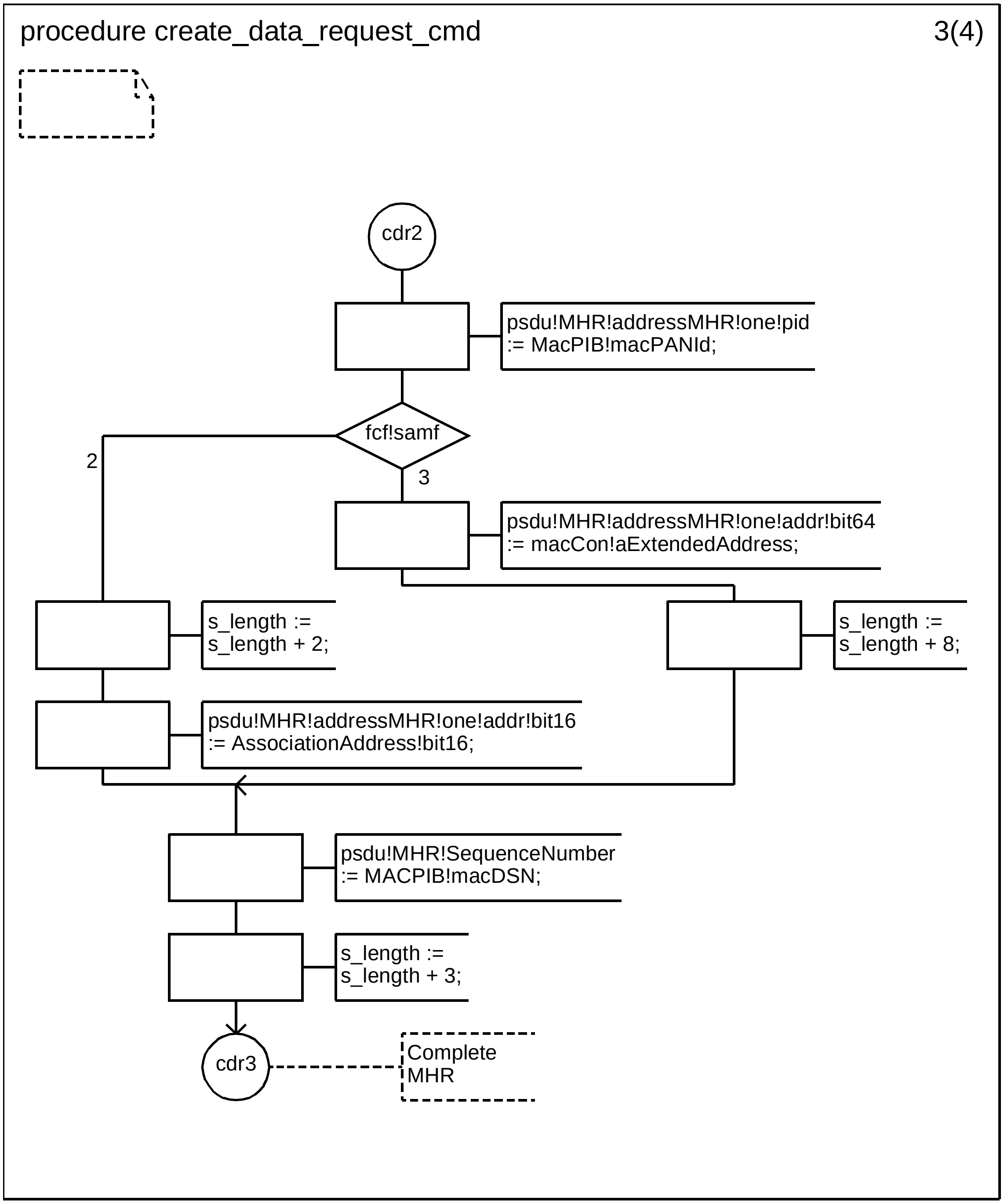
D.3.1.154.44 Procedure create_data_request_cmd (2)

.



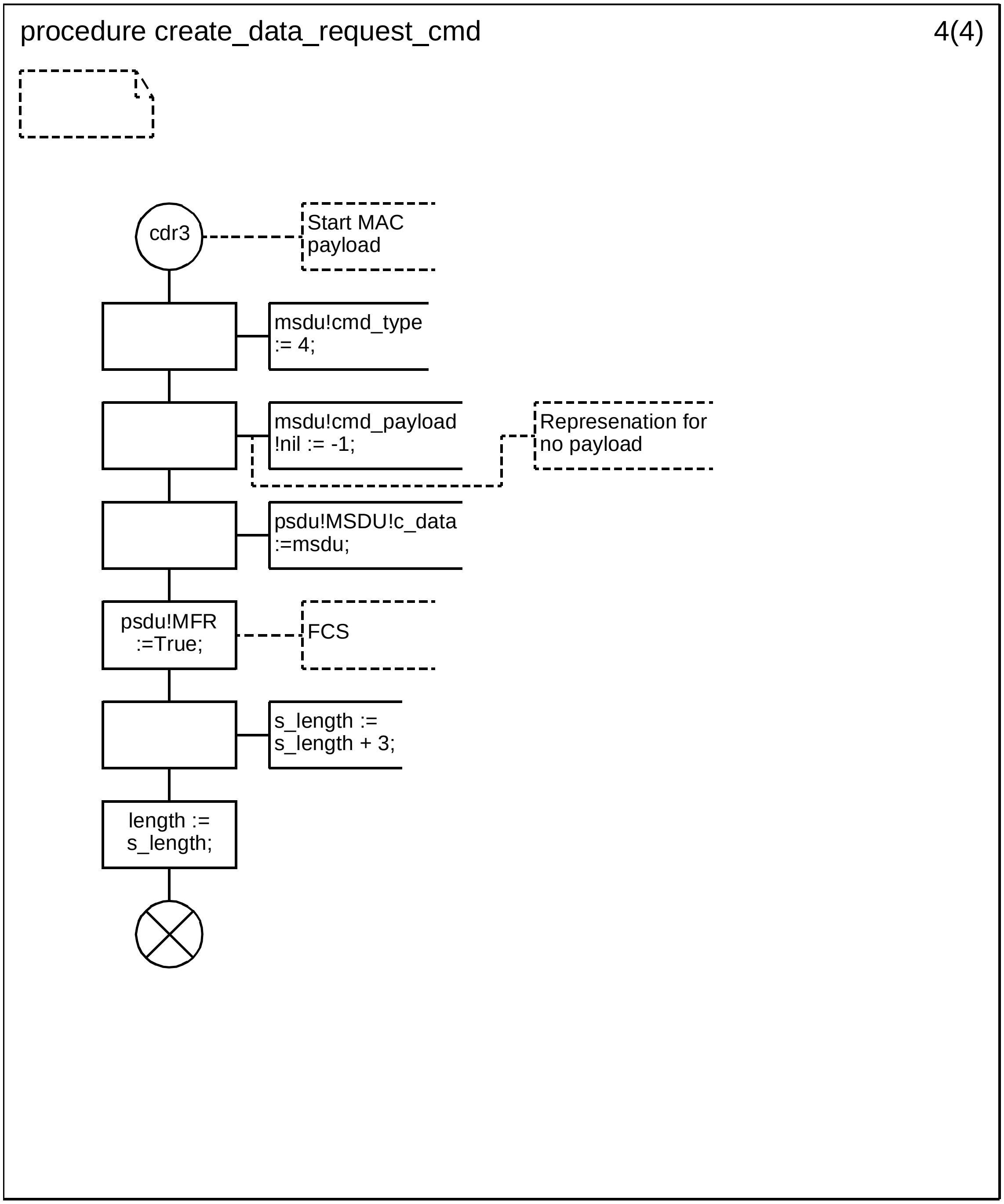
D.3.1.154.45 Procedure create_data_request_cmd (3)

.


procedure create_data_request_cmd
3(4)
cdr2
psdu!MHR!addressMHR!one!pid
:= MacPIB!macPANId;
fcf!samf
2
3
psdu!MHR!addressMHR!one!addr!bit64
:= macCon!aExtendedAddress;
s_length :=
s_length + 2;
s_length :=
s_length + 8;
psdu!MHR!addressMHR!one!addr!bit16
:= AssociationAddress!bit16;
psdu!MHR!SequenceNumber
:= MACPIB!macDSN;
s_length :=
s_length + 3;
cdr3
Complete
MHR

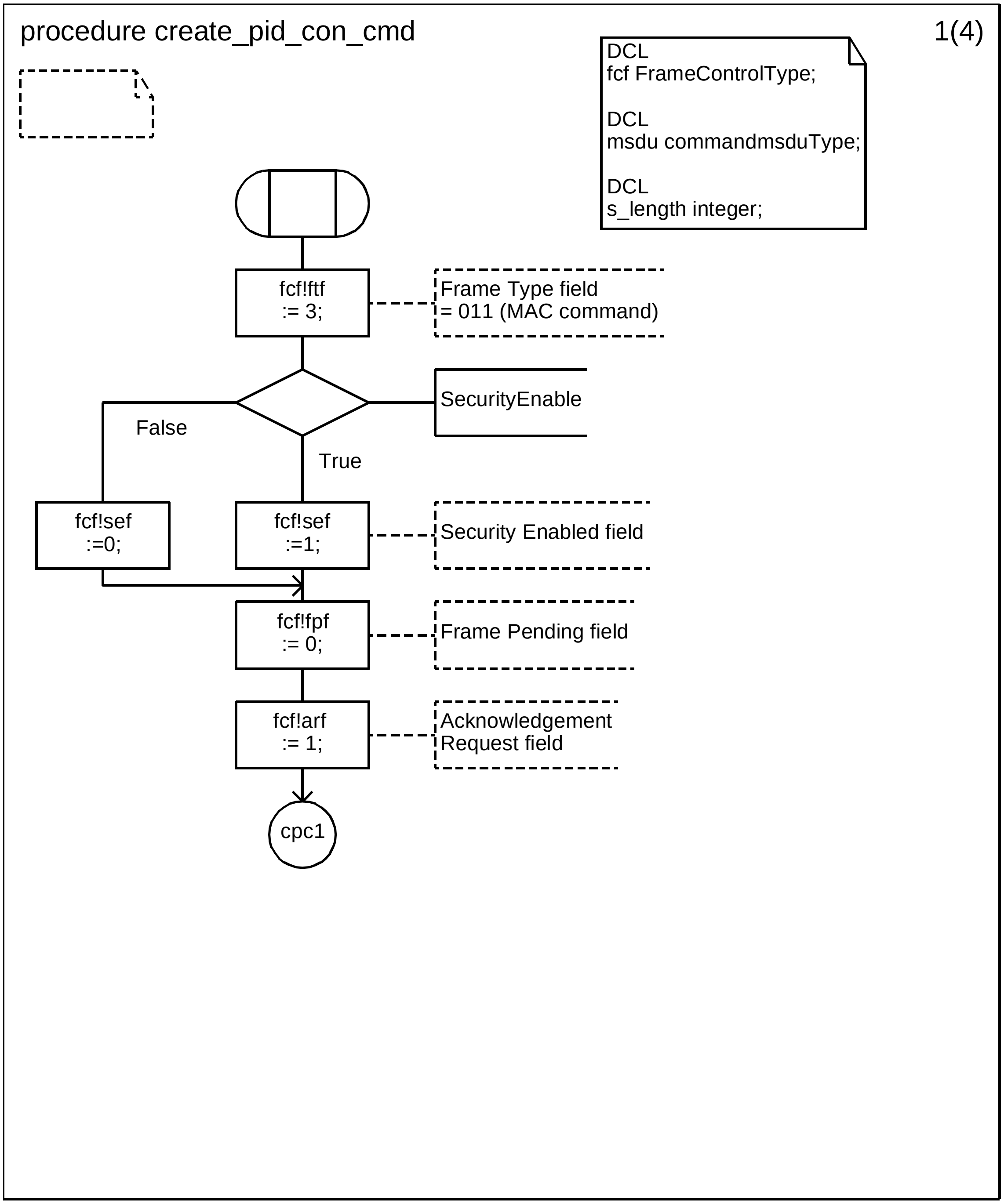
D.3.1.154.46 Procedure create_data_request_cmd (4)

.


procedure create_data_request_cmd
4(4)
cdr3
Start MAC payload
msdu!cmd_type := 4;
msdu!cmd_payload !nil := -1;
Represenation for no payload
psdu!MSDU!c_data :=msdu;
psdu!MFR :=True;
FCS
s_length := s_length + 3;
length := s_length;

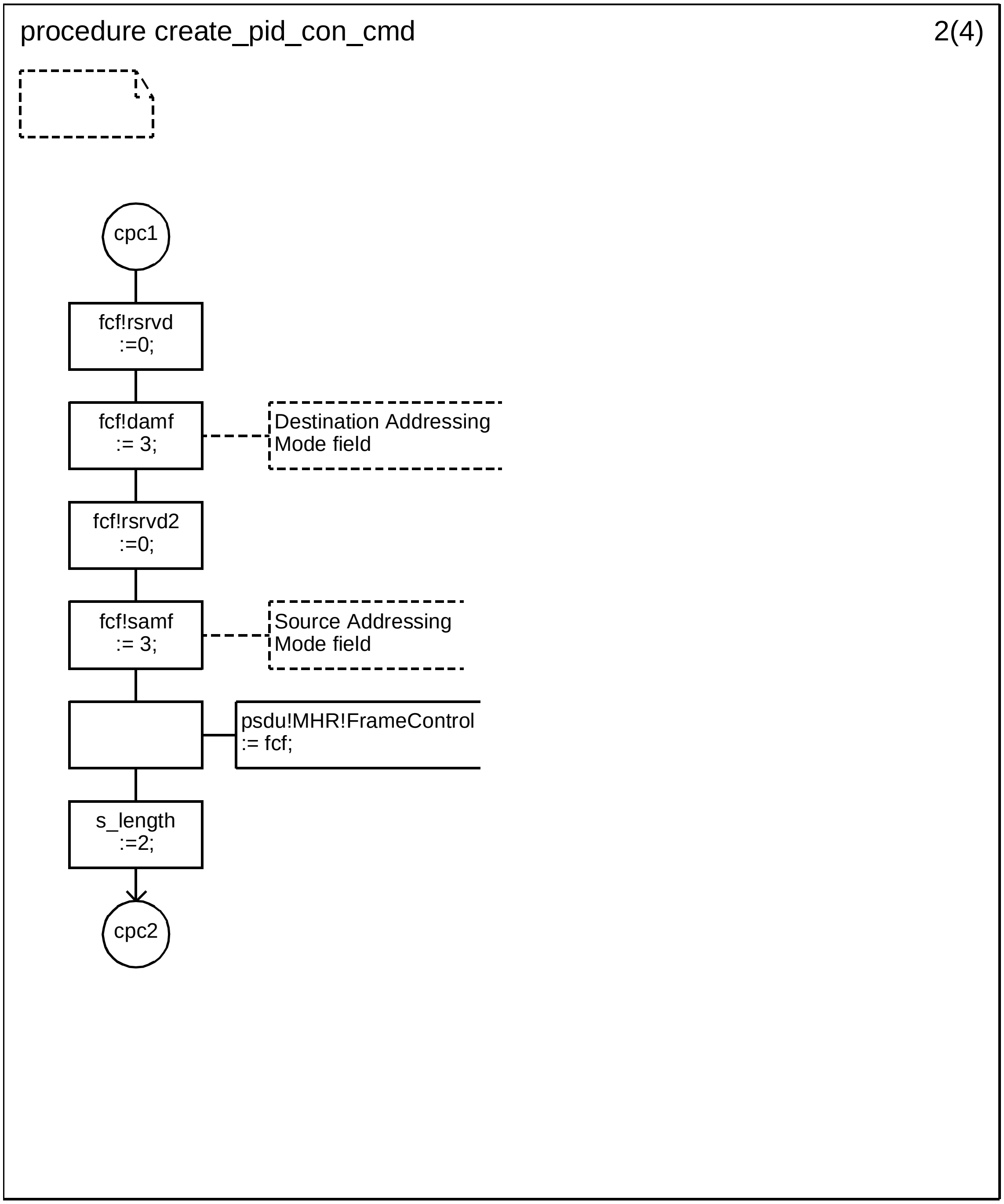
D.3.1.154.47 Procedure create_pid_con_cmd (1)

.



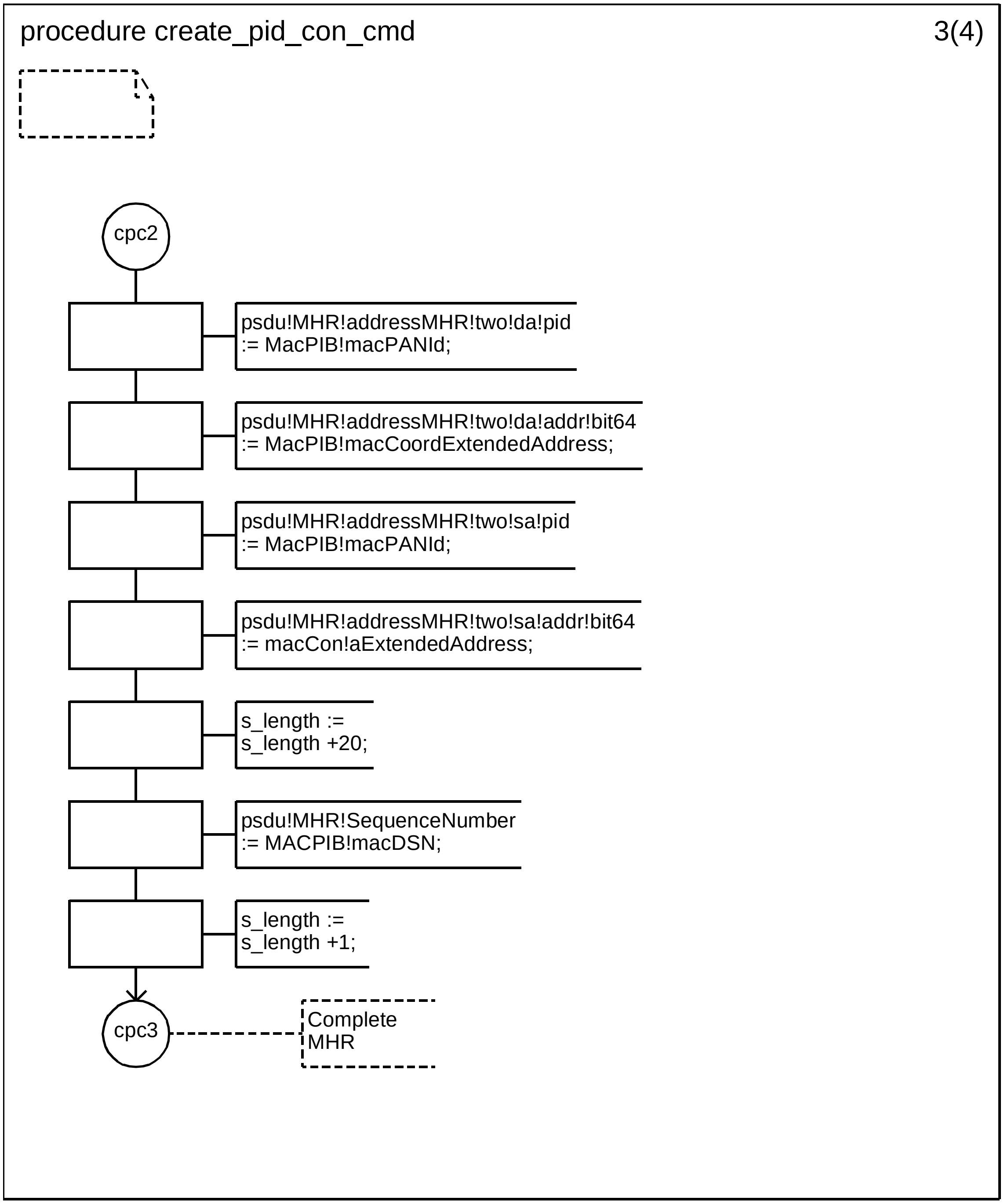
D.3.1.154.48 Procedure create_pid_con_cmd (2)

.



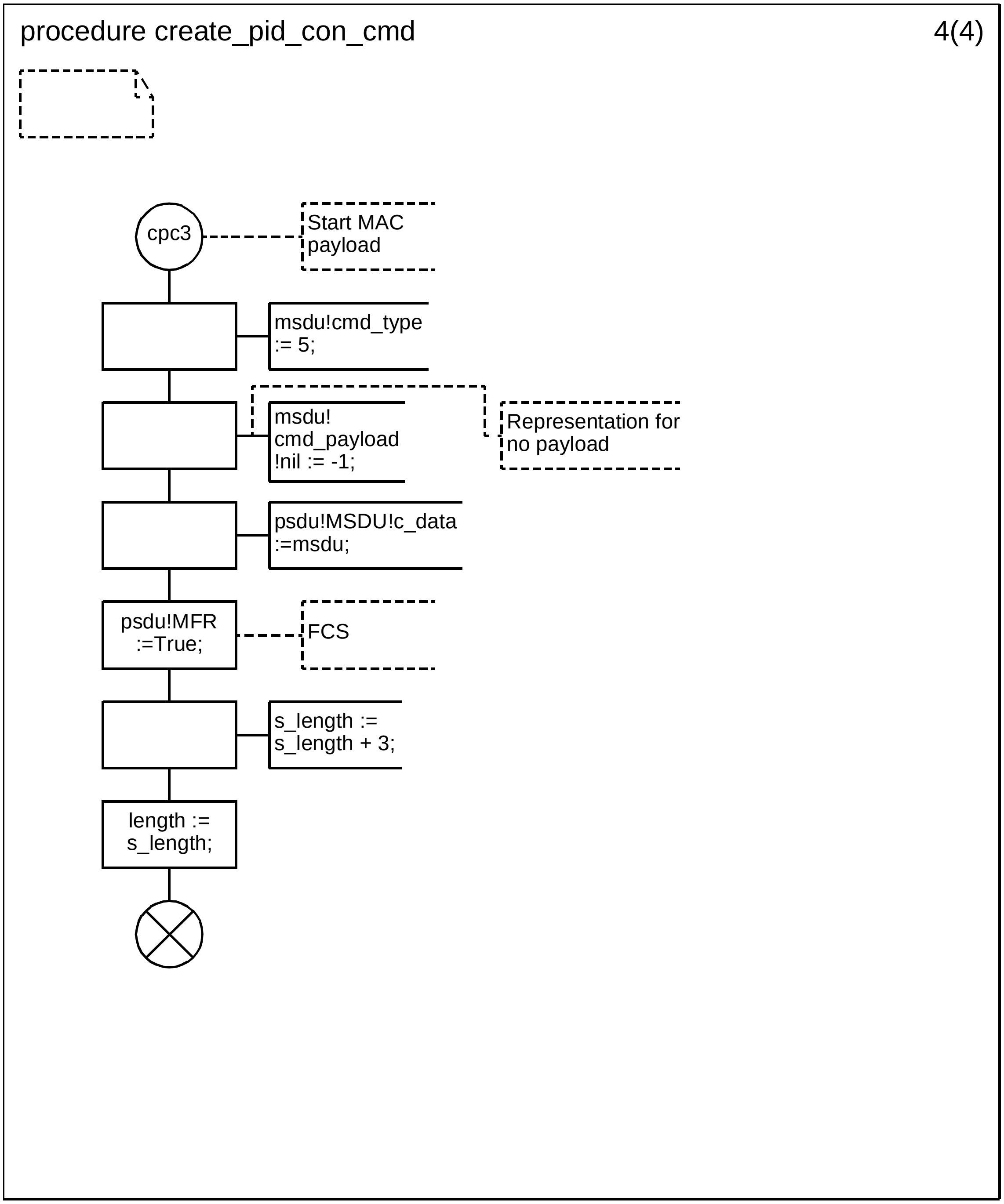
D.3.1.154.49 Procedure create_pid_con_cmd (3)

.


procedure create_pid_con_cmd
3(4)
cpc2
psdu!MHR!addressMHR!two!da!pid
:= MacPIB!macPANId;
psdu!MHR!addressMHR!two!da!addr!bit64
:= MacPIB!macCoordExtendedAddress;
psdu!MHR!addressMHR!two!sa!pid
:= MacPIB!macPANId;
psdu!MHR!addressMHR!two!sa!addr!bit64
:= macCon!aExtendedAddress;
s_length :=
s_length +20;
psdu!MHR!SequenceNumber
:= MACPIB!macDSN;
s_length :=
s_length +1;
cpc3
Complete
MHR

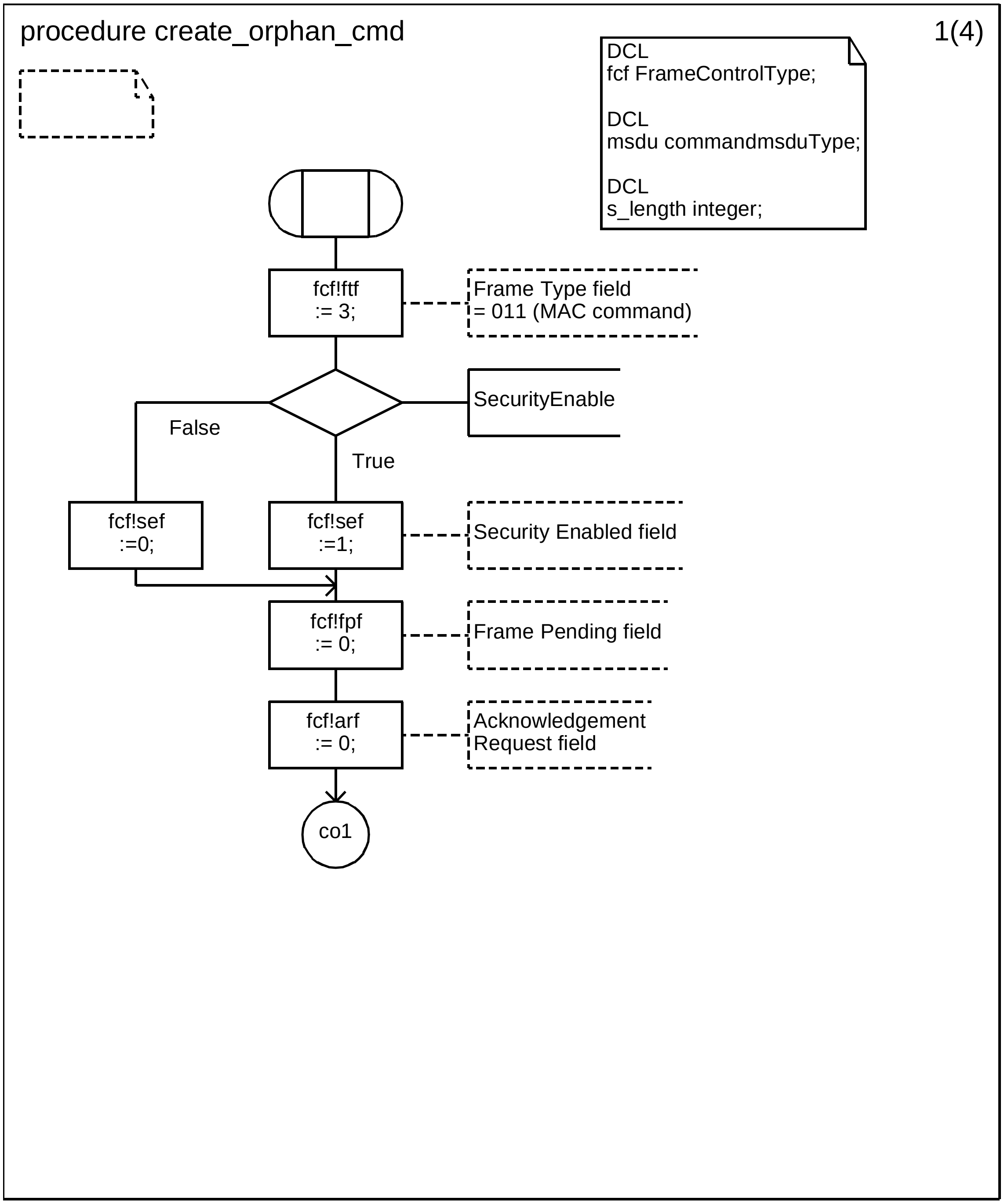
D.3.1.154.50 Procedure create_pid_con_cmd (4)

.


procedure create_pid_con_cmd
4(4)
cpc3
Start MAC payload
msdu!cmd_type := 5;
msdu! cmd_payload !nil := -1;
Representation for no payload
psdu!MSDU!c_data :=msdu;
psdu!MFR :=True;
FCS
s_length := s_length + 3;
length := s_length;

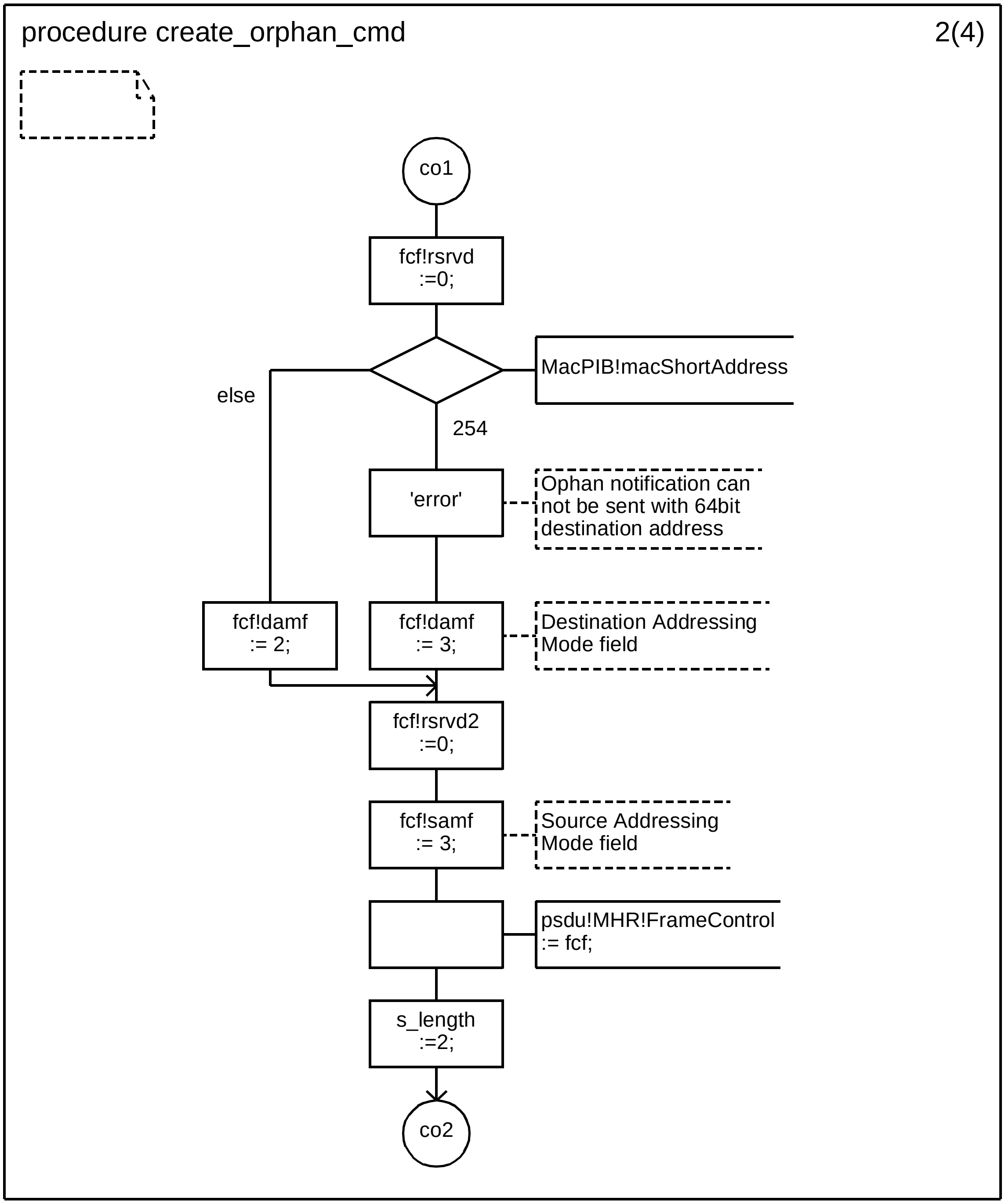
D.3.1.154.51 Procedure create_orphan_cmd (1)

.



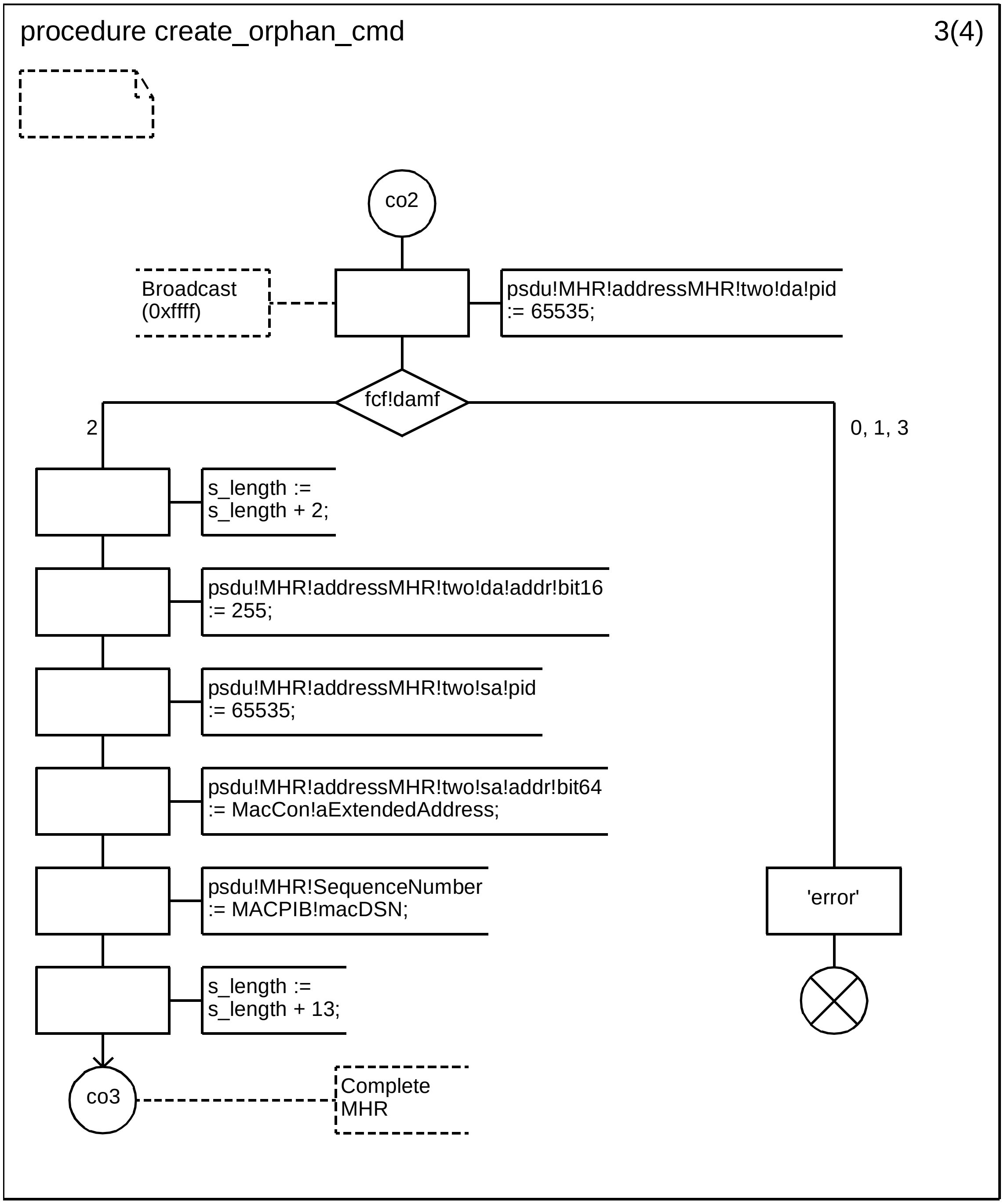
D.3.1.154.52 Procedure create_orphan_cmd (2)

.


procedure create_orphan_cmd
2(4)
co1
fcf!rsrvd :=0;
MacPIB!macShortAddress
else
254
'error'
Ophan notification can not be sent with 64bit destination address
fcf!damf := 2;
fcf!damf := 3;
Destination Addressing Mode field
fcf!rsrvd2 :=0;
fcf!samf := 3;
Source Addressing Mode field
psdu!MHR!FrameControl := fcf;
s_length :=2;
co2

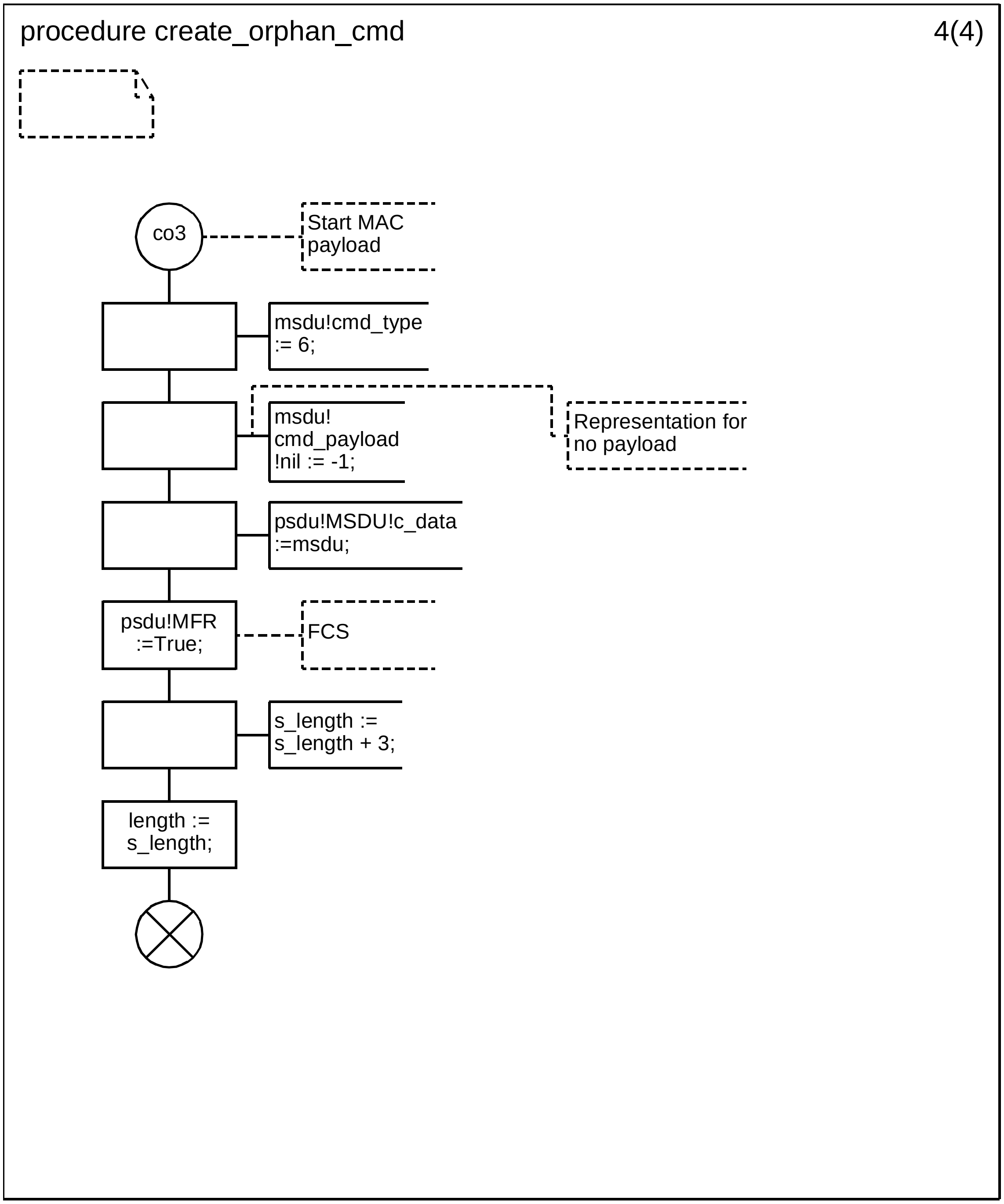
D.3.1.154.53 Procedure create_orphan_cmd (3)

.


procedure create_orphan_cmd
3(4)
co2
Broadcast
(0xffff)
psdu!MHR!addressMHR!two!da!pid
:= 65535;
fcf!damf
2
0, 1, 3
s_length :=
s_length + 2;
psdu!MHR!addressMHR!two!da!addr!bit16
:= 255;
psdu!MHR!addressMHR!two!sa!pid
:= 65535;
psdu!MHR!addressMHR!two!sa!addr!bit64
:= MacCon!aExtendedAddress;
psdu!MHR!SequenceNumber
:= MACPIB!macDSN;
'error'
s_length :=
s_length + 13;
co3
Complete
MHR

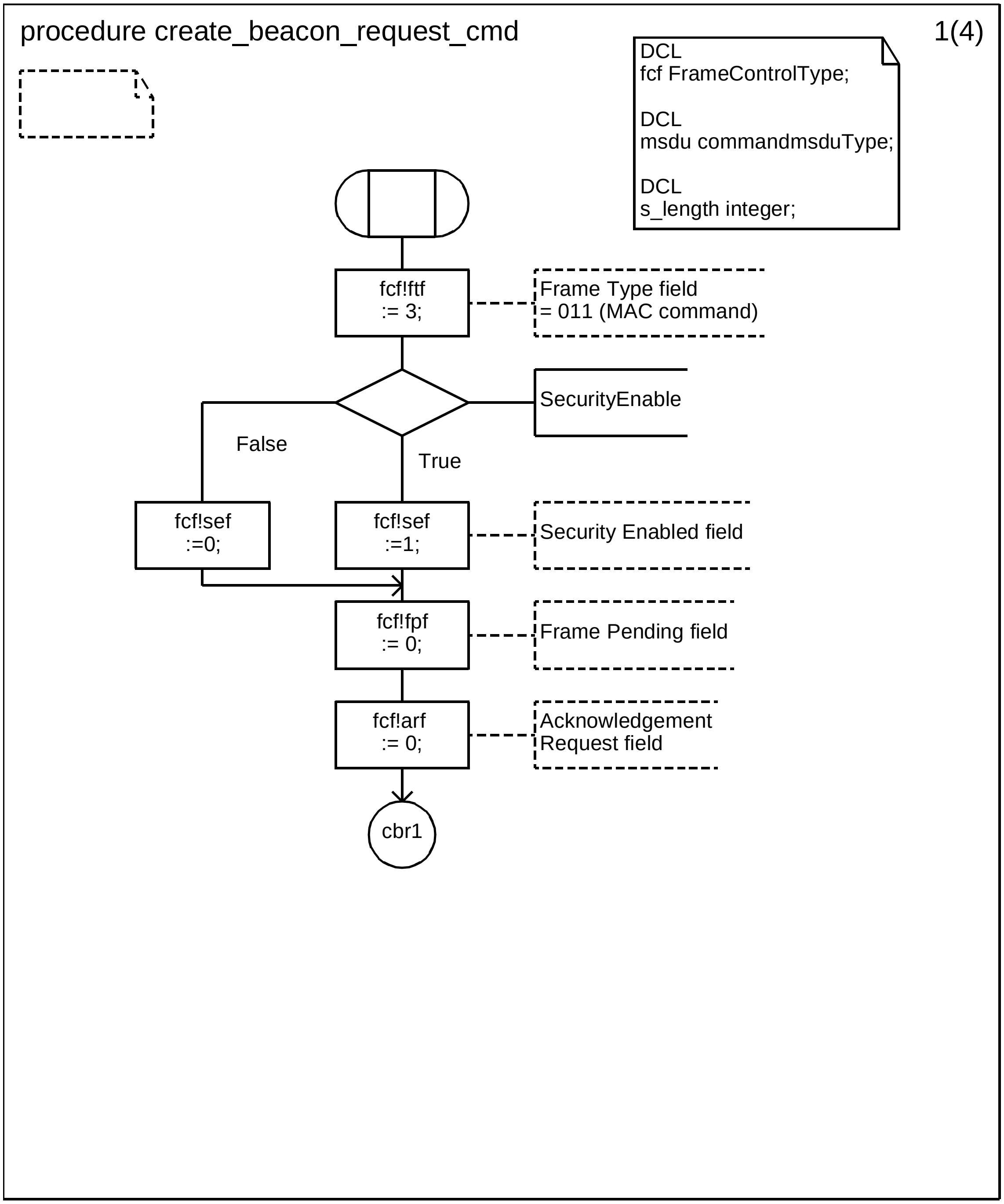
D.3.1.154.54 Procedure create_orphan_cmd (4)

.


procedure create_orphan_cmd
4(4)
co3
Start MAC payload
msdu!cmd_type := 6;
msdu! cmd_payload !nil := -1;
Representation for no payload
psdu!MSDU!c_data :=msdu;
psdu!MFR :=True;
FCS
s_length := s_length + 3;
length := s_length;

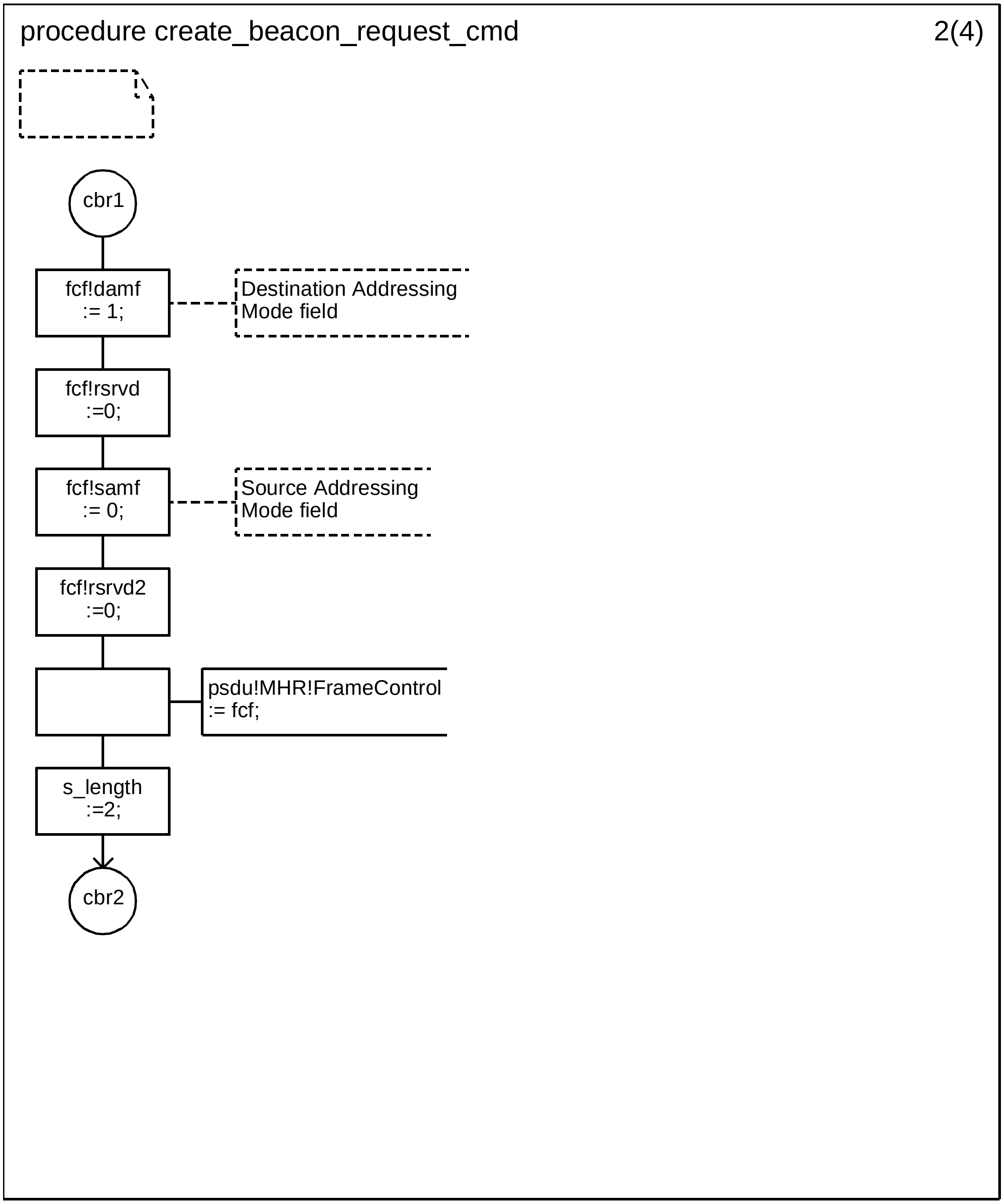
D.3.1.154.55 Procedure create_beacon_request_cmd (1)

.



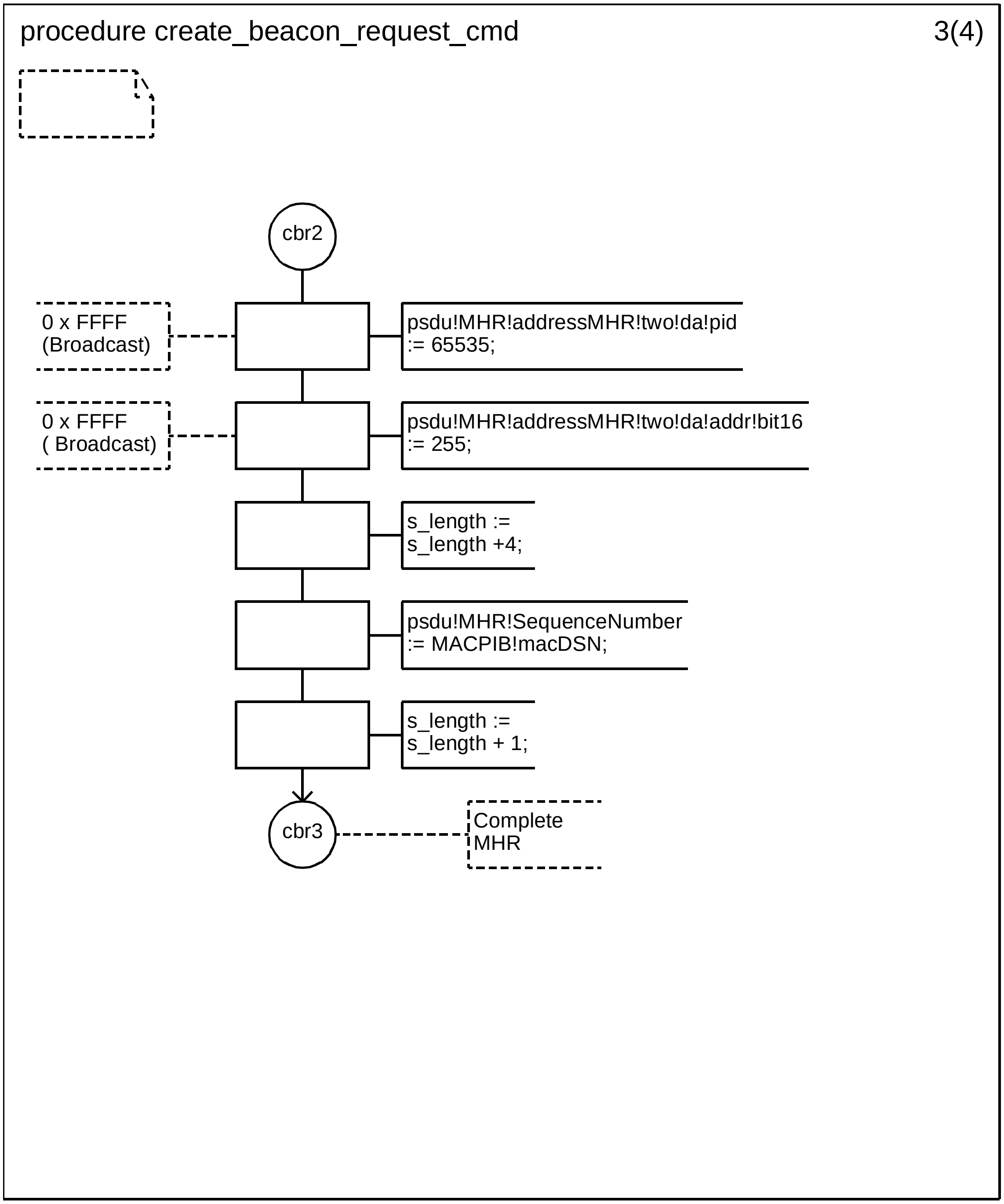
D.3.1.154.56 Procedure create_beacon_request_cmd (2)

.



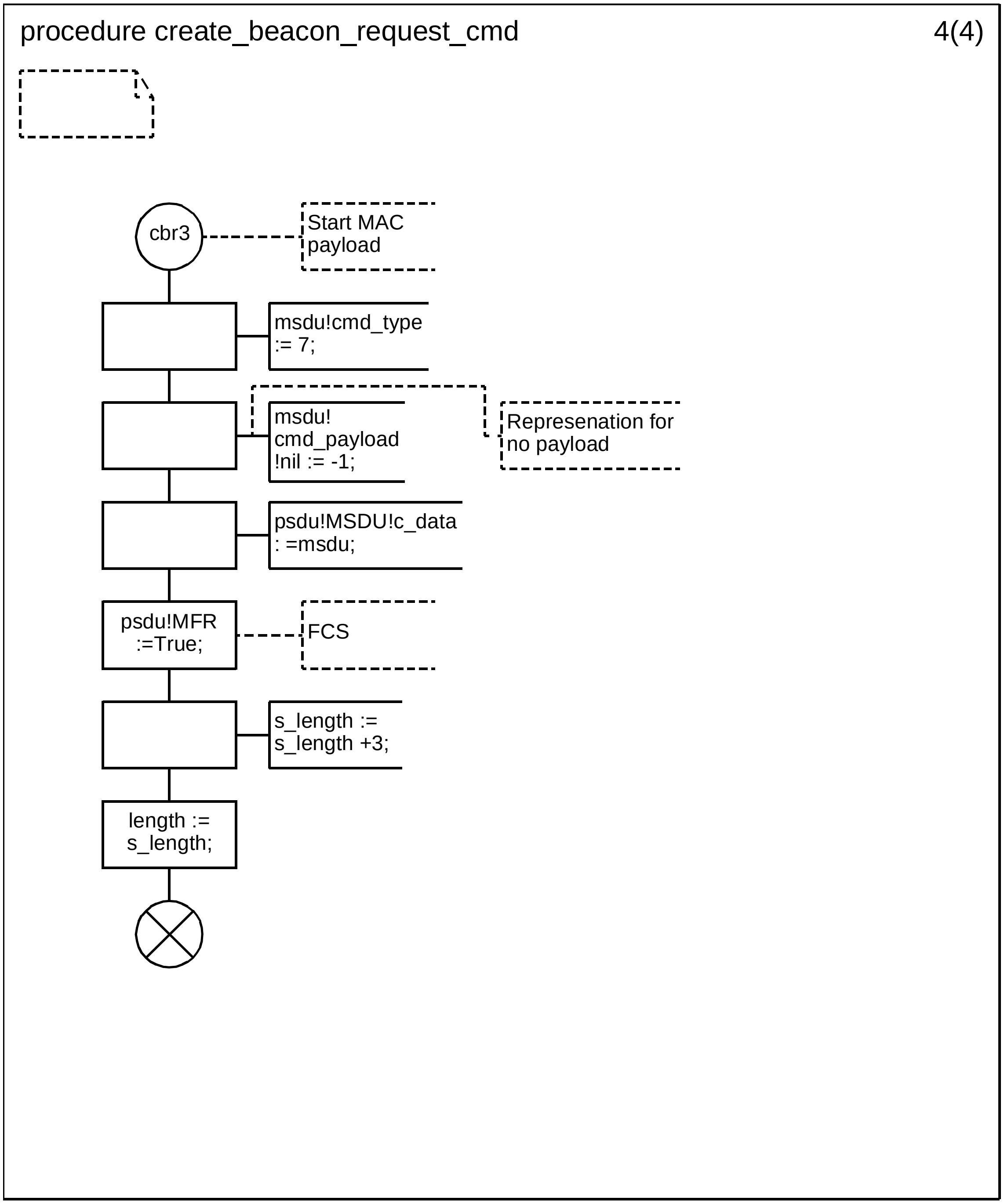
D.3.1.154.57 Procedure create_beacon_request_cmd (3)

.



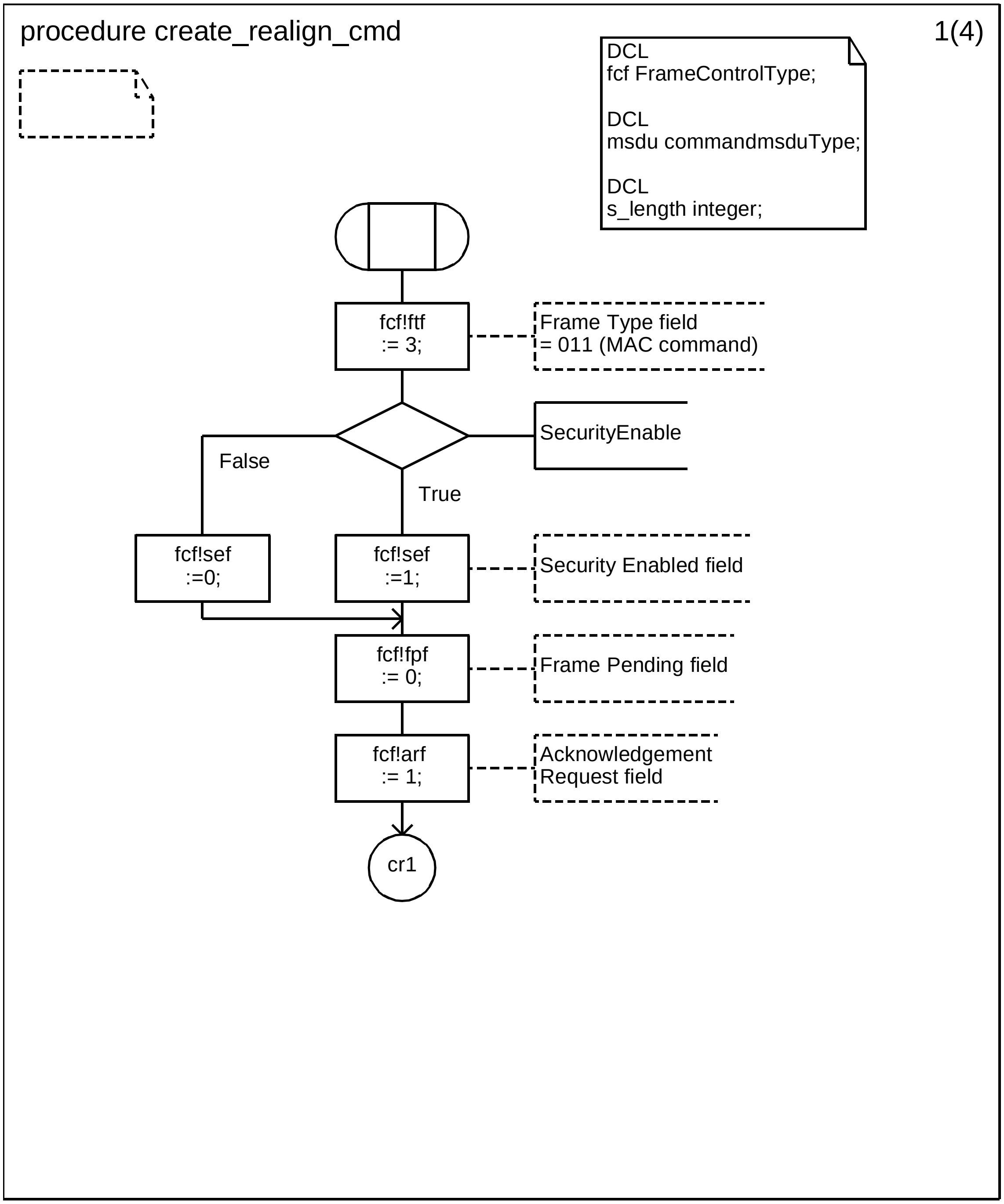
D.3.1.154.58 Procedure create_beacon_request_cmd (4)

.


procedure create_beacon_request_cmd
4(4)
cbr3
Start MAC payload
msdu!cmd_type := 7;
msdu! cmd_payload !nil := -1;
Represenation for no payload
psdu!MSDU!c_data : =msdu;
psdu!MFR :=True;
FCS
s_length := s_length +3;
length := s_length;

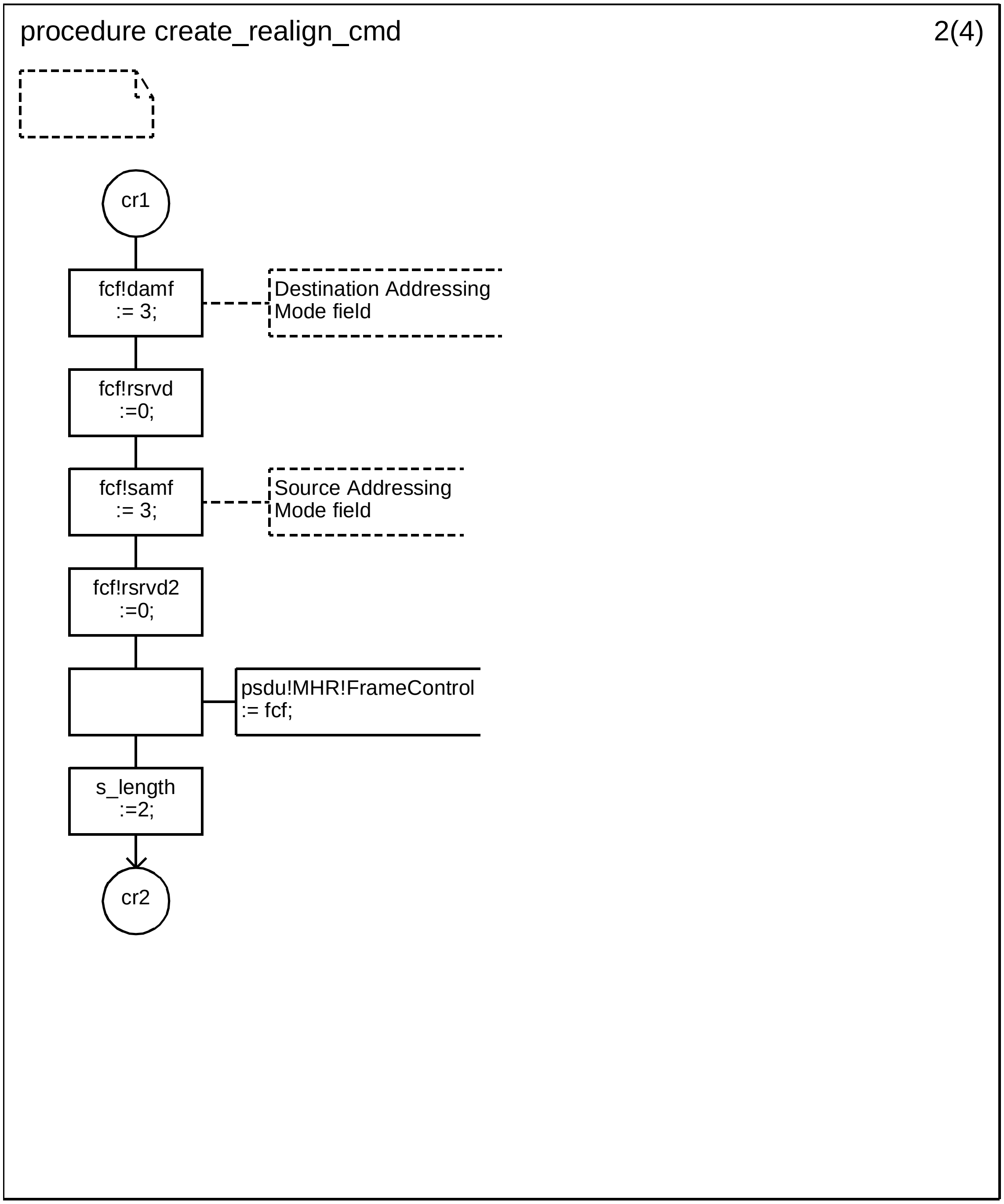
D.3.1.154.59 Procedure create_realign_cmd (1)

.



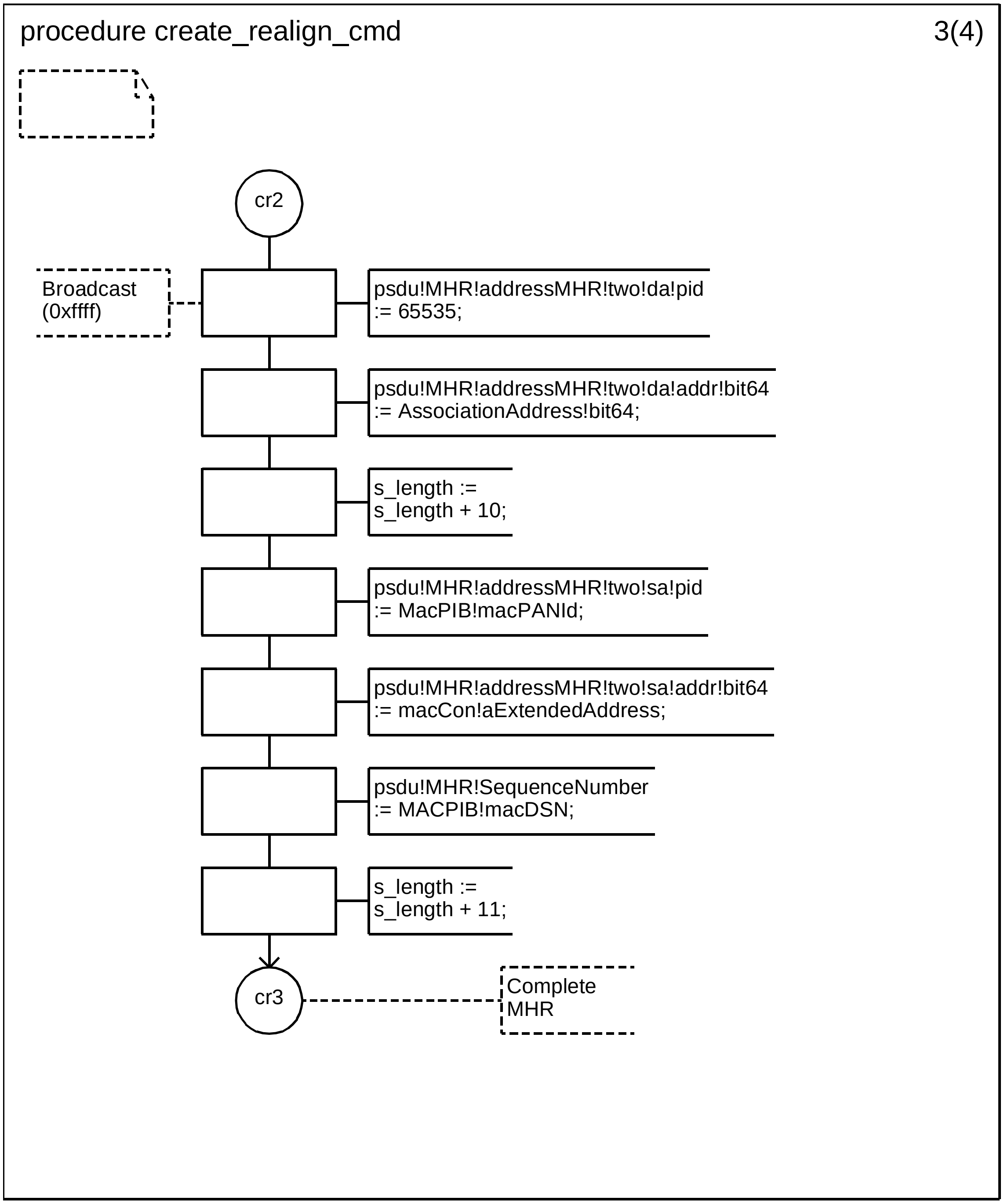
D.3.1.154.60 Procedure create_realign_cmd (2)

.



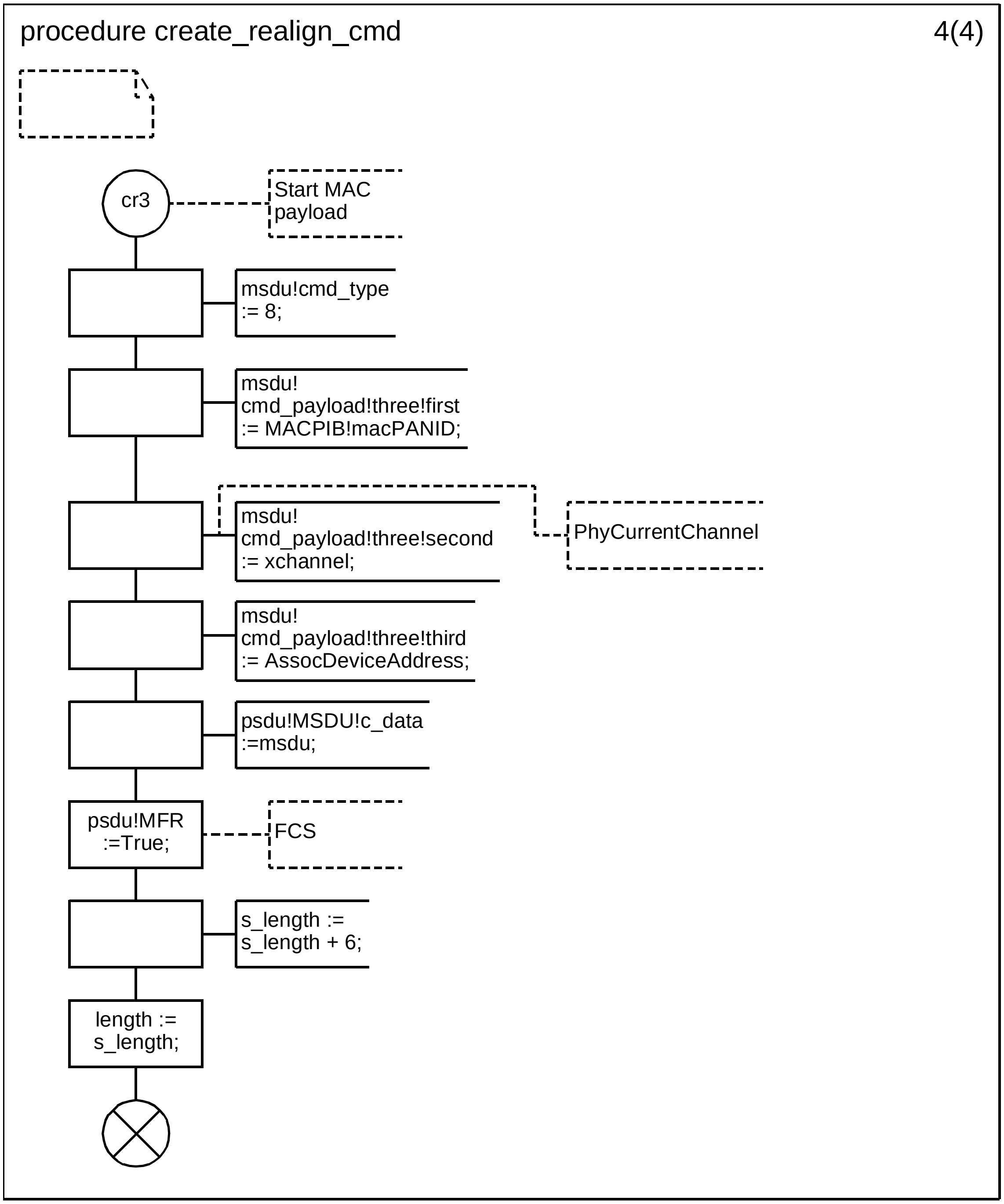
D.3.1.154.61 Procedure create_realign_cmd (3)

.


procedure create_realign_cmd
3(4)
cr2
Broadcast
(0xffff)
psdu!MHR!addressMHR!two!da!pid
:= 65535;
psdu!MHR!addressMHR!two!da!addr!bit64
:= AssociationAddress!bit64;
s_length :=
s_length + 10;
psdu!MHR!addressMHR!two!sa!pid
:= MacPIB!macPANId;
psdu!MHR!addressMHR!two!sa!addr!bit64
:= macCon!aExtendedAddress;
psdu!MHR!SequenceNumber
:= MACPIB!macDSN;
s_length :=
s_length + 11;
cr3
Complete
MHR

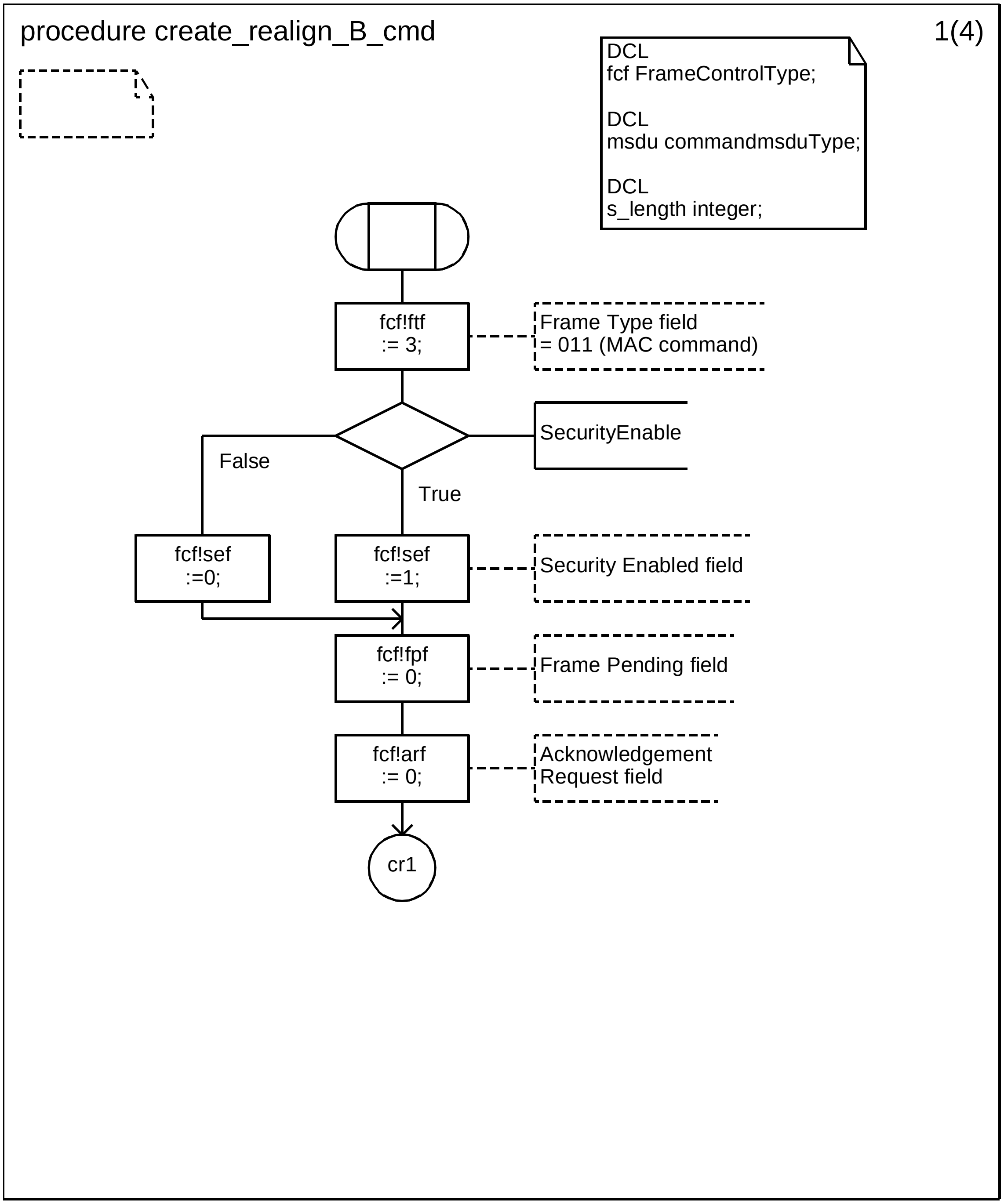
D.3.1.154.62 Procedure create_realign_cmd (4)

.


procedure create_realign_cmd
4(4)
cr3
Start MAC payload
msdu!cmd_type := 8;
msdu! cmd_payload!three!first := MACPIB!macPANID;
msdu! cmd_payload!three!second := xchannel;
PhyCurrentChannel
msdu! cmd_payload!three!third := AssocDeviceAddress;
psdu!MSDU!c_data :=msdu;
psdu!MFR :=True;
FCS
s_length := s_length + 6;
length := s_length;

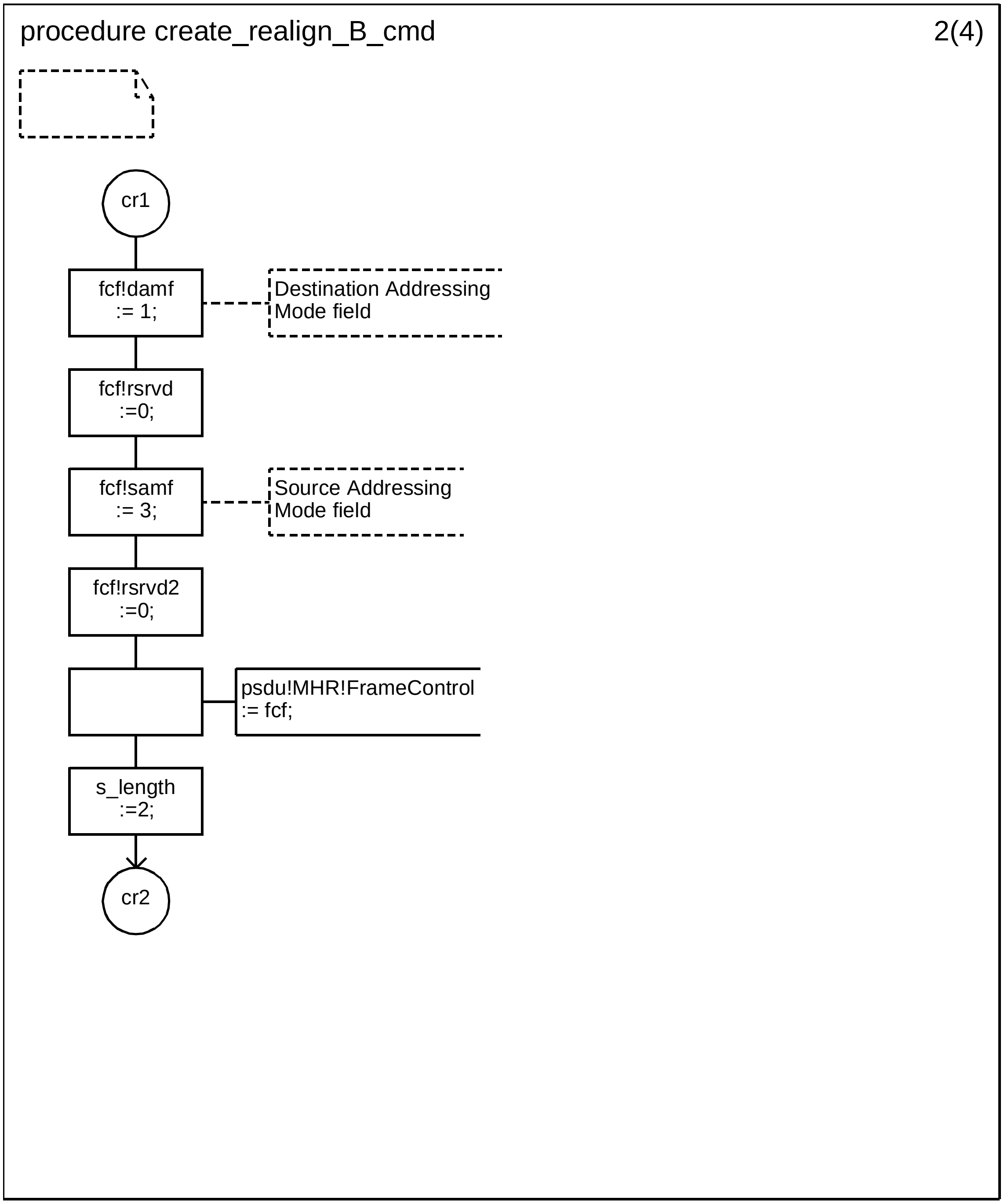
D.3.1.154.63 Procedure create_realign_B_cmd (1)

.



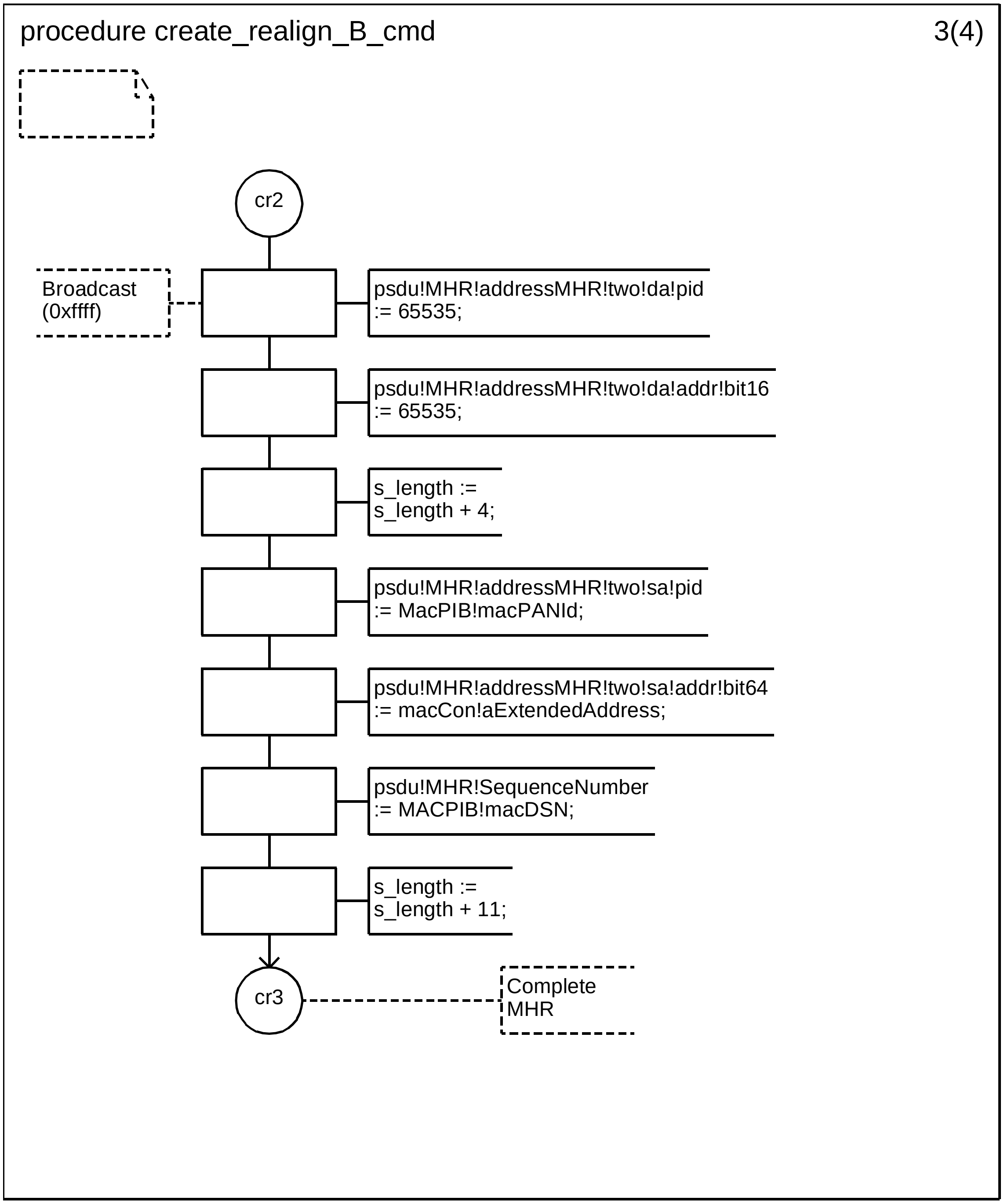
D.3.1.154.64 Procedure create_realign_B_cmd (2)

.



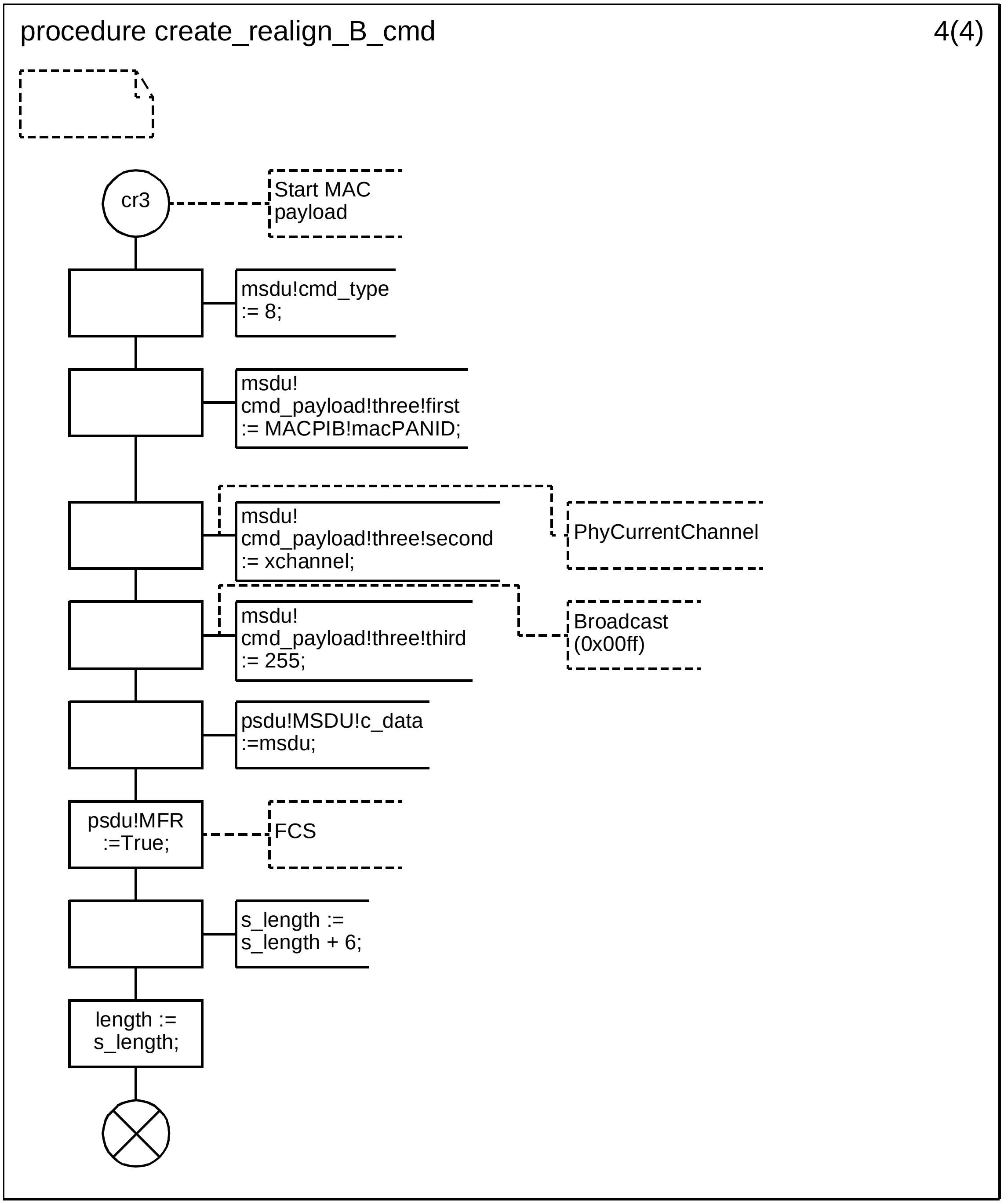
D.3.1.154.65 Procedure create_realign_B_cmd (3)

.


procedure create_realign_B_cmd
3(4)
cr2
Broadcast (0xffff)
psdu!MHR!addressMHR!two!da!pid := 65535;
psdu!MHR!addressMHR!two!da!addr!bit16 := 65535;
s_length := s_length + 4;
psdu!MHR!addressMHR!two!sa!pid := MacPIB!macPANId;
psdu!MHR!addressMHR!two!sa!addr!bit64 := macCon!aExtendedAddress;
psdu!MHR!SequenceNumber := MACPIB!macDSN;
s_length := s_length + 11;
cr3
Complete MHR

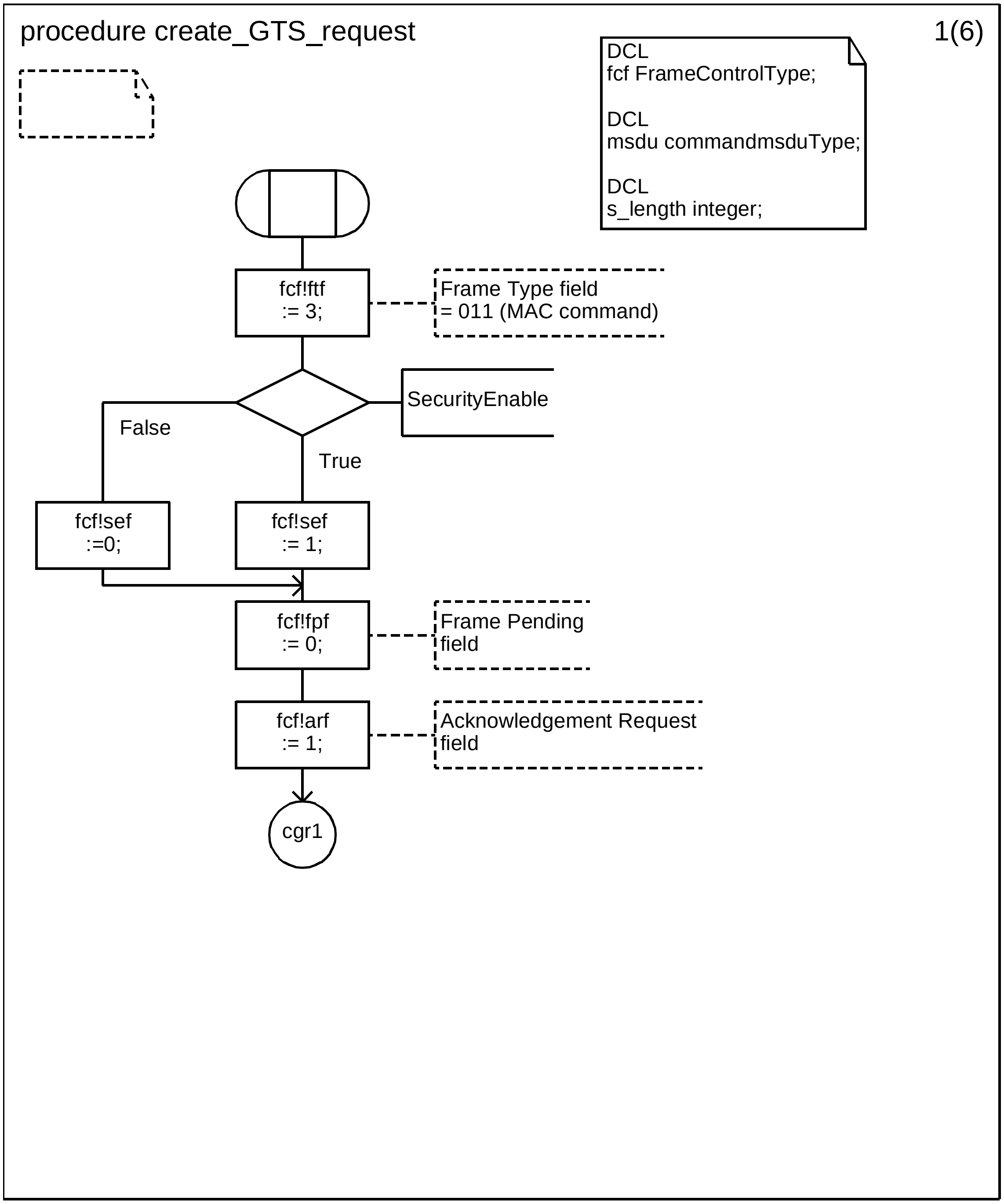
D.3.1.154.66 Procedure create_realign_B_cmd (4)

.


procedure create_realign_B_cmd
4(4)
cr3
Start MAC payload
msdu!cmd_type := 8;
msdu! cmd_payload!three!first := MACPIB!macPANID;
msdu! cmd_payload!three!second := xchannel;
PhyCurrentChannel
msdu! cmd_payload!three!third := 255;
Broadcast (0x00ff)
psdu!MSDU!c_data :=msdu;
psdu!MFR :=True;
FCS
s_length := s_length + 6;
length := s_length;

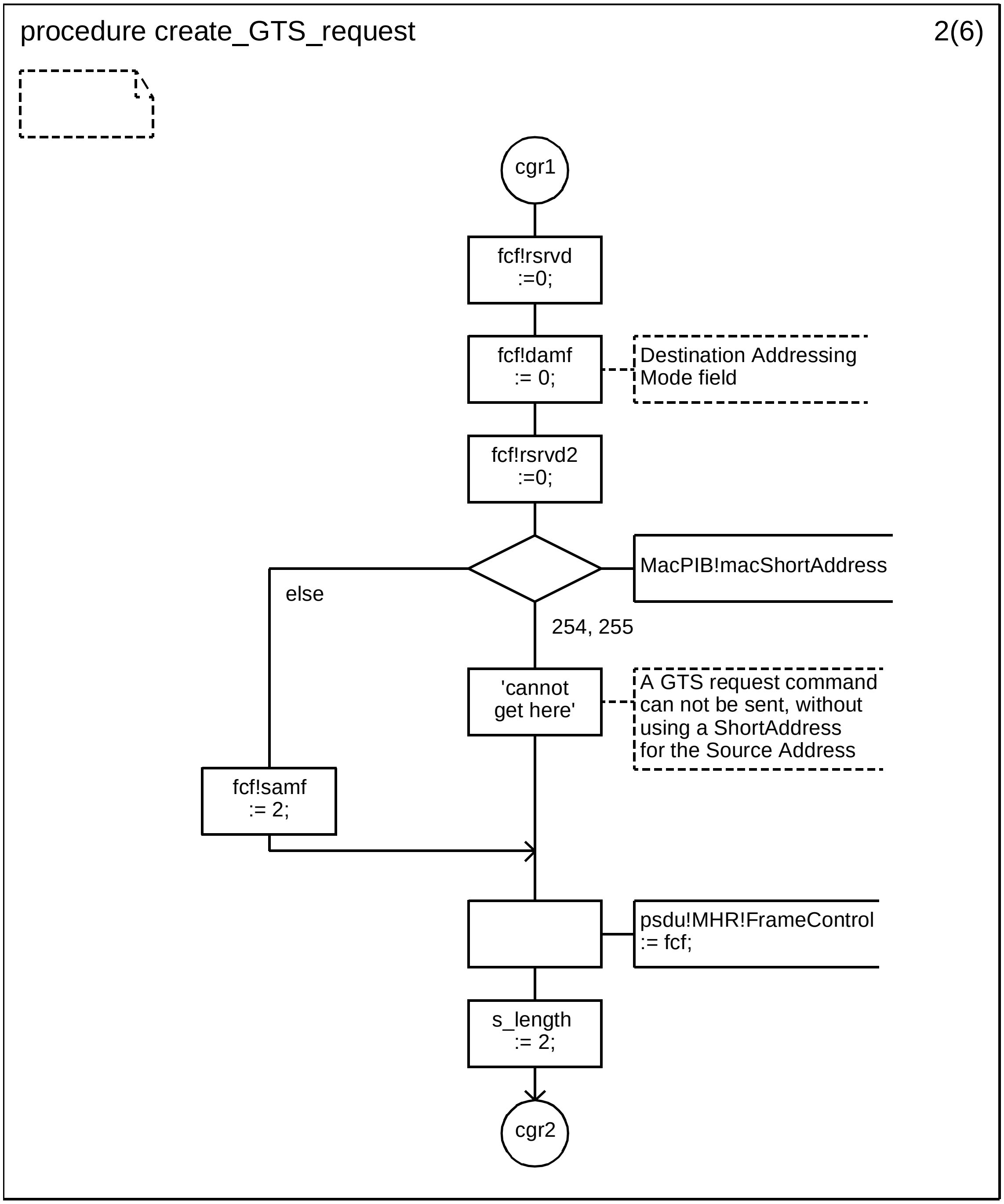
D.3.1.154.67 Procedure create_GTS_request (1)

.



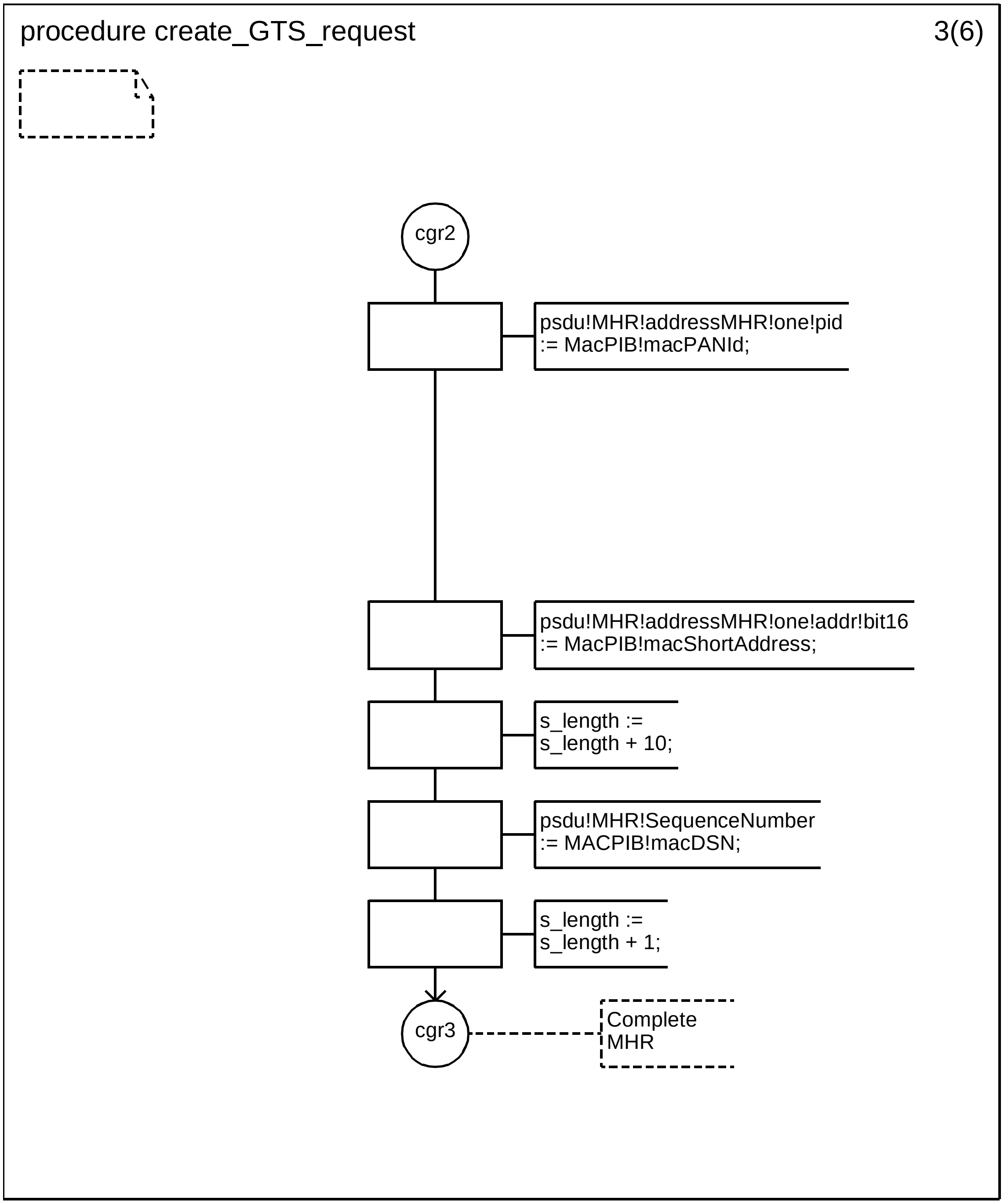
D.3.1.154.68 Procedure create_GTS_request (2)

.



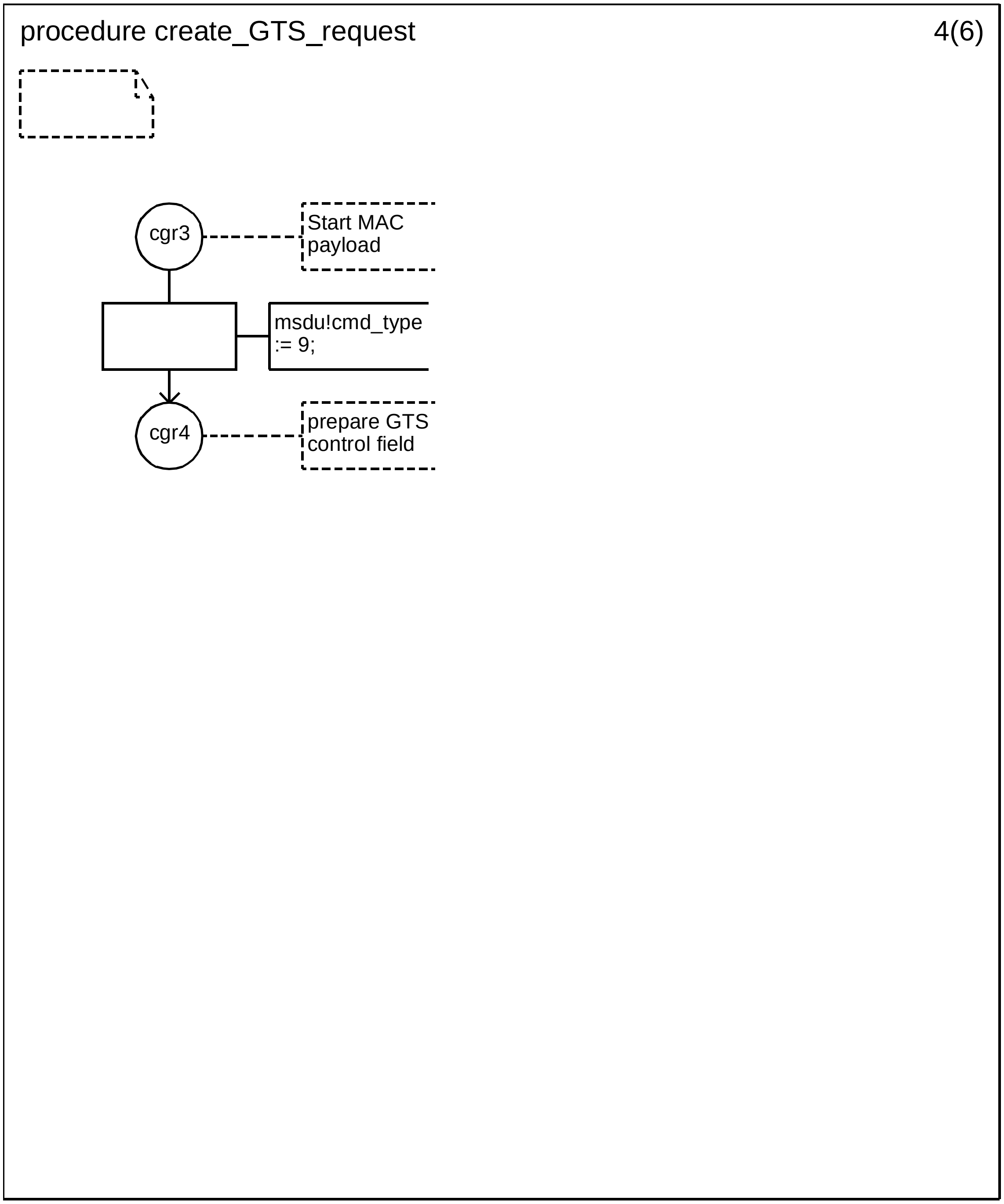
D.3.1.154.69 Procedure create_GTS_request (3)

.



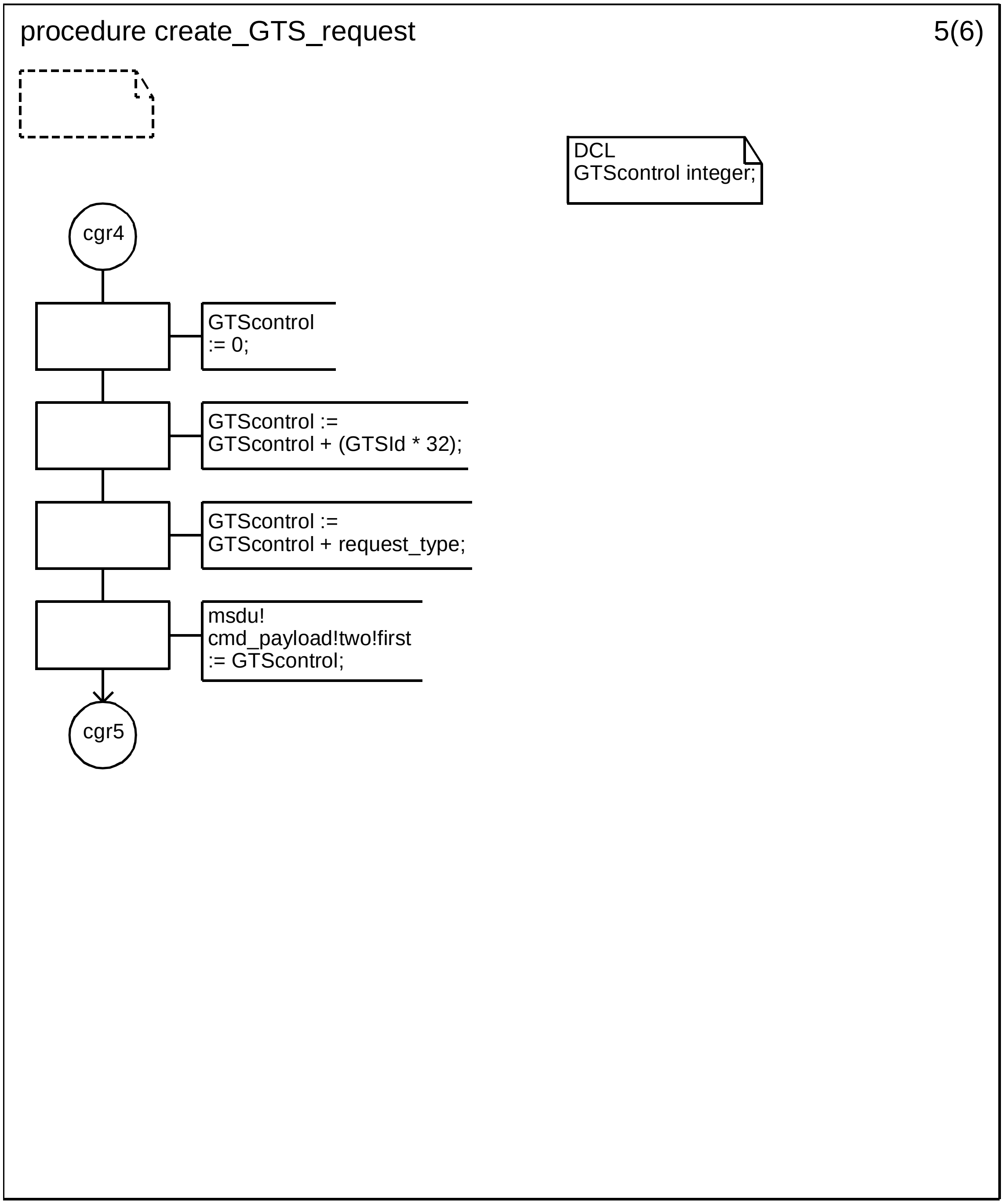
D.3.1.154.70 Procedure create_GTS_request (4)

.



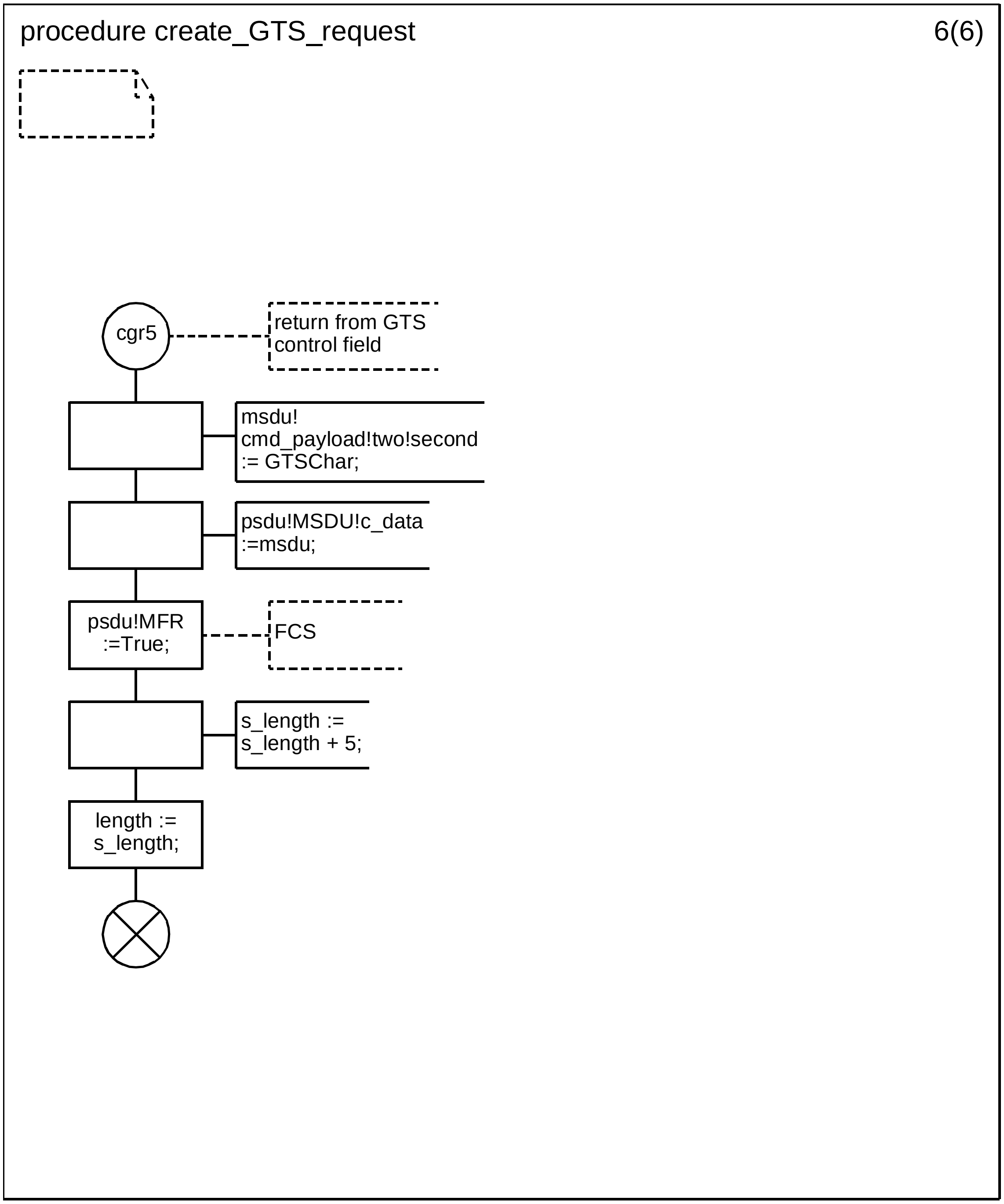
D.3.1.154.71 Procedure create_GTS_request (5)

.



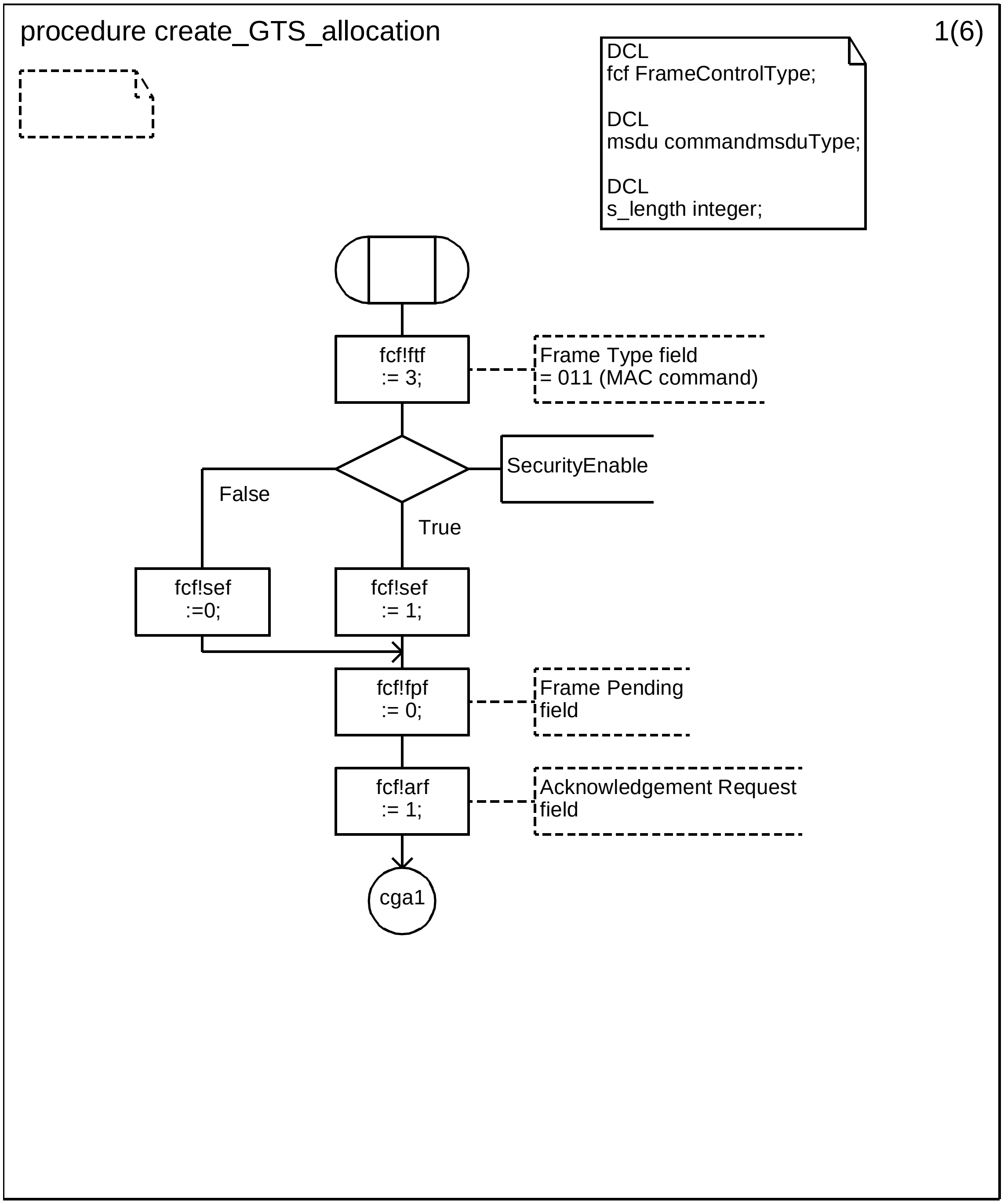
D.3.1.154.72 Procedure create_GTS_request (6)

.


procedure create_GTS_request
6(6)
cgr5
return from GTS control field
msdu! cmd_payload!two!second := GTSChar;
psdu!MSDU!c_data :=msdu;
psdu!MFR :=True;
FCS
s_length := s_length + 5;
length := s_length;

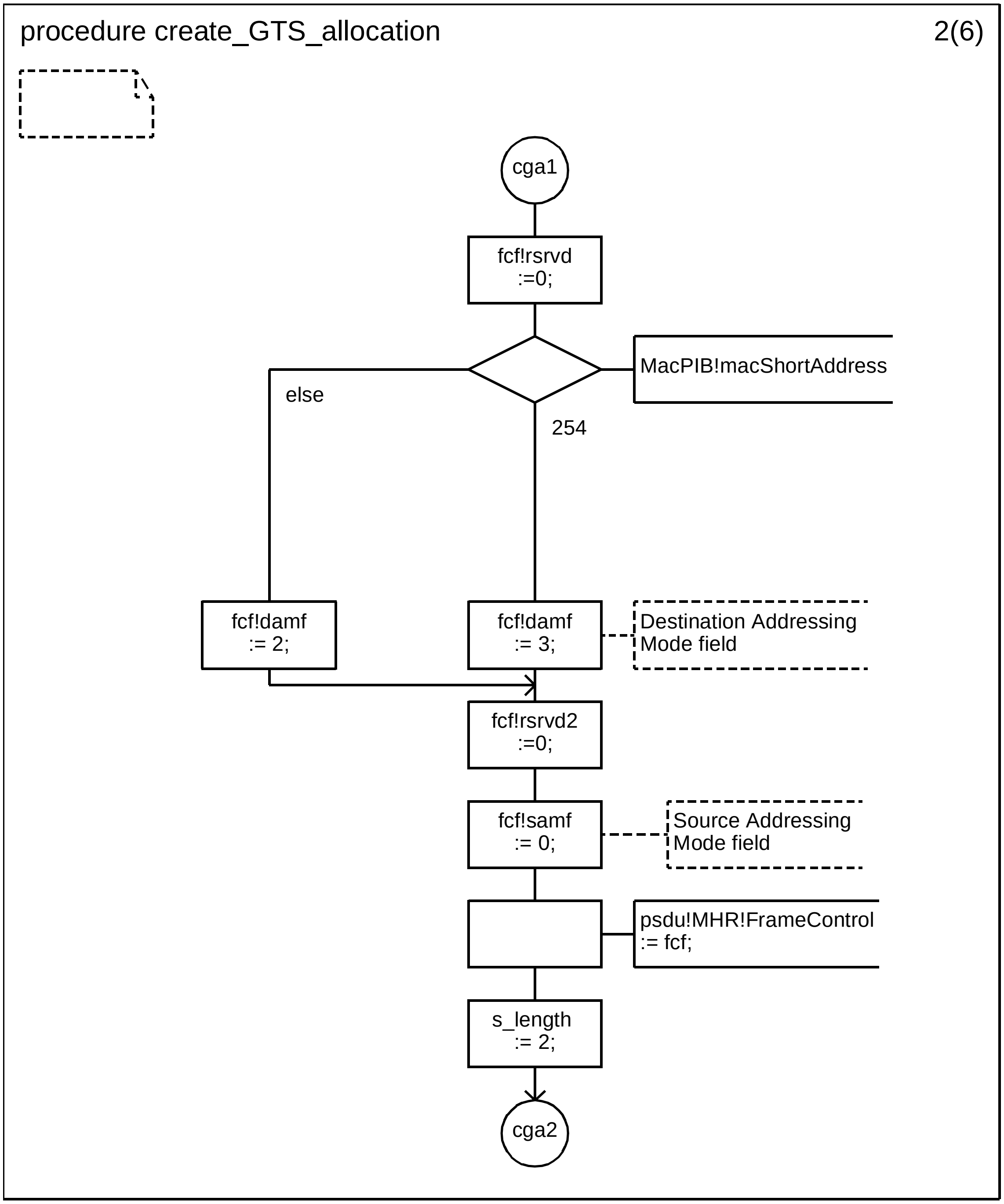
D.3.1.154.73 Procedure create_GTS_allocation (1)

.



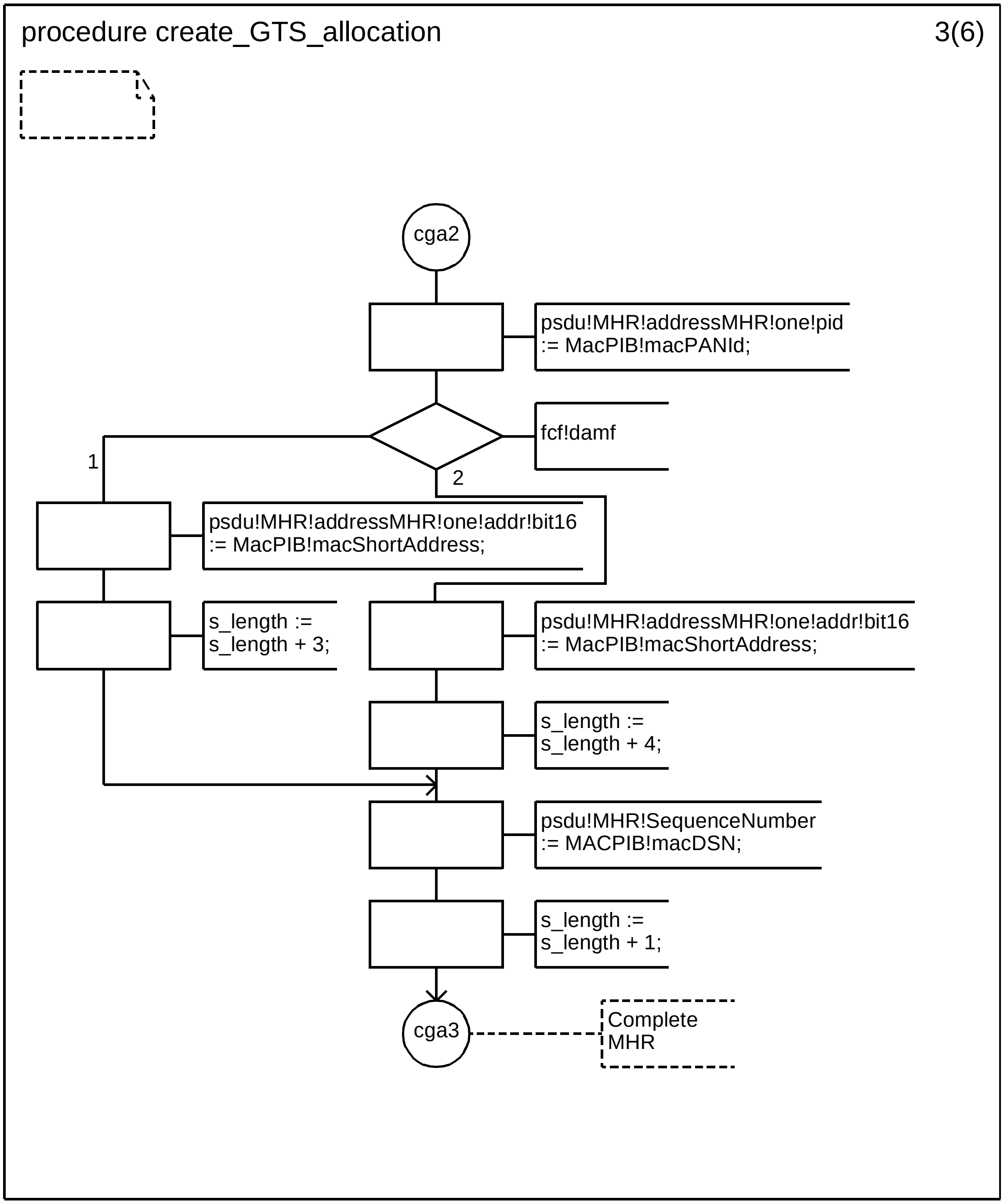
D.3.1.154.74 Procedure create_GTS_allocation (2)

.



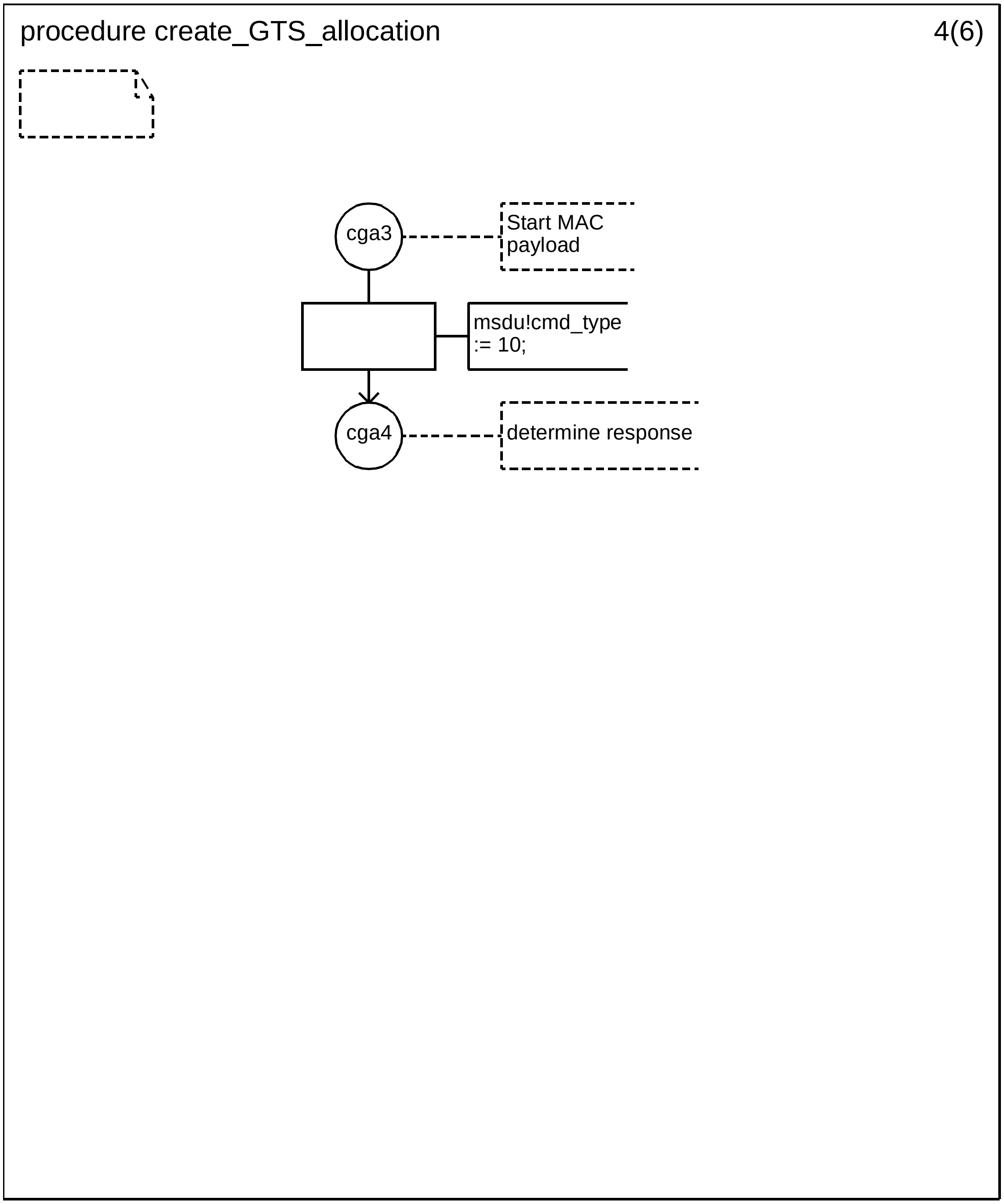
D.3.1.154.75 Procedure create_GTS_allocation (3)

.


procedure create_GTS_allocation
3(6)
cga2
psdu!MHR!addressMHR!one!pid
:= MacPIB!macPANId;
fcf!damf
1
2
psdu!MHR!addressMHR!one!addr!bit16
:= MacPIB!macShortAddress;
s_length :=
s_length + 3;
psdu!MHR!addressMHR!one!addr!bit16
:= MacPIB!macShortAddress;
s_length :=
s_length + 4;
psdu!MHR!SequenceNumber
:= MACPIB!macDSN;
s_length :=
s_length + 1;
cga3
Complete
MHR

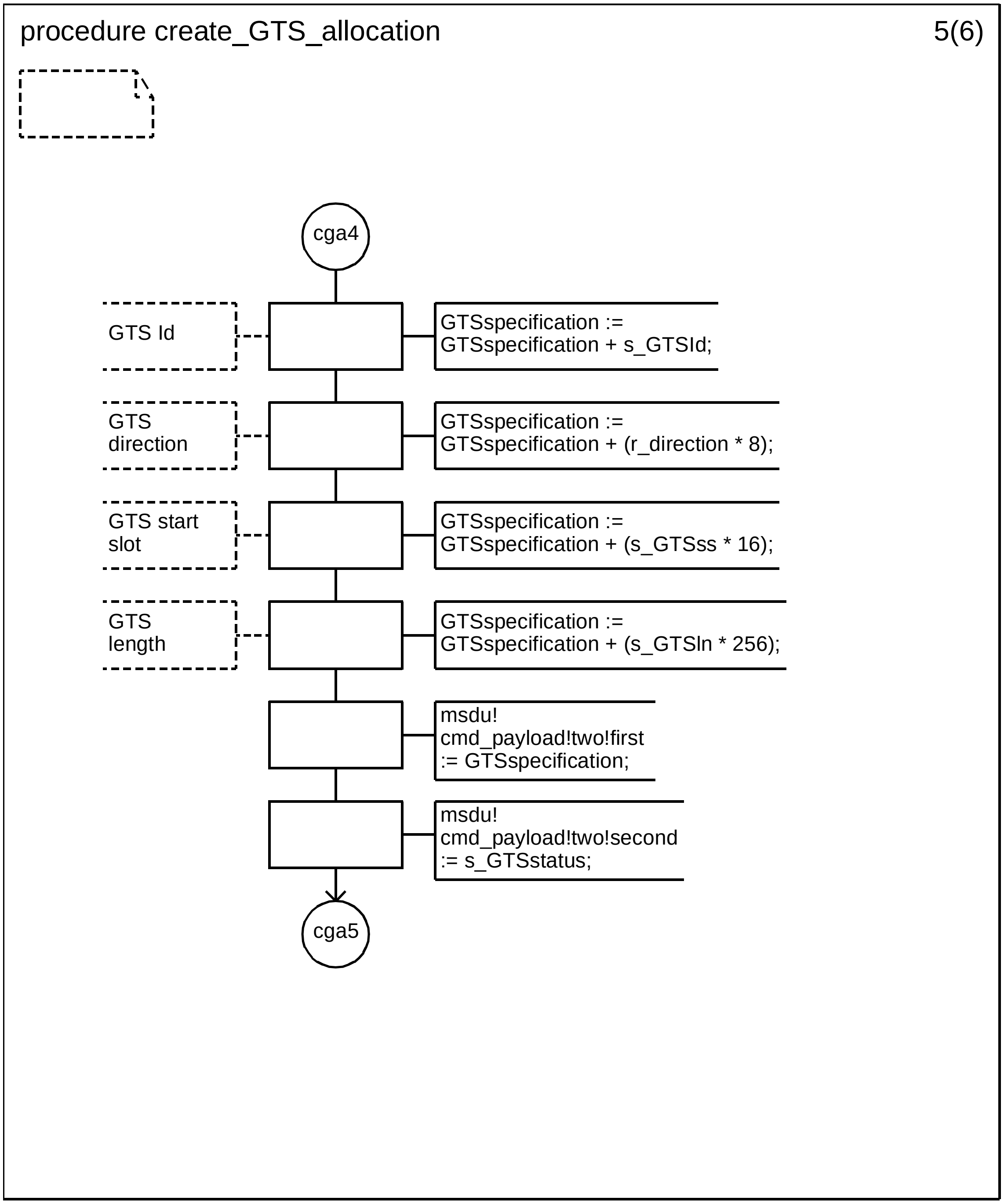
D.3.1.154.76 Procedure create_GTS_allocation (4)

.



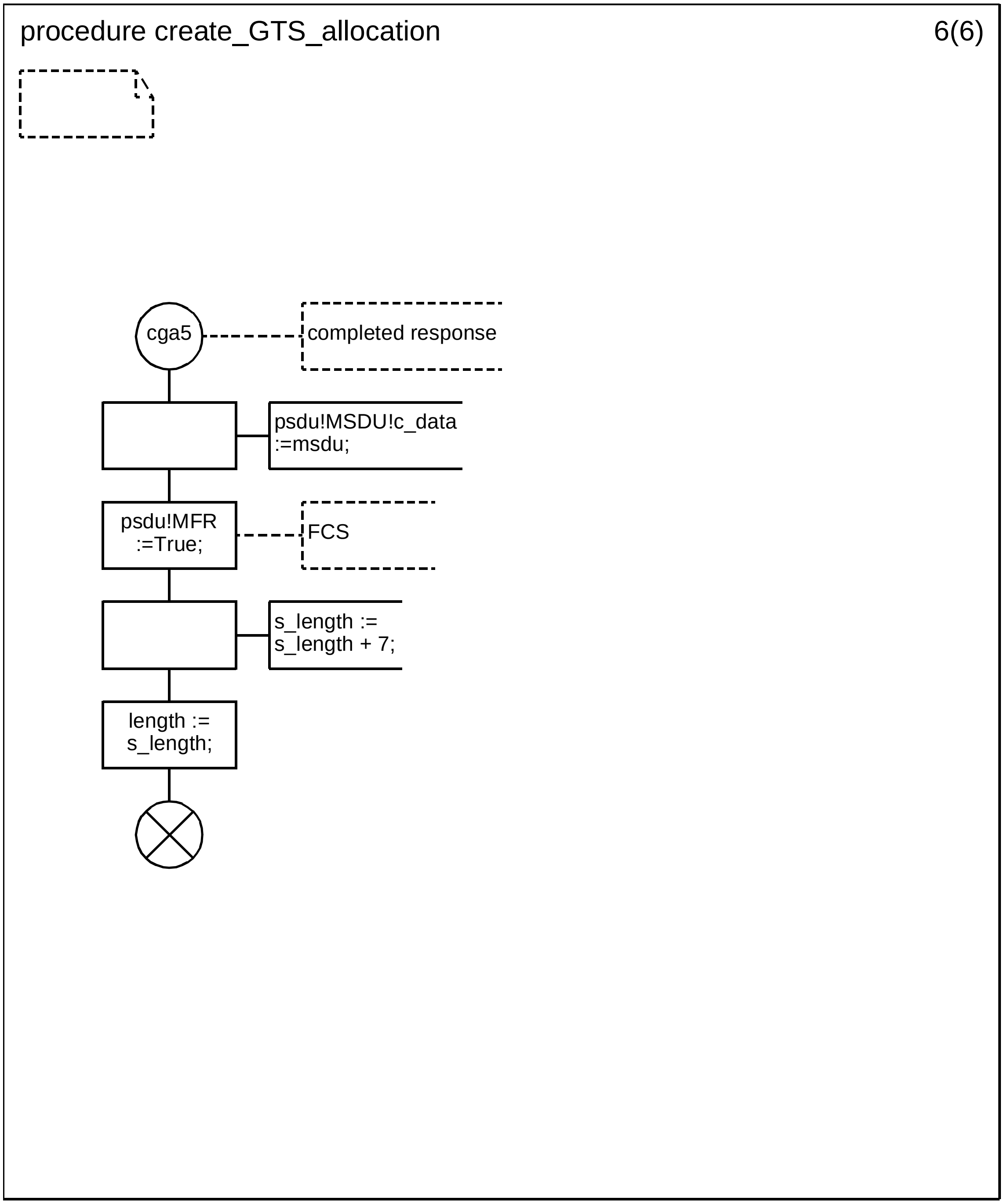
D.3.1.154.77 Procedure create_GTS_allocation (5)

.



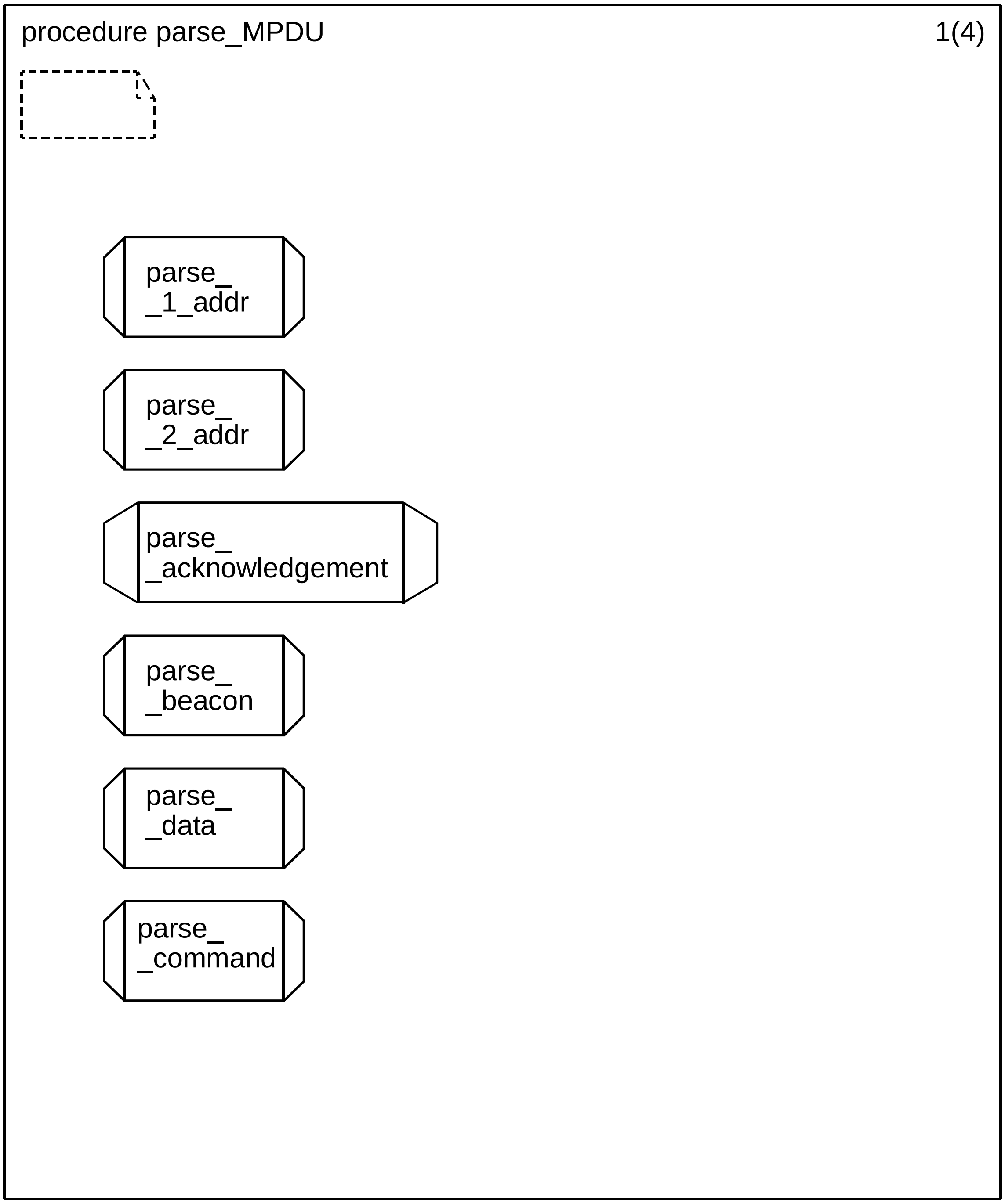
D.3.1.154.78 Procedure create_GTS_allocation (6)

.



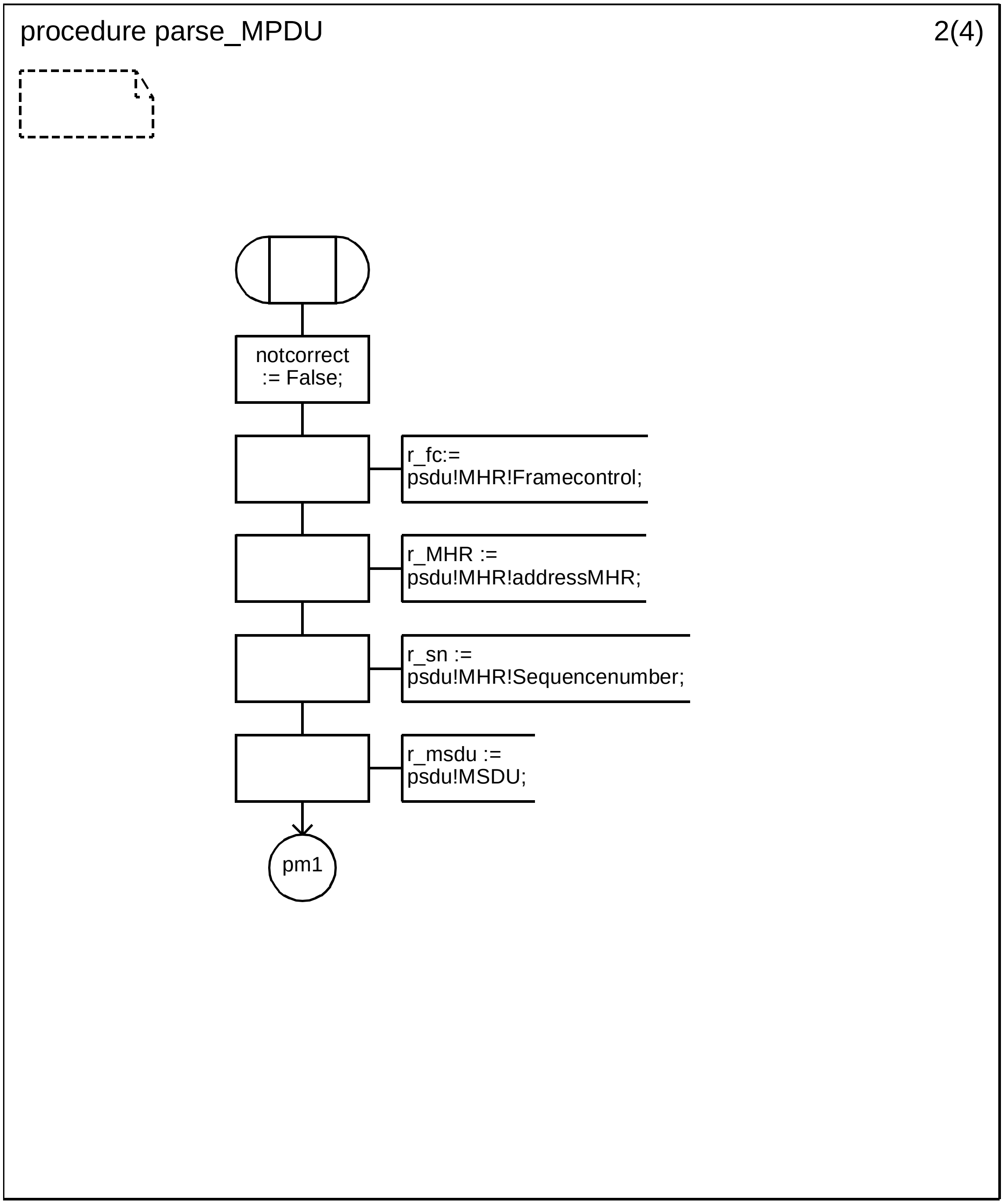
D.3.1.154.79 Procedure parse_MPDU (1)

.



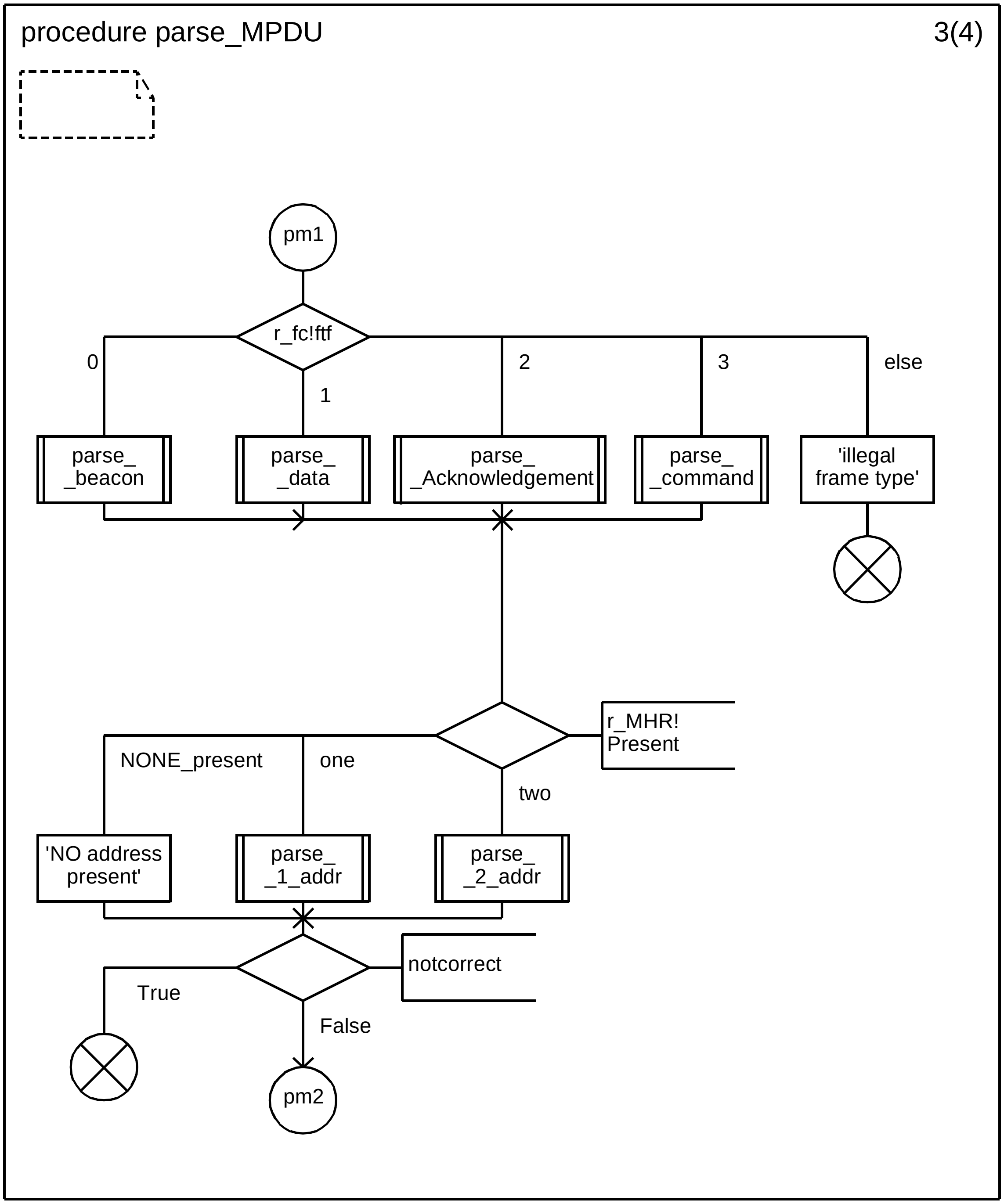
D.3.1.154.80 Procedure parse_MPDU (2)

.

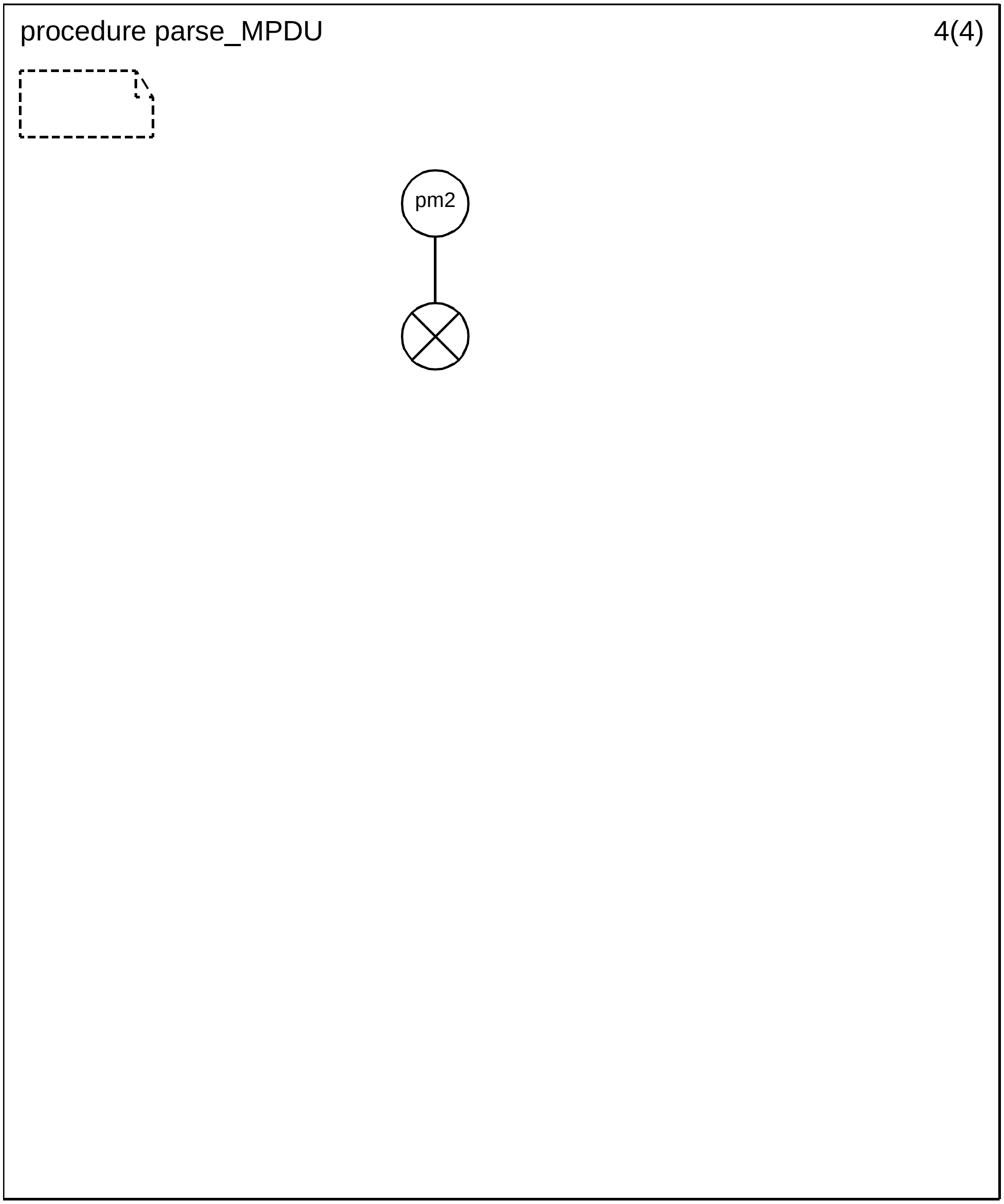


D.3.1.154.81 Procedure parse_MPDU (3)

.

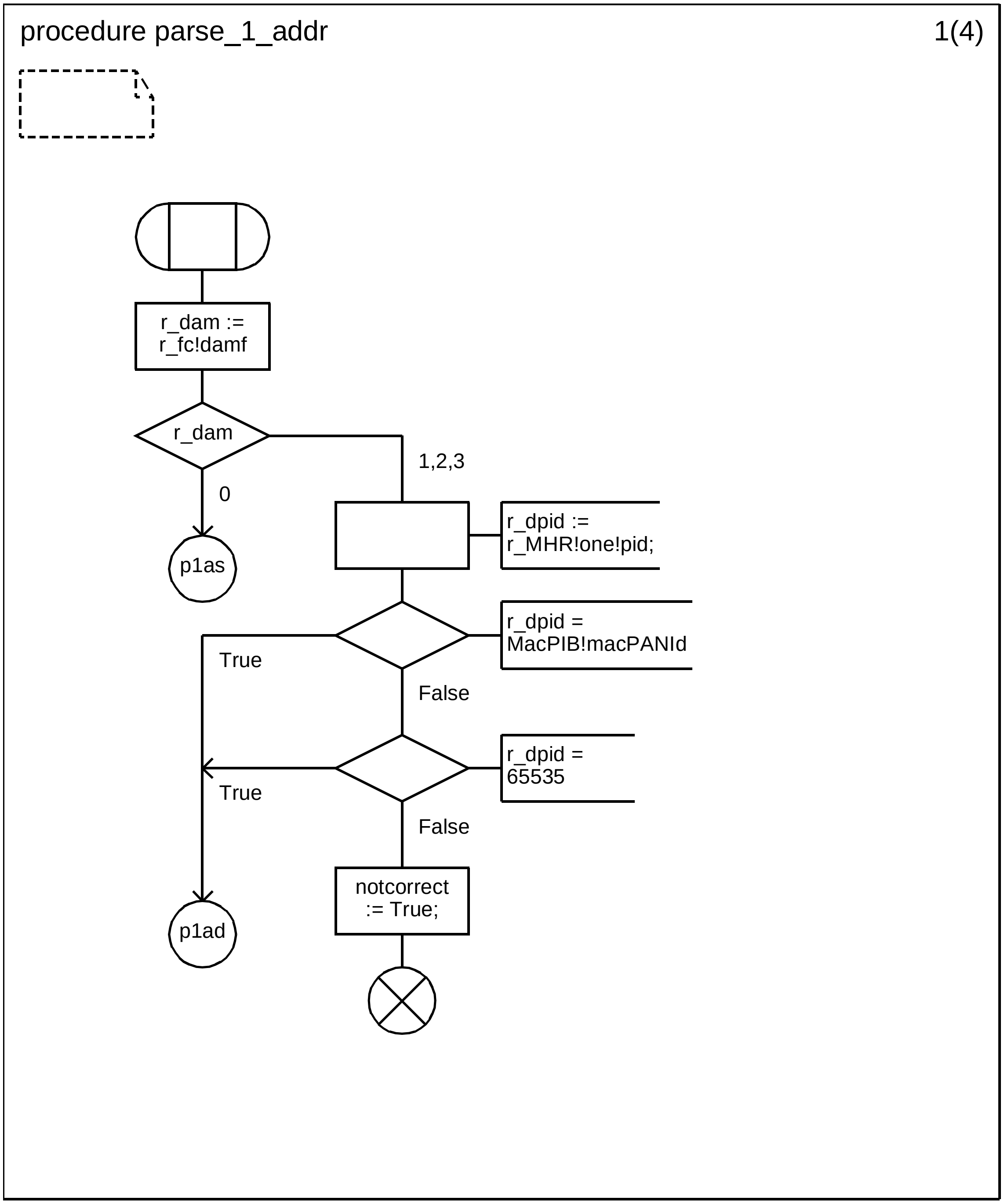


D.3.1.154.82 Procedure parse_MPDU (4)



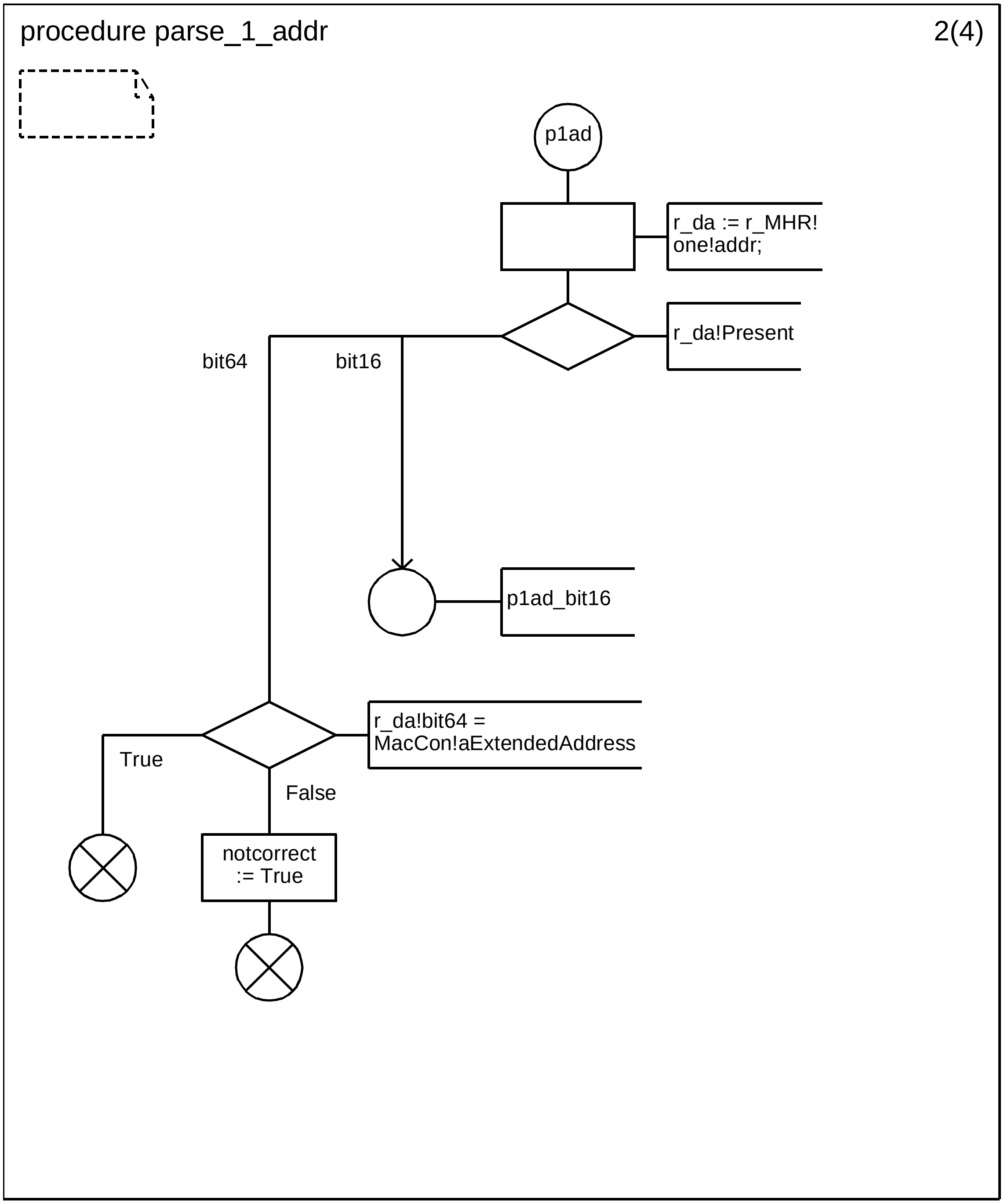
D.3.1.154.82.1 Procedure parse_1_addr (1)

.



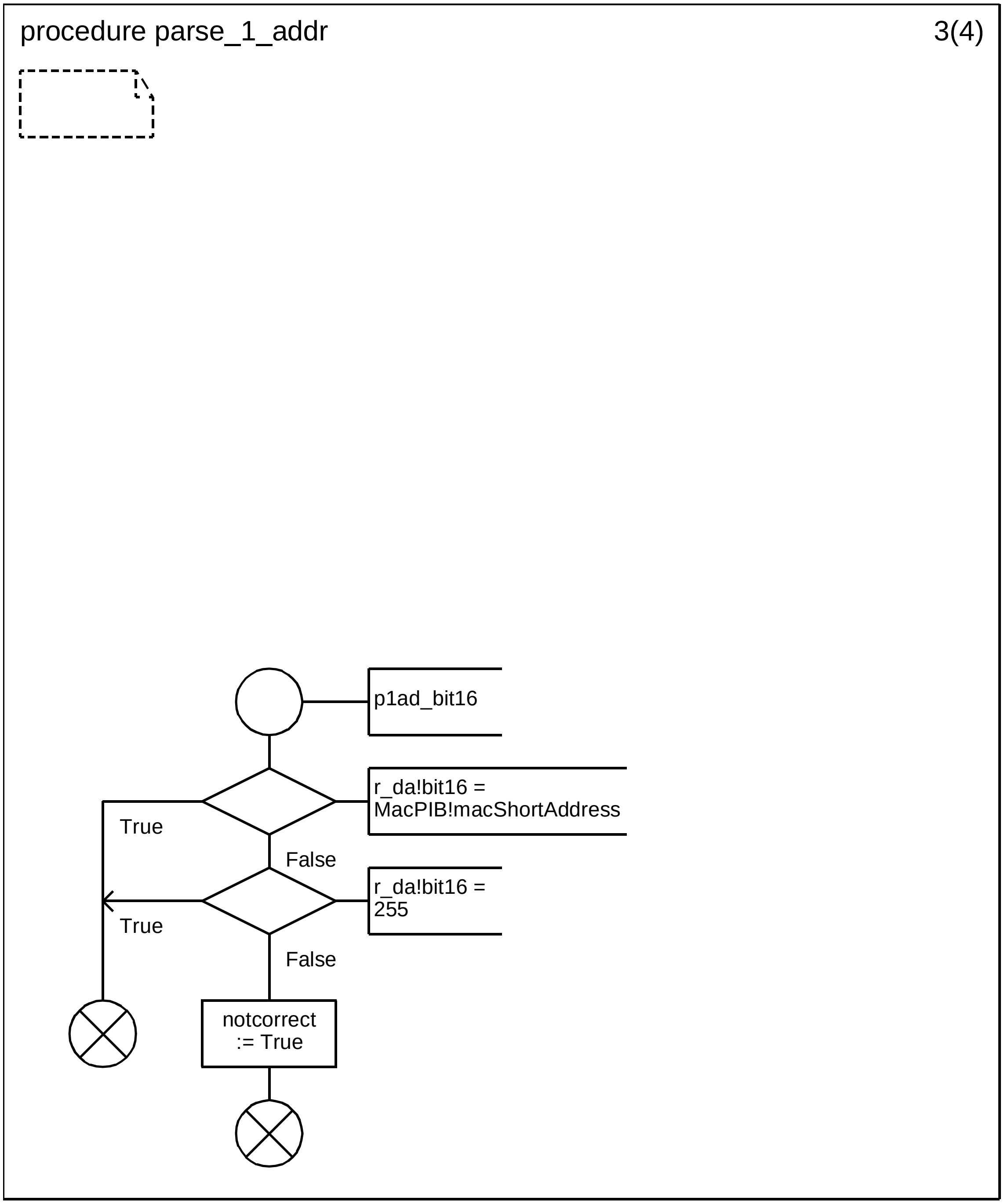
D.3.1.154.82.2 Procedure parse_1_addr (2)

.



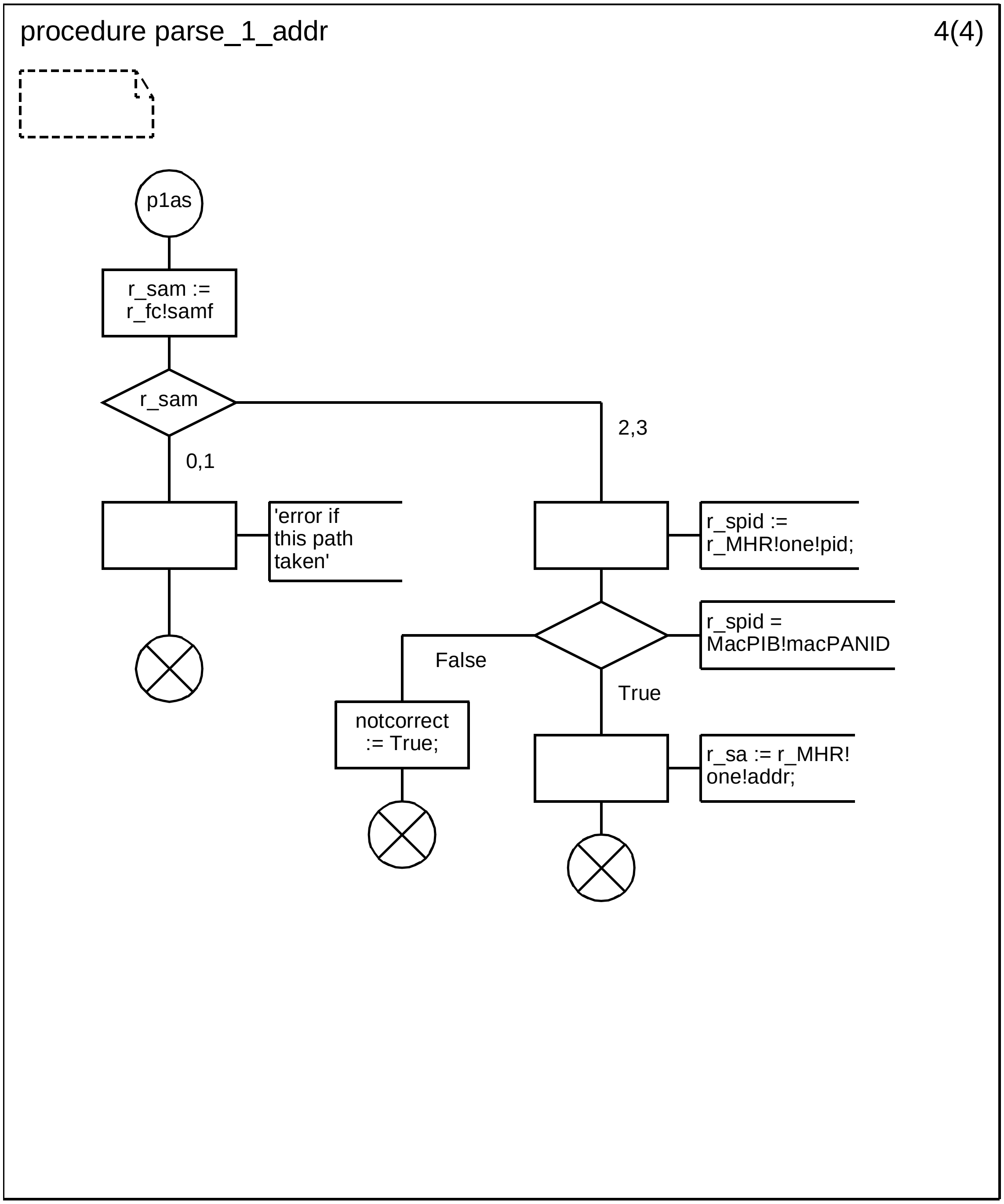
D.3.1.154.82.3 Procedure parse_1_addr (3)

.



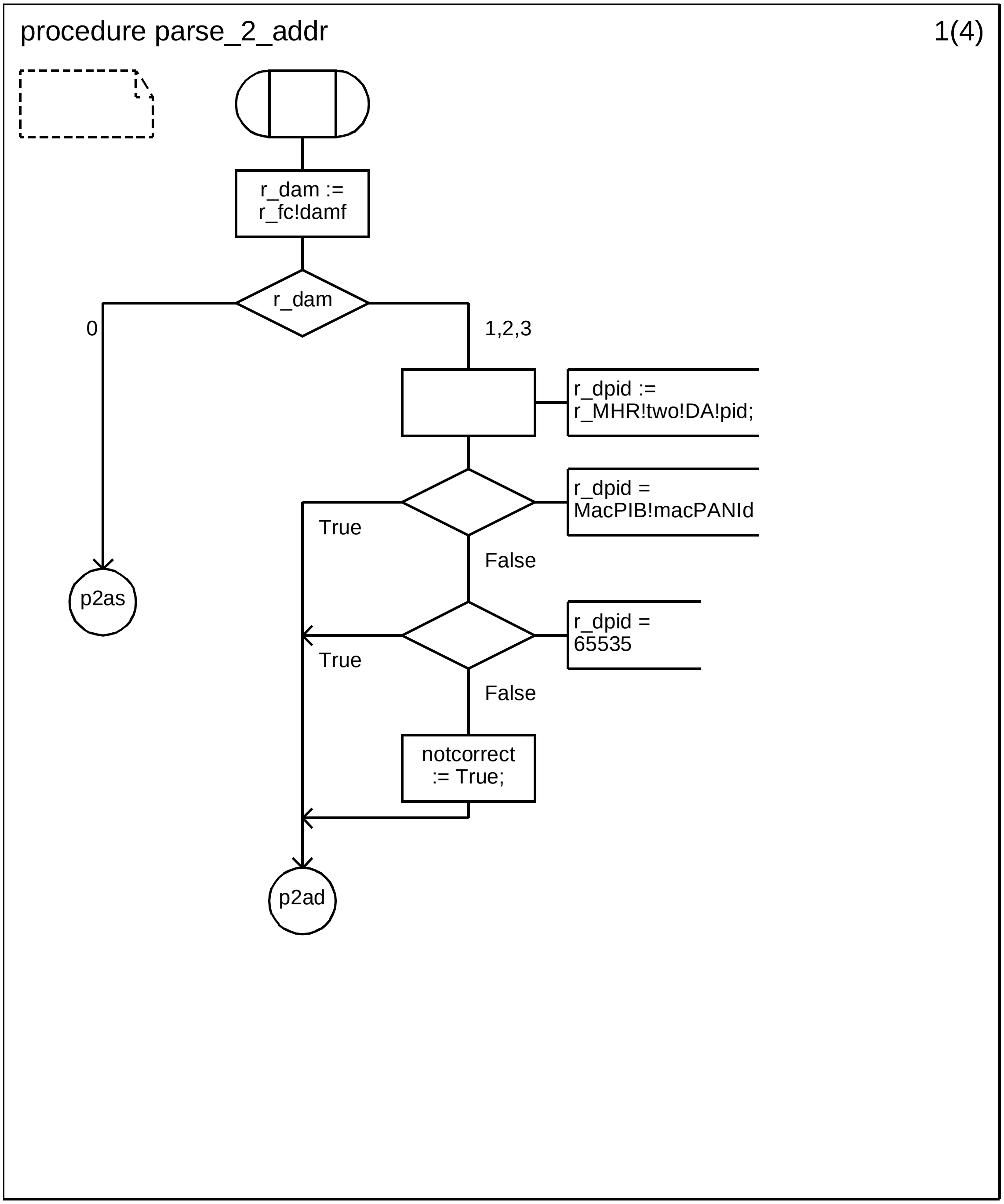
D.3.1.154.82.4 Procedure parse_1_addr (4)

.



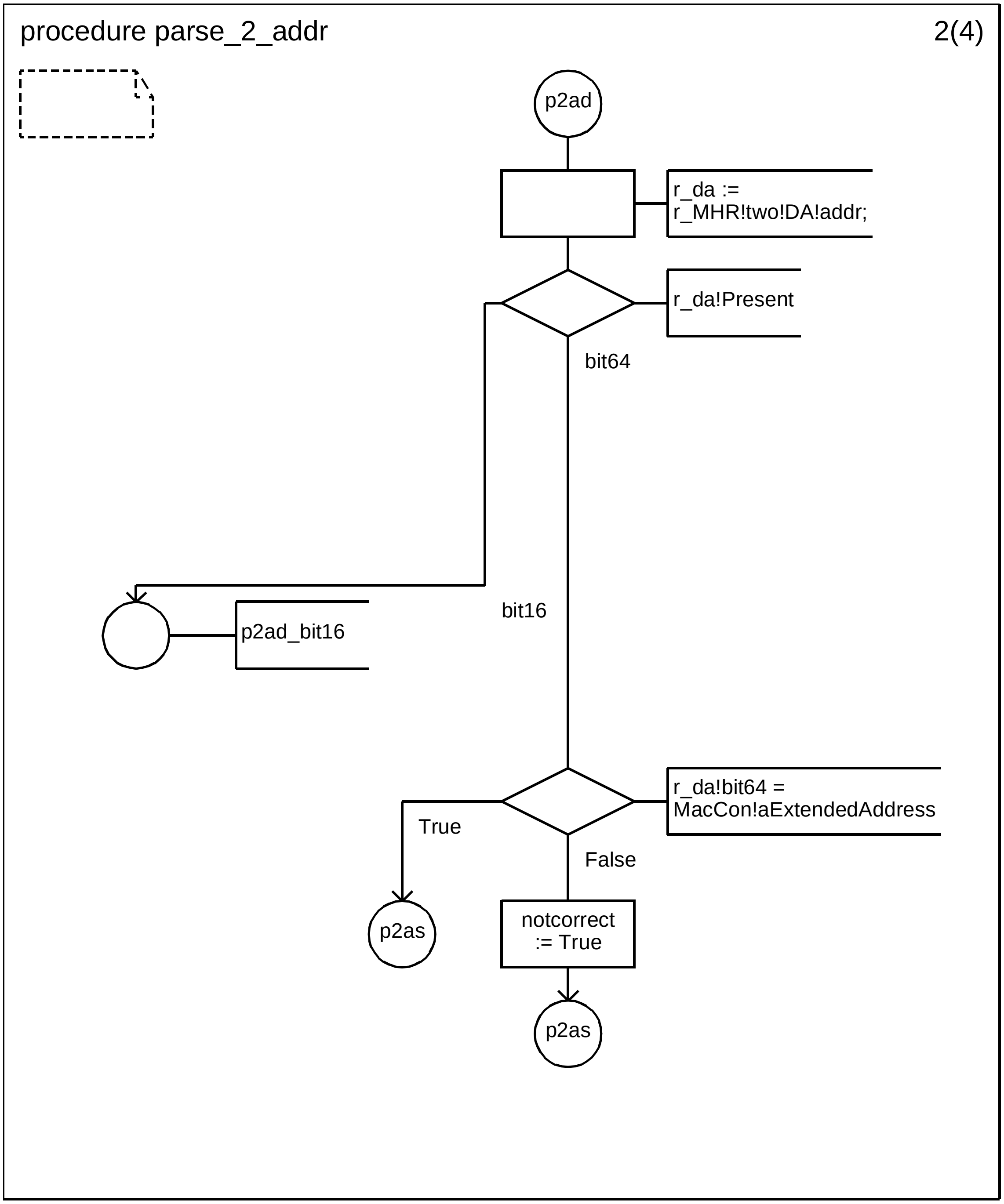
D.3.1.154.82.5 Procedure parse_2_addr (1)

.



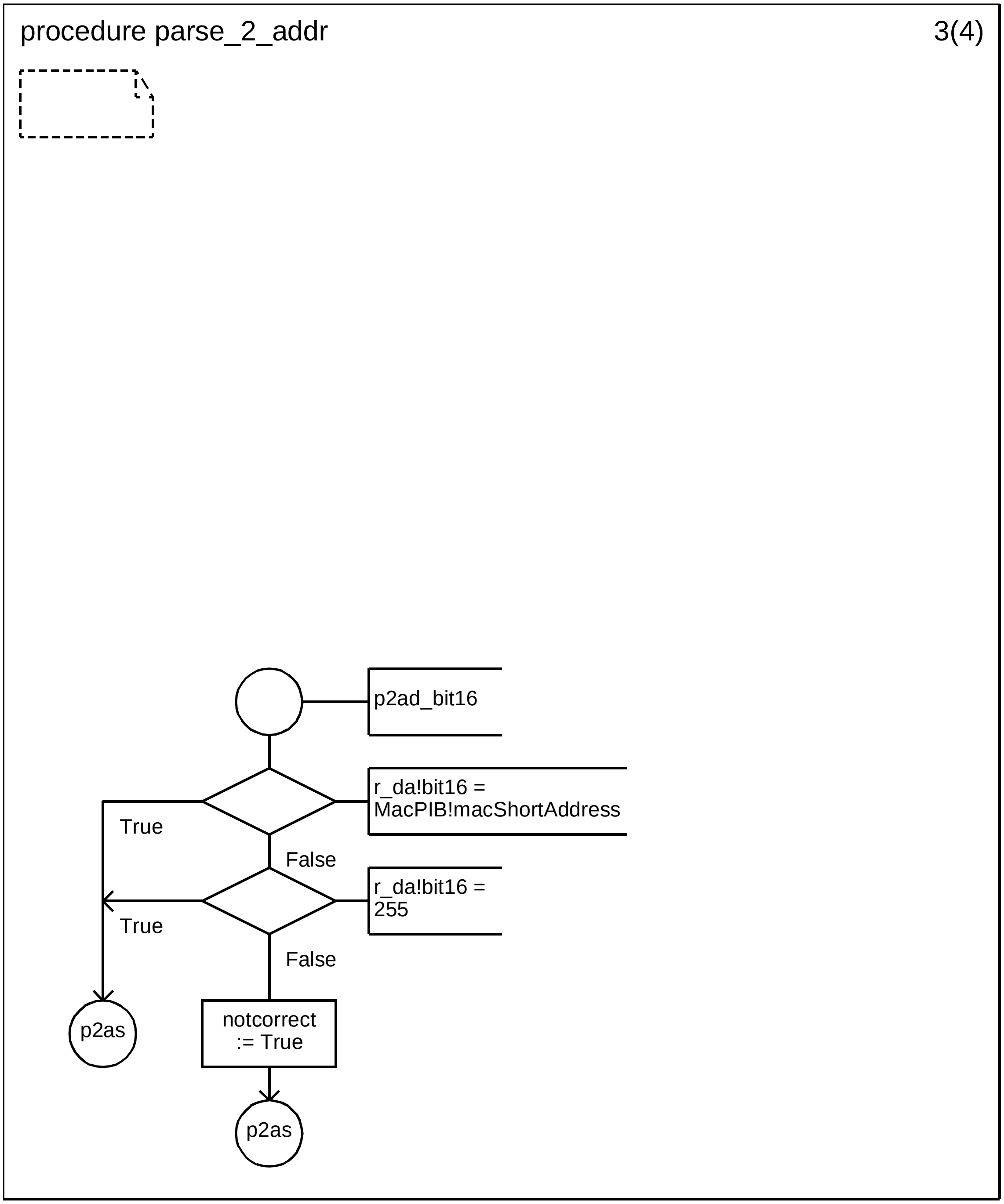
D.3.1.154.82.6 Procedure parse_2_addr (2)

.



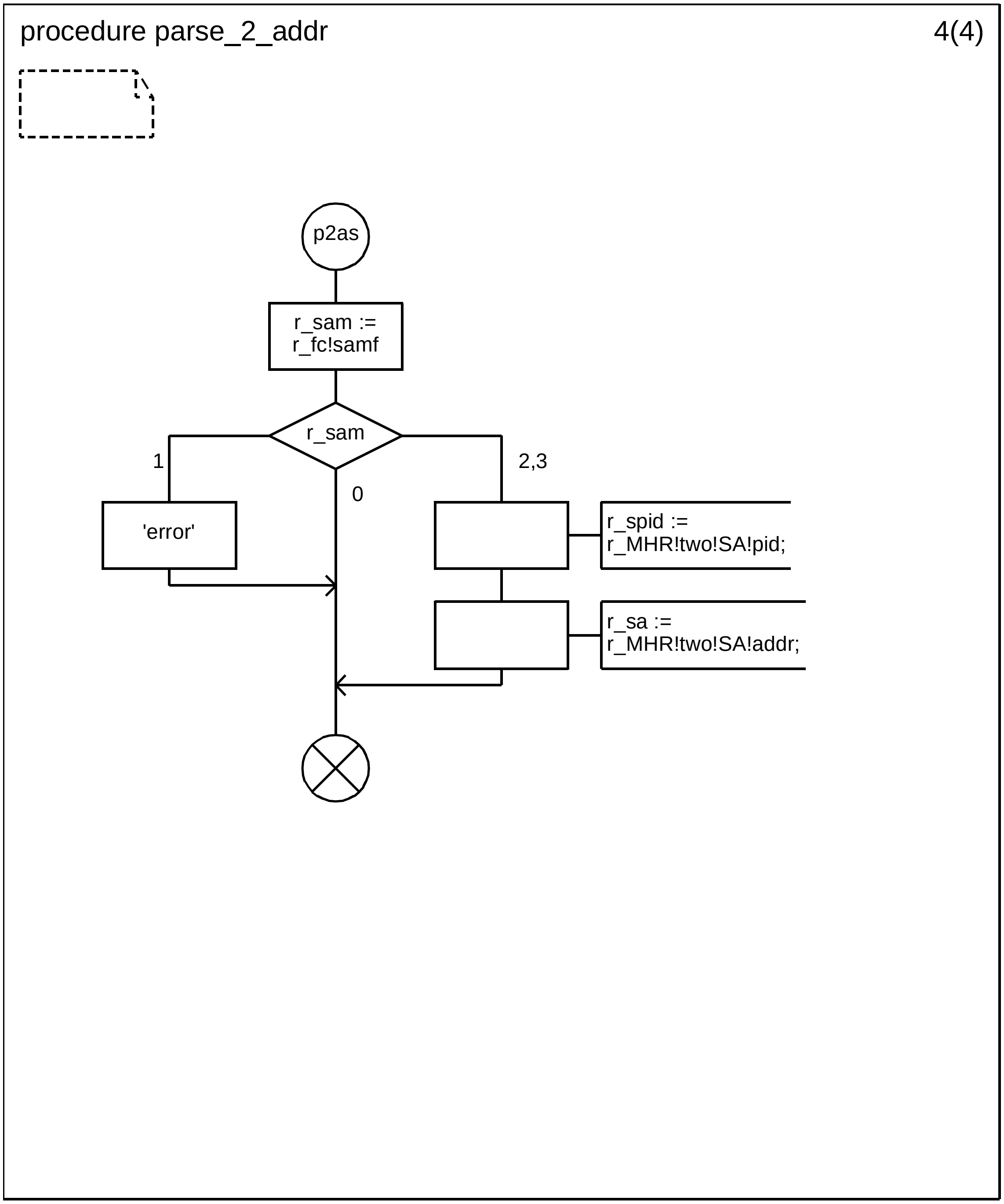
D.3.1.154.82.7 Procedure parse_2_addr (3)

.



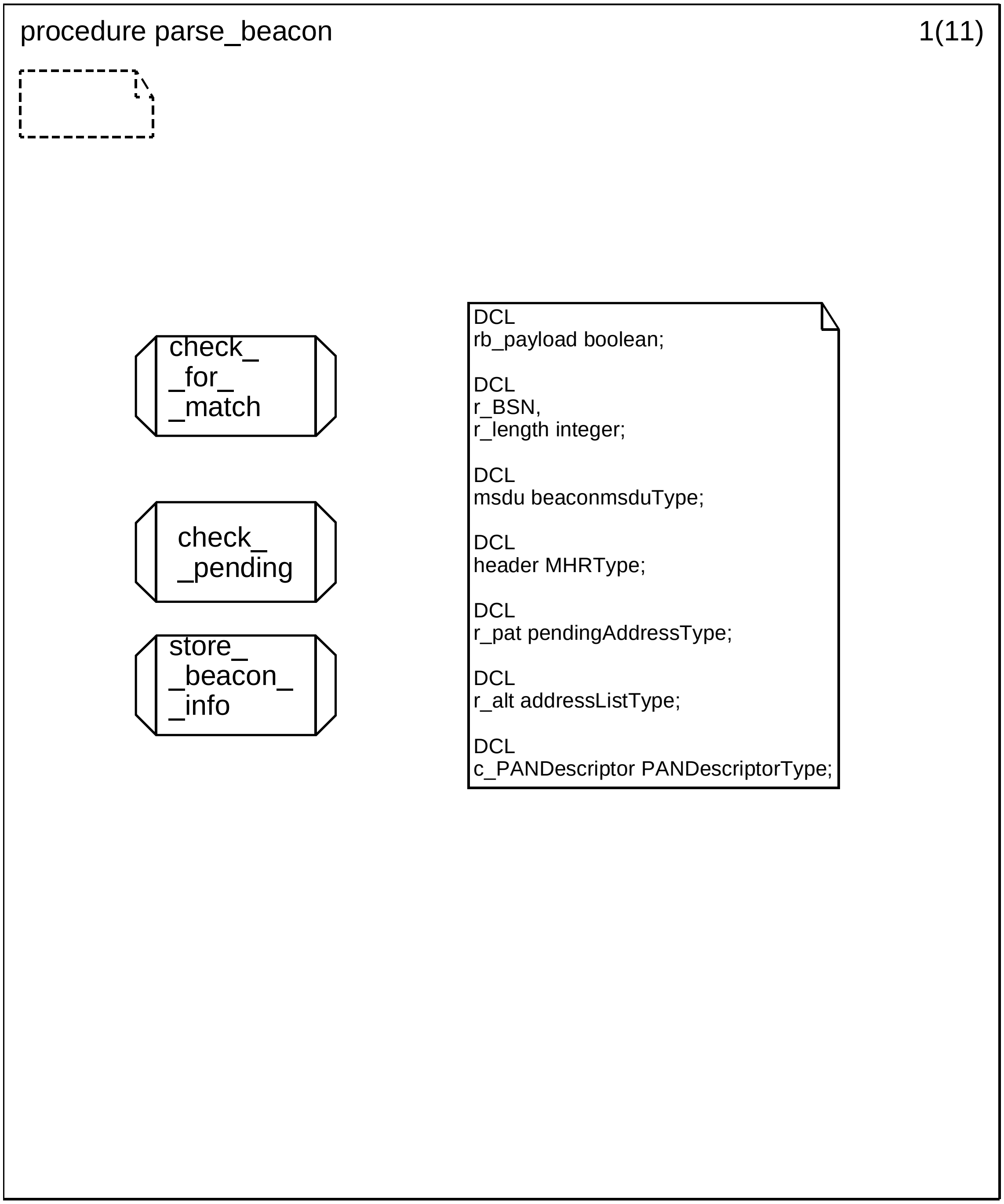
D.3.1.154.82.8 Procedure parse_2_addr (4)

.



D.3.1.154.82.9 Procedure parse_beacon (1)

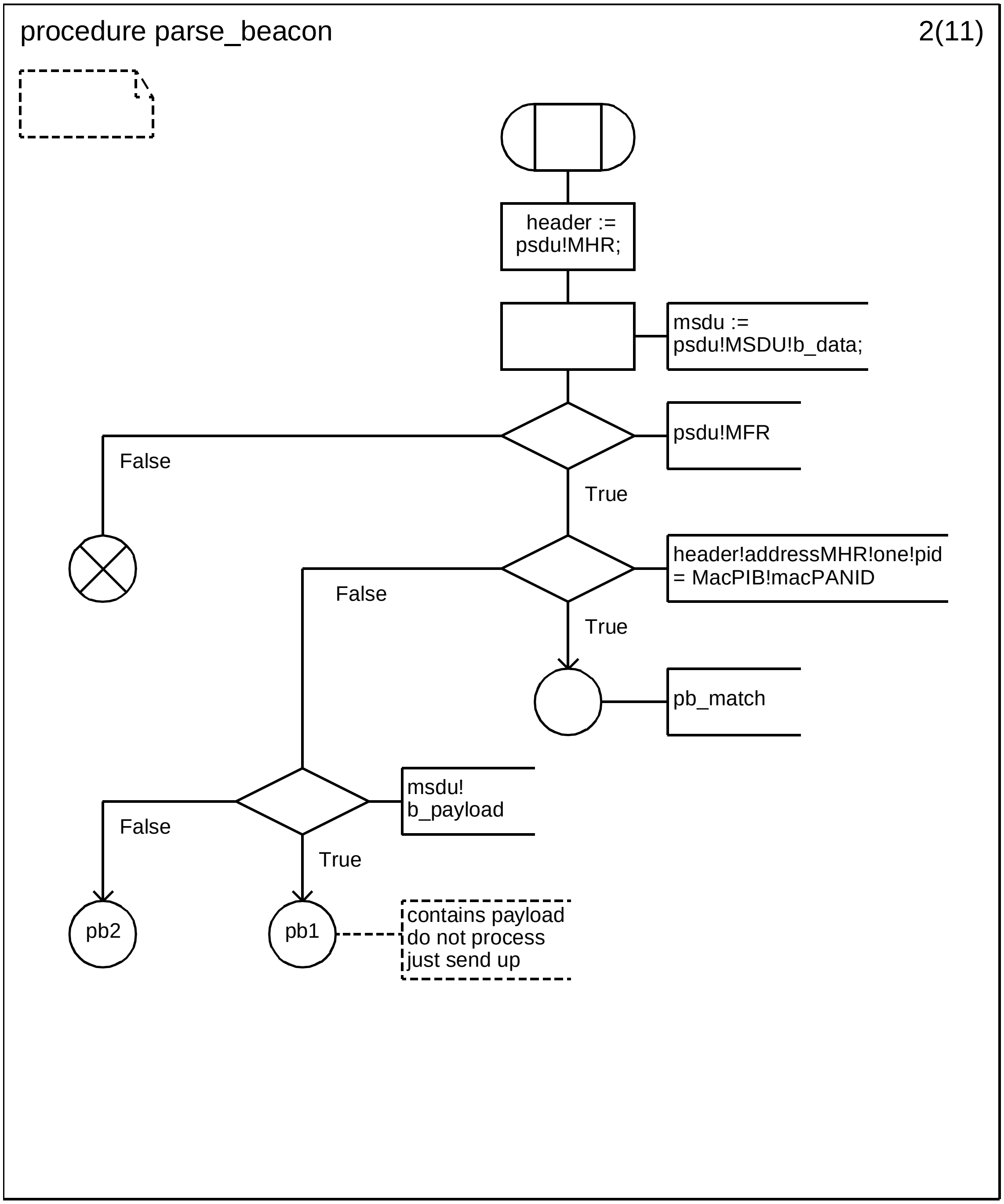
.



```
DCL
rb_payload boolean;

DCL
r_BSN,
r_length integer;

DCL
msdu beaconmsduType;

DCL
header MHRType;

DCL
r_pat pendingAddressType;

DCL
r_alt addressListType;

DCL
c_PANDescriptor PANDescriptorType;
```

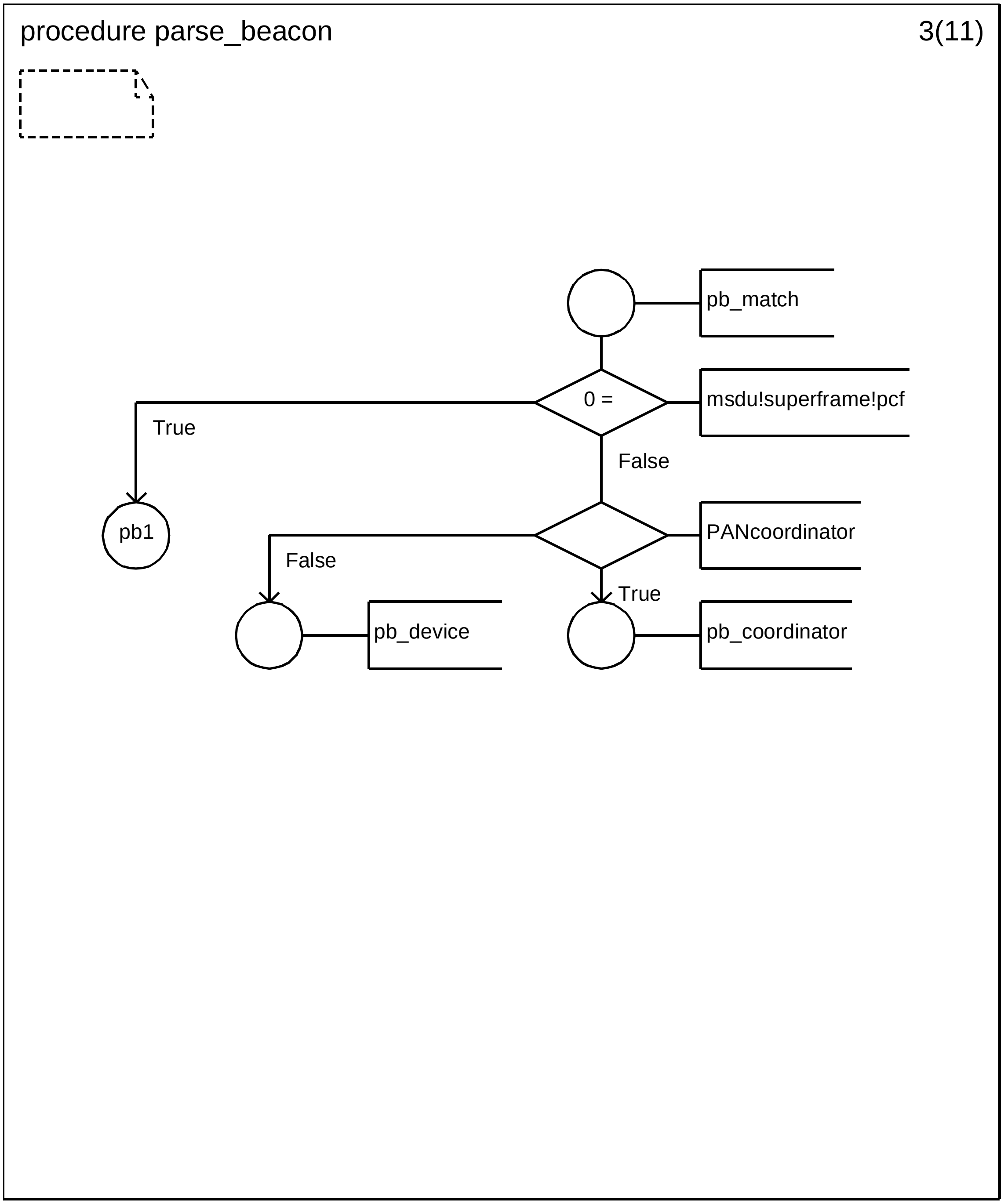
D.3.1.154.82.10 Procedure parse_beacon (2)

.



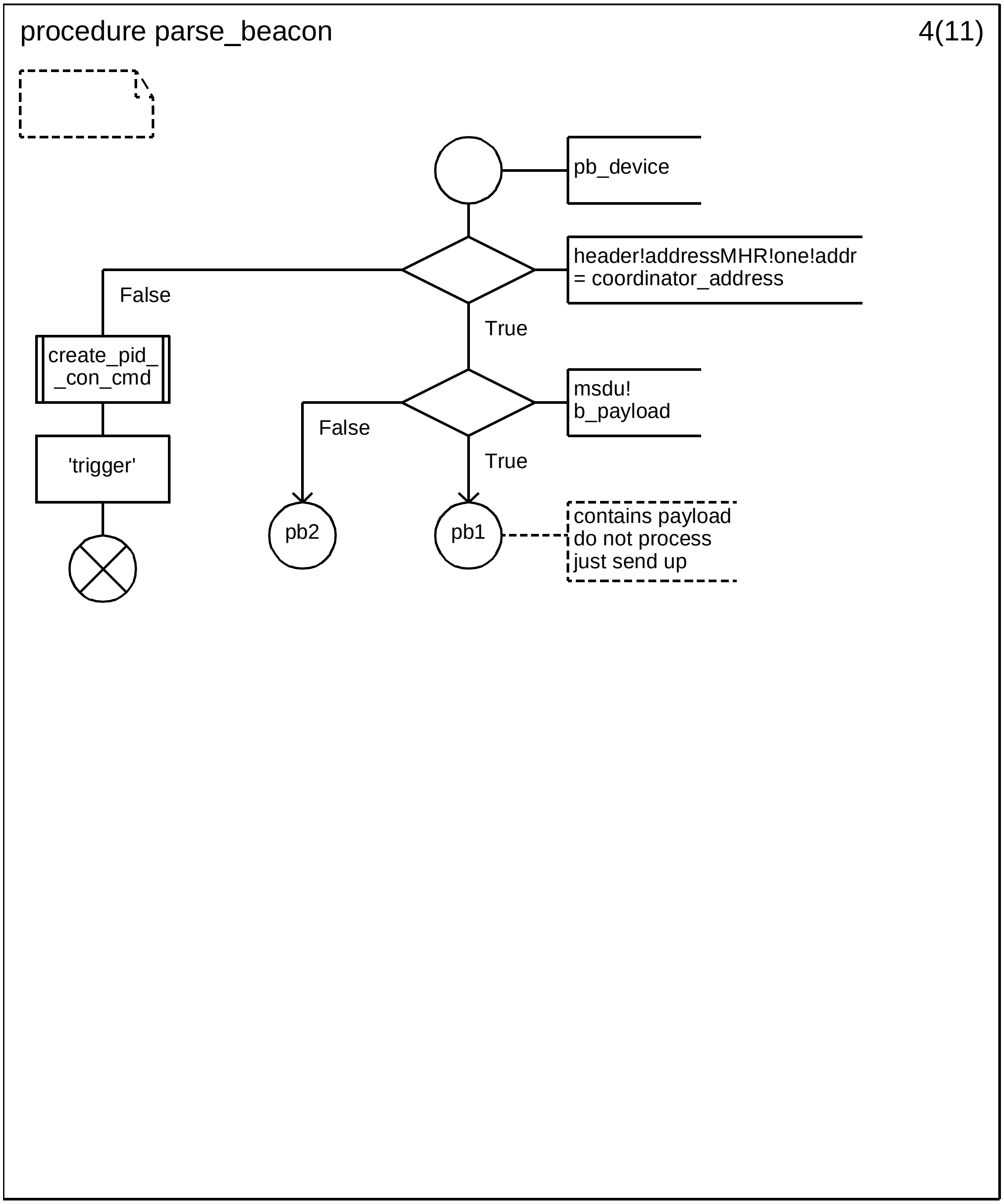
D.3.1.154.82.11 Procedure parse_beacon (3)

.



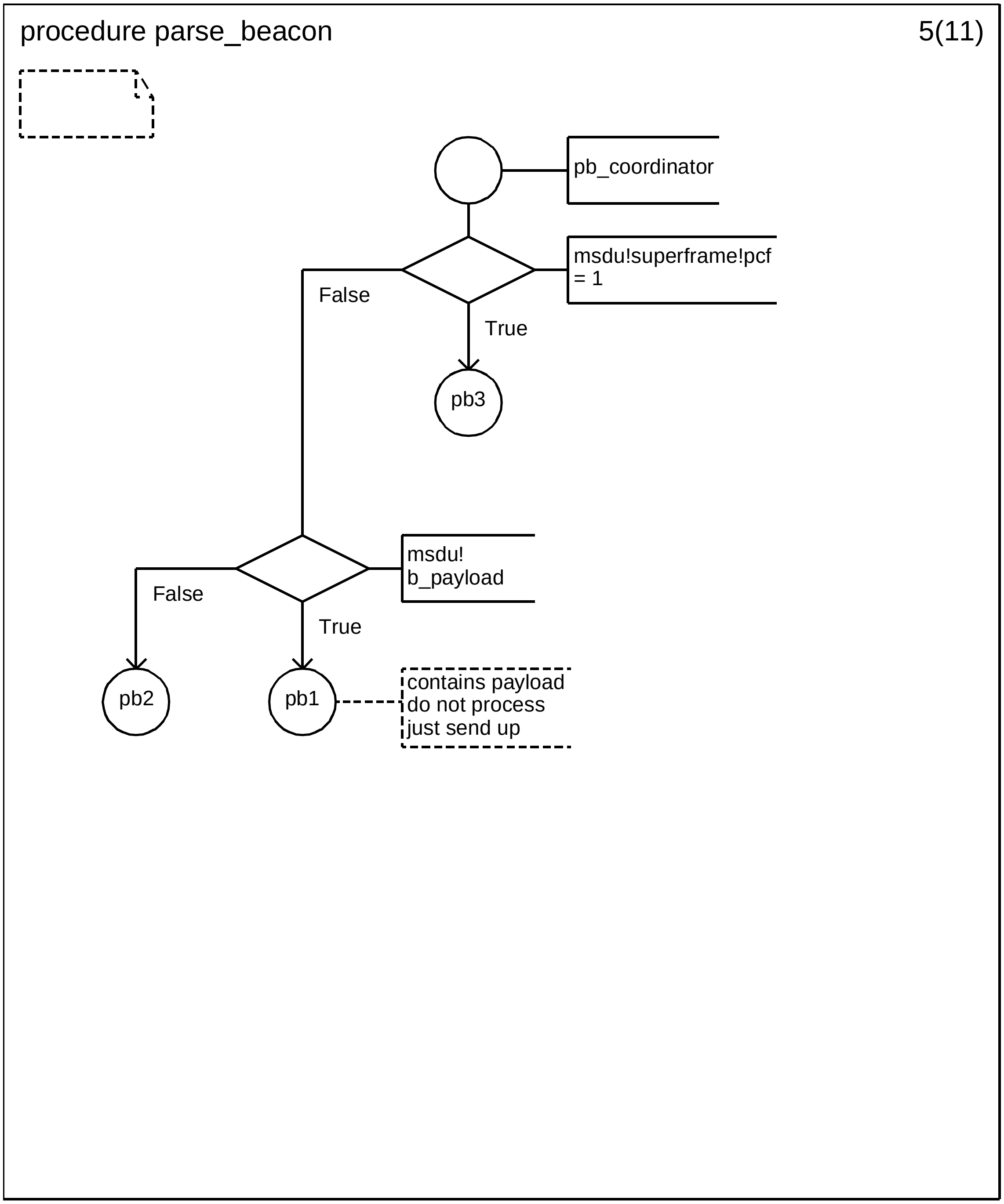
D.3.1.154.82.12 Procedure parse_beacon (4)

.



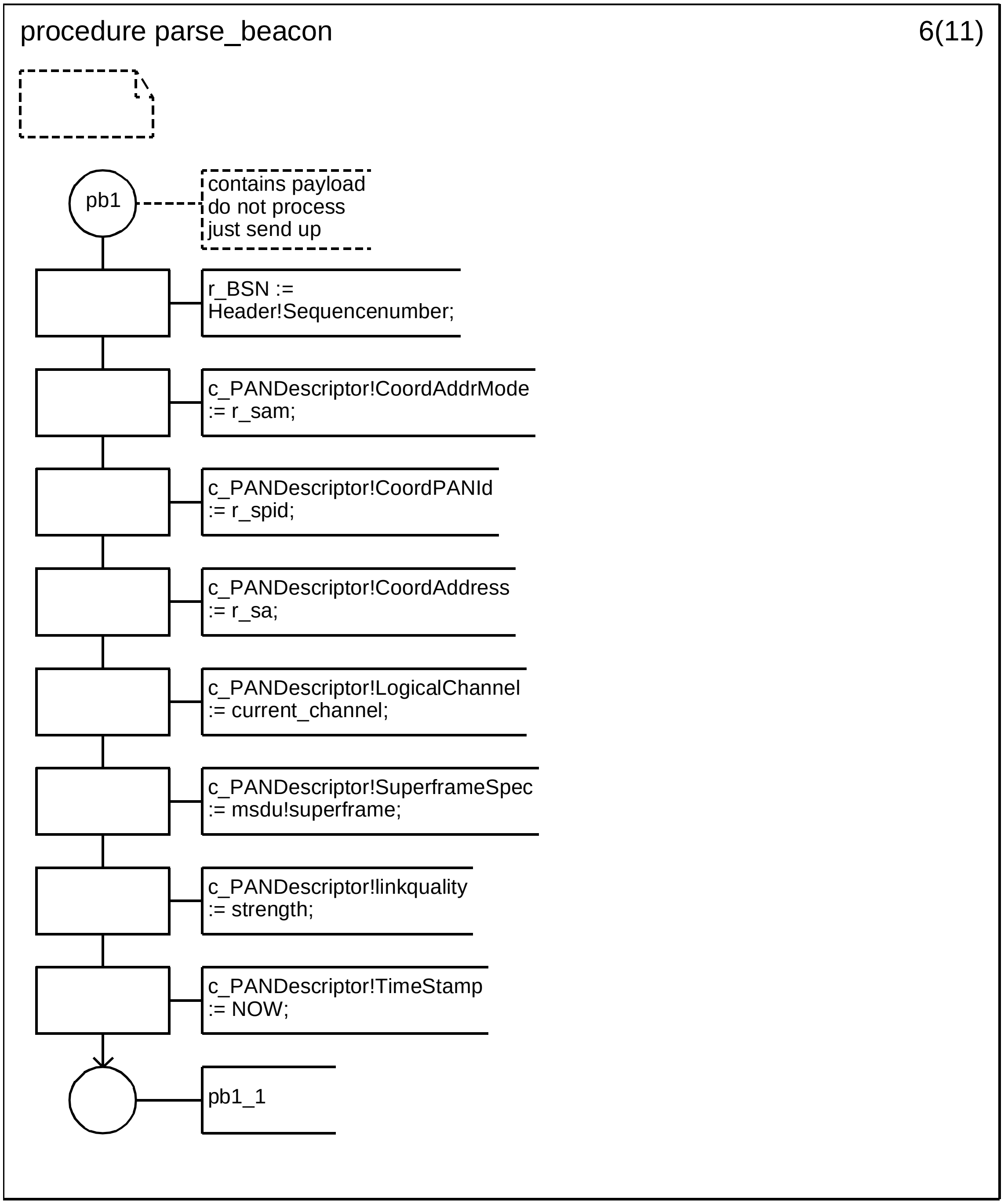
D.3.1.154.82.13 Procedure parse_beacon (5)

.



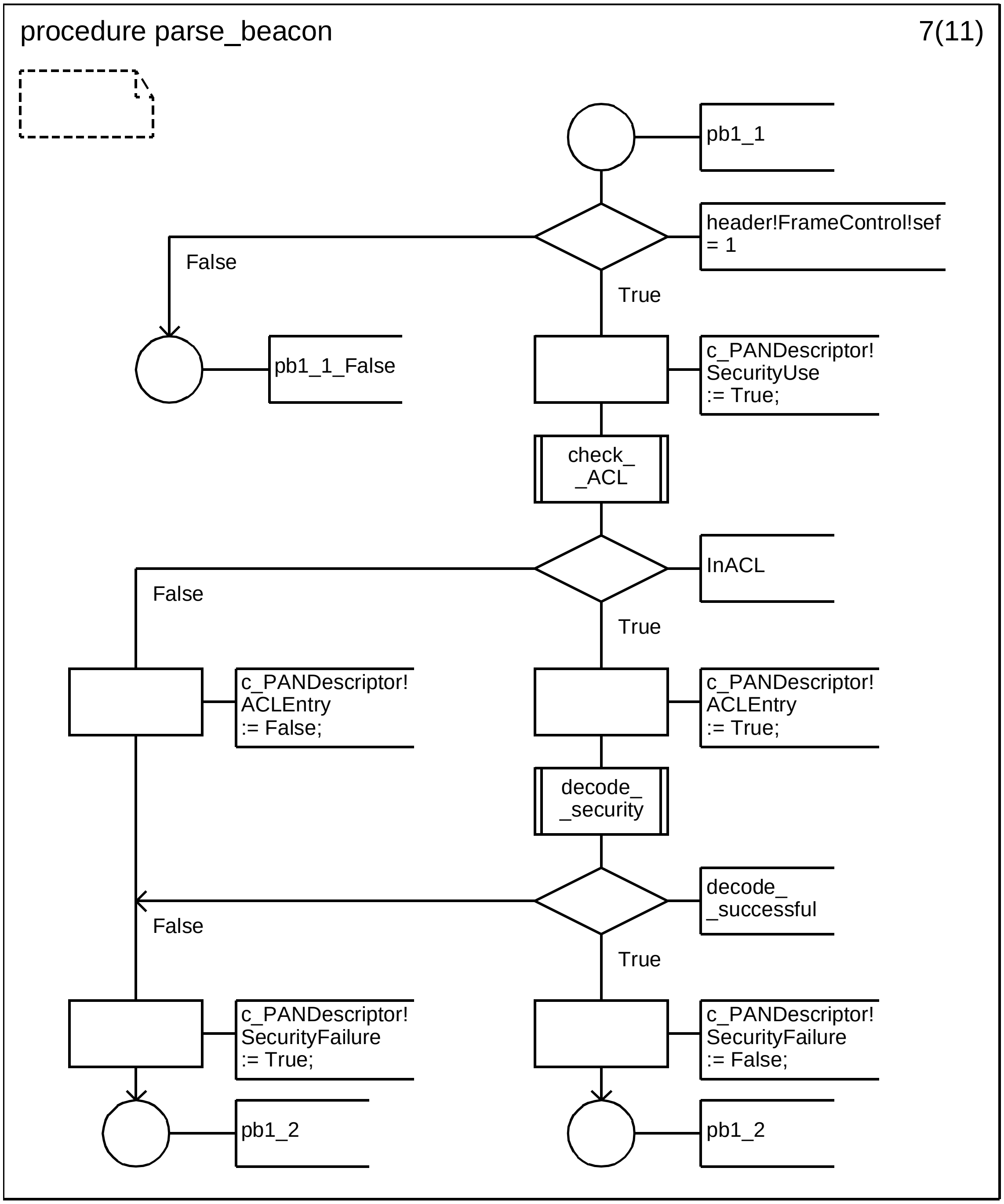
D.3.1.154.82.14 Procedure parse_beacon (6)

.


procedure parse_beacon
6(11)
pb1
contains payload
do not process
just send up
r_BSN :=
Header!Sequencenumber;
c_PANDescriptor!CoordAddrMode
:= r_sam;
c_PANDescriptor!CoordPANId
:= r_spid;
c_PANDescriptor!CoordAddress
:= r_sa;
c_PANDescriptor!LogicalChannel
:= current_channel;
c_PANDescriptor!SuperframeSpec
:= msdu!superframe;
c_PANDescriptor!linkquality
:= strength;
c_PANDescriptor!TimeStamp
:= NOW;
pb1_1

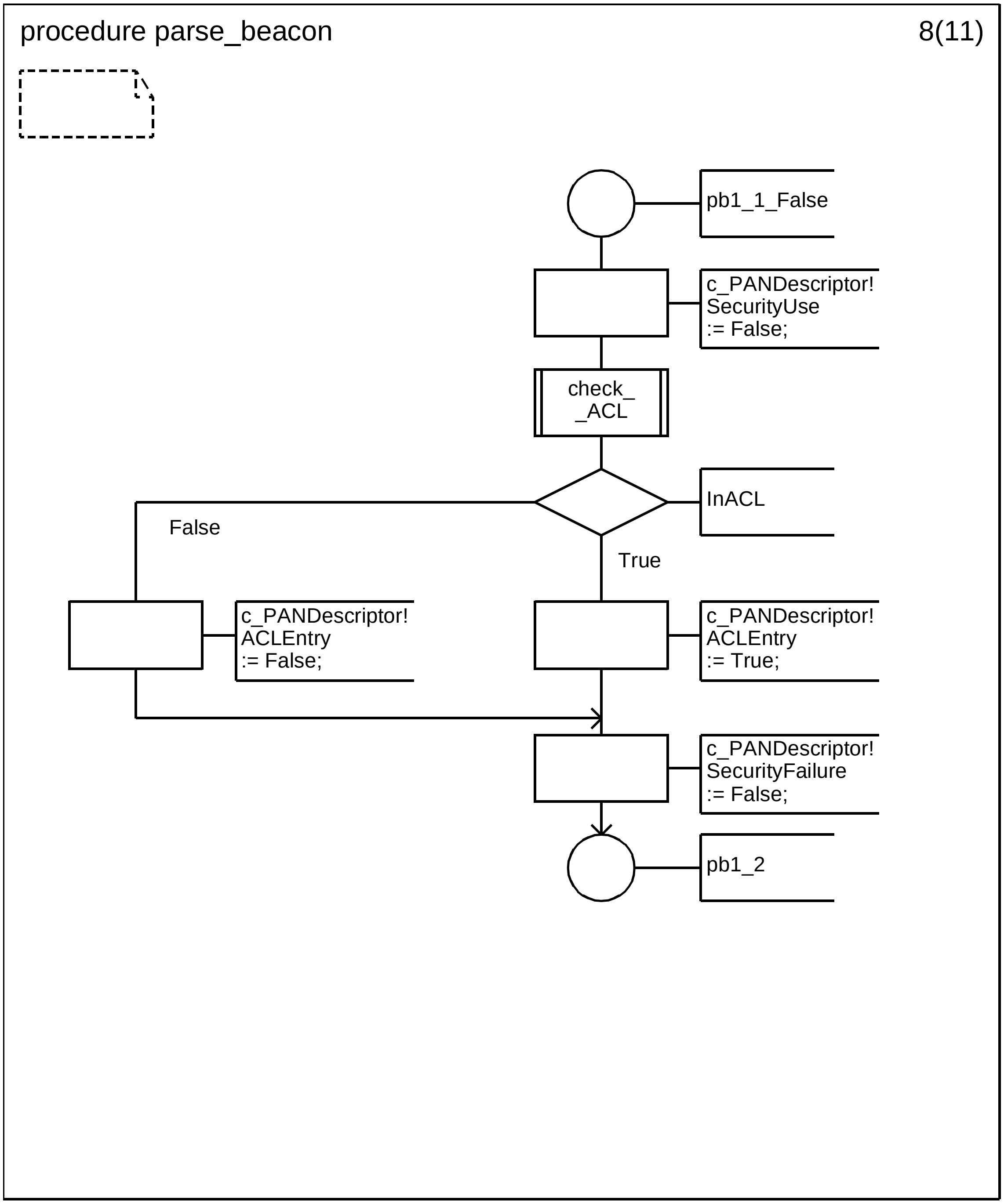
D.3.1.154.82.15 Procedure parse_beacon (7)

.

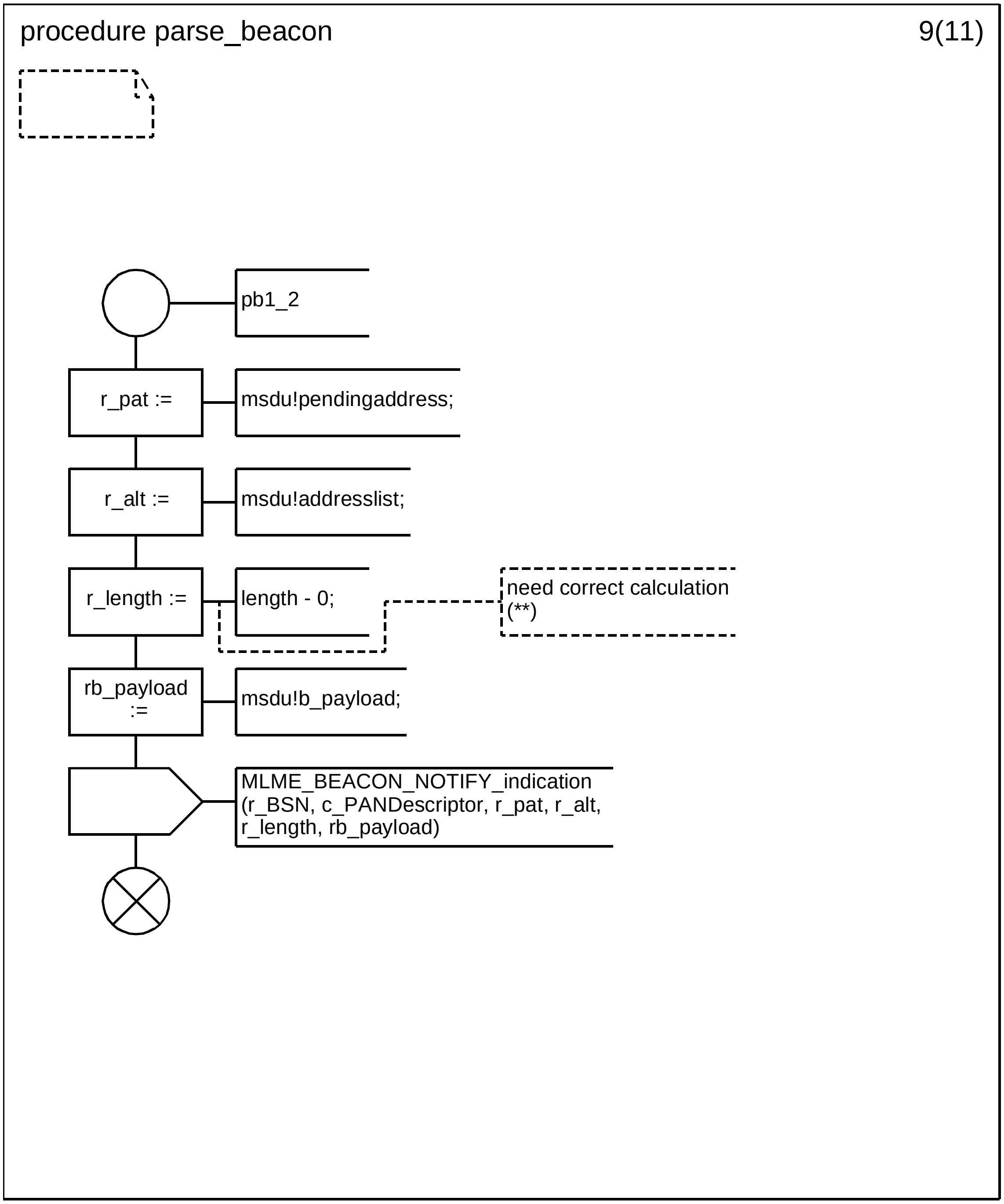


D.3.1.154.82.16 Procedure parse_beacon (8)

.

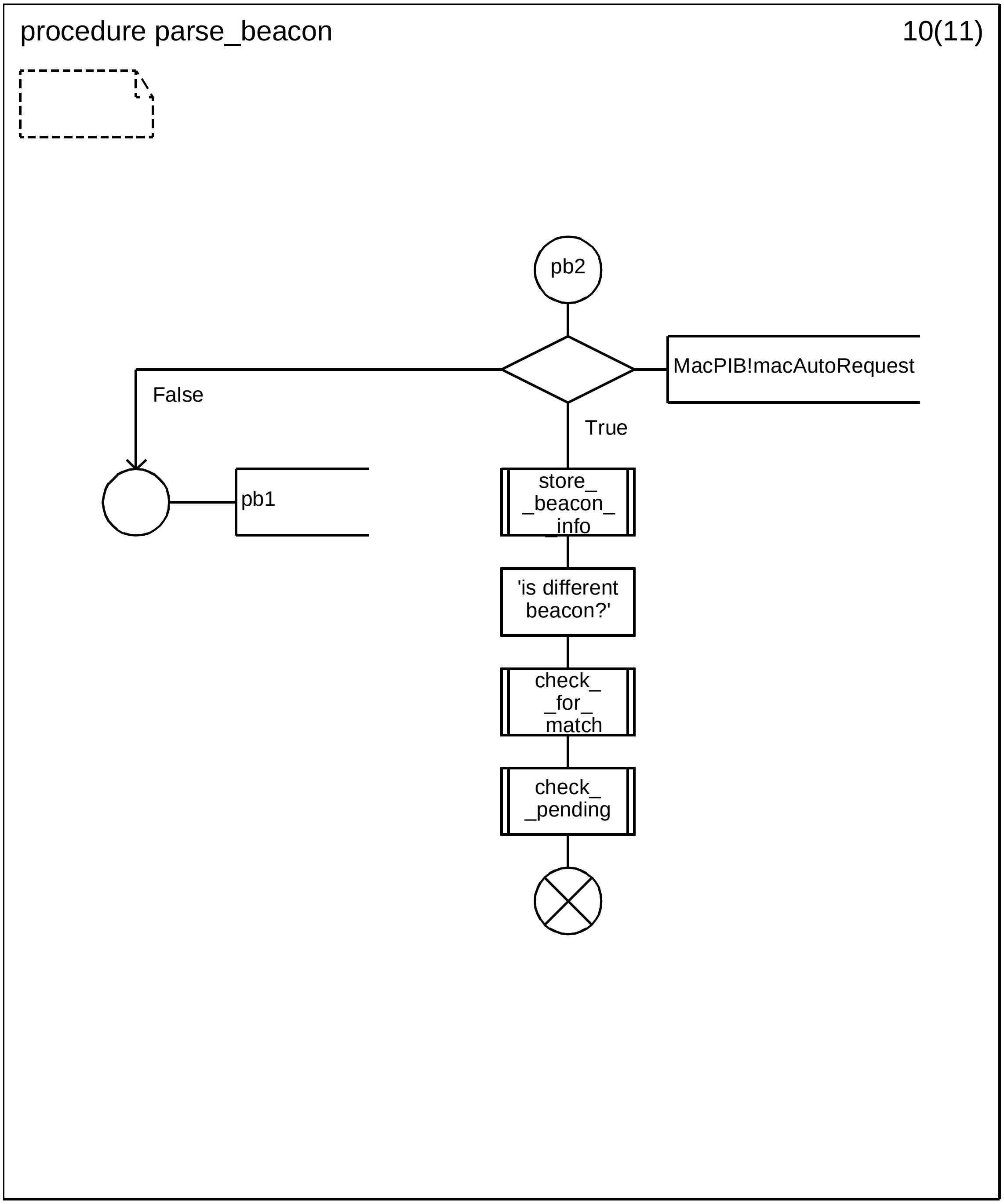


D.3.1.154.82.17 Procedure parse_beacon (9)



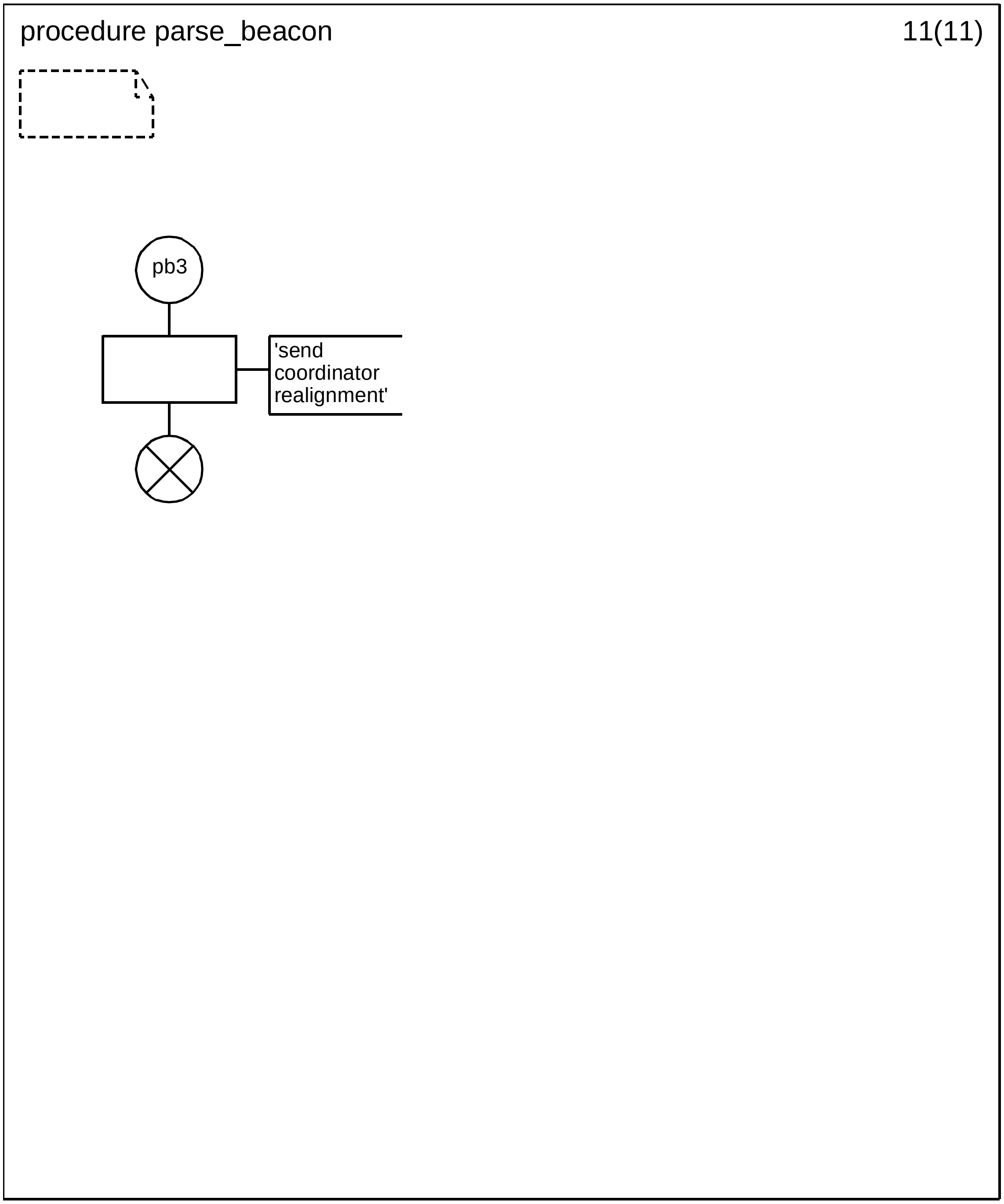
D.3.1.154.82.18 Procedure parse_beacon (10)

.



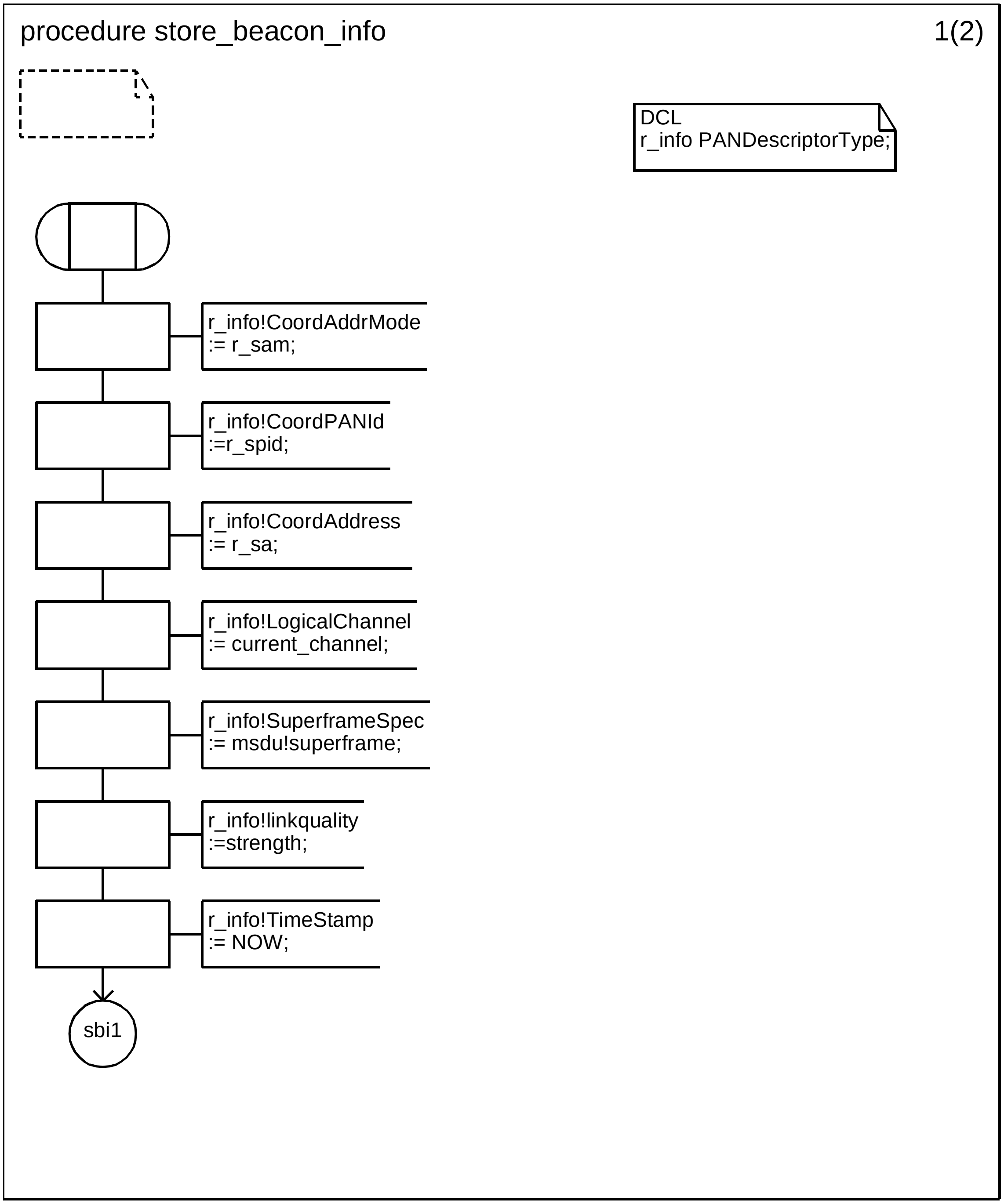
D.3.1.154.82.19 Procedure parse_beacon (11)

.



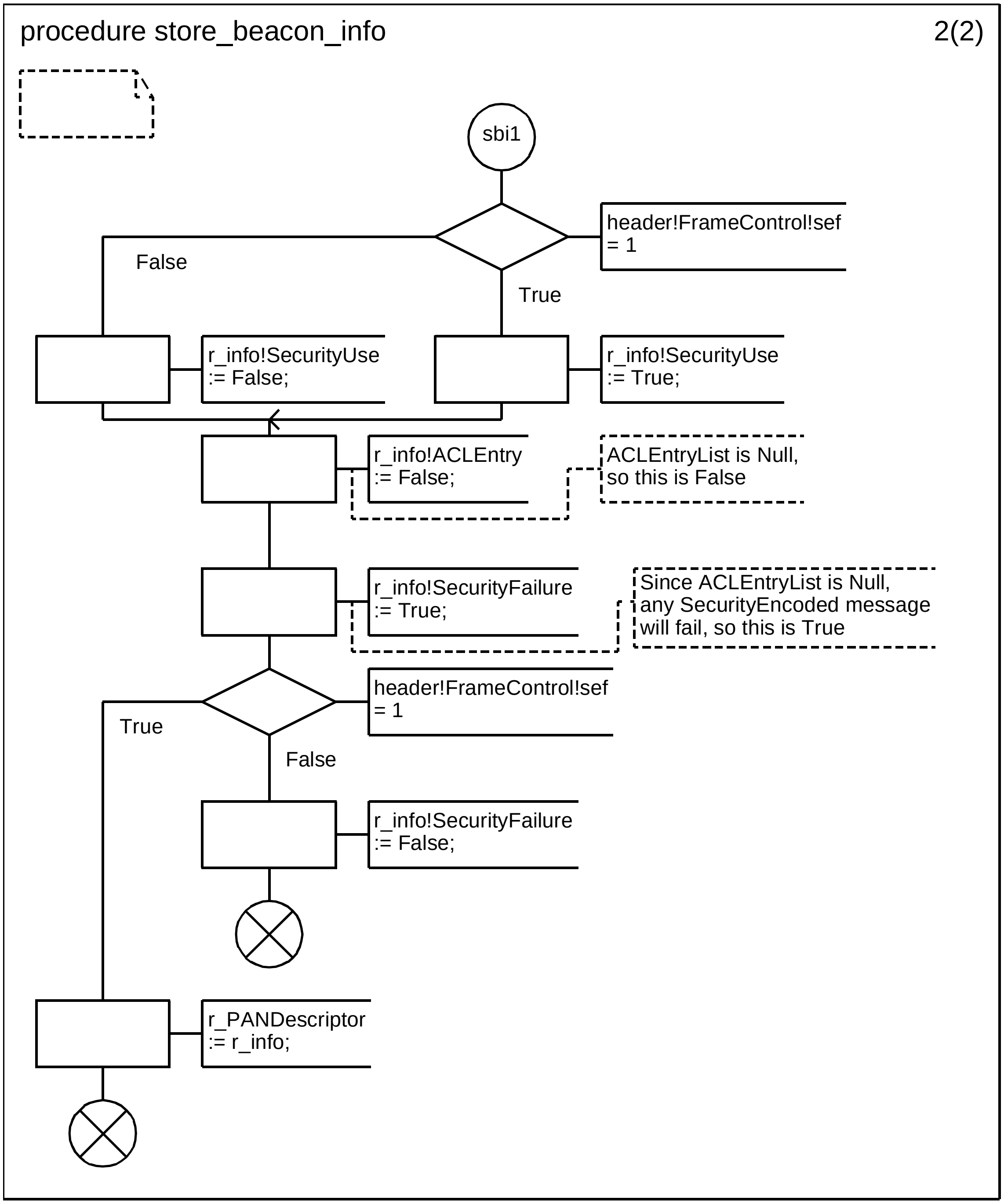
D.3.1.154.82.20 Procedure store_beacon_info (1)

.



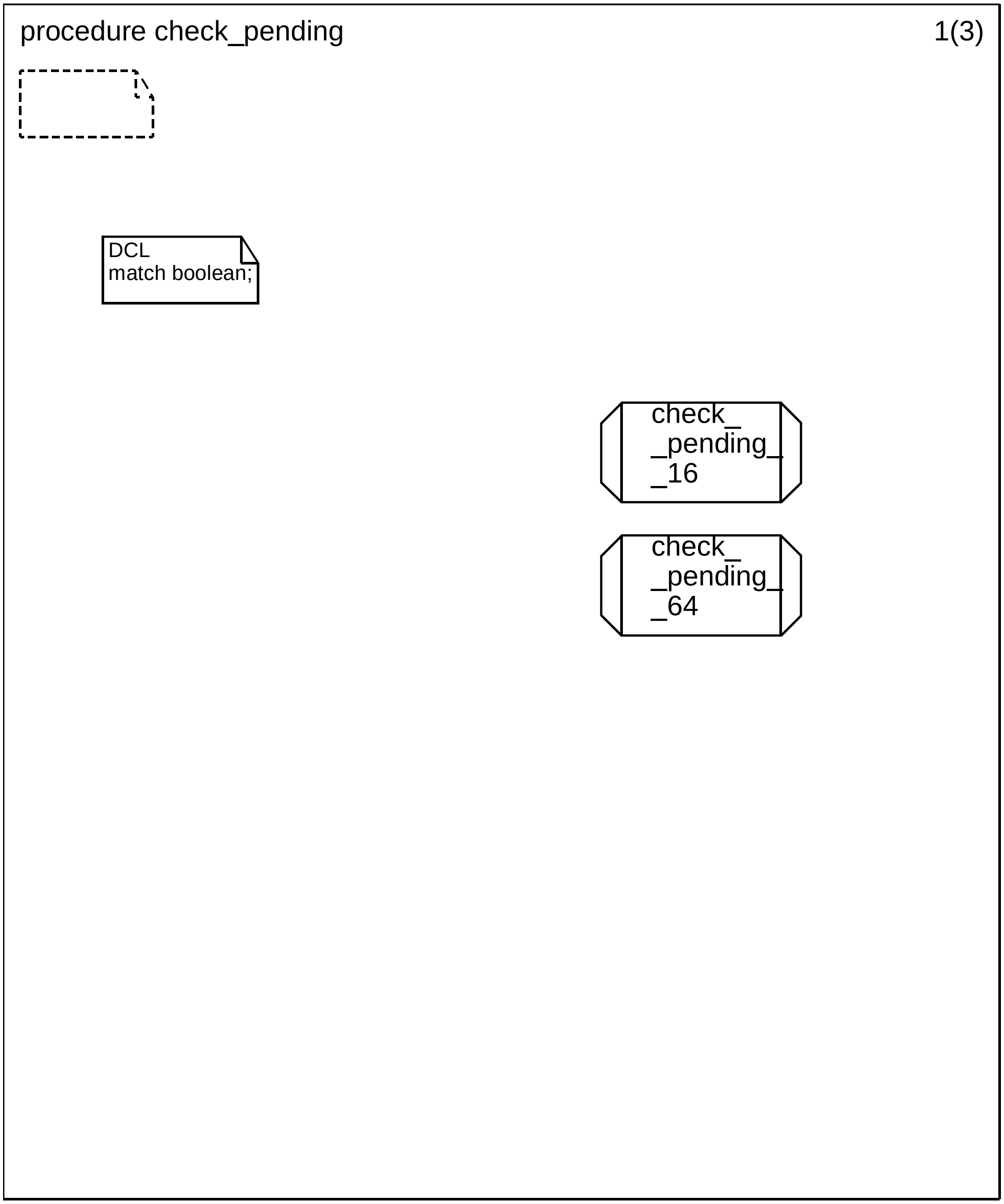
D.3.1.154.82.21 Procedure store_beacon_info (2)

.


procedure store_beacon_info
2(2)
sbi1
header!FrameControl!sef = 1
False
True
r_info!SecurityUse := False;
r_info!SecurityUse := True;
r_info!ACLEntry := False;
ACLEntryList is Null, so this is False
r_info!SecurityFailure := True;
Since ACLEntryList is Null, any SecurityEncoded message will fail, so this is True
header!FrameControl!sef = 1
True
False
r_info!SecurityFailure := False;
r_PANDescriptor := r_info;

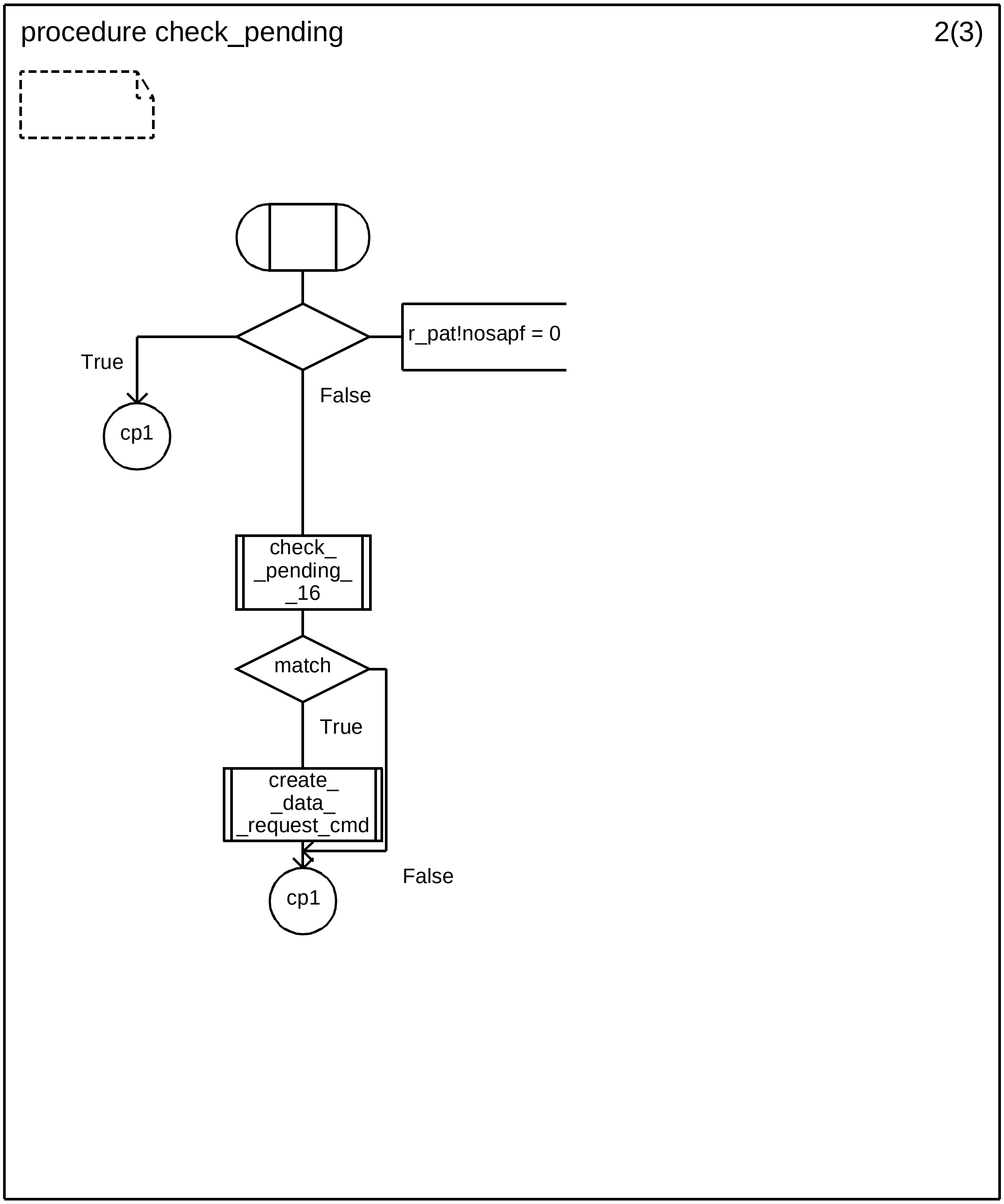
D.3.1.154.82.22 Procedure check_pending (1)

.



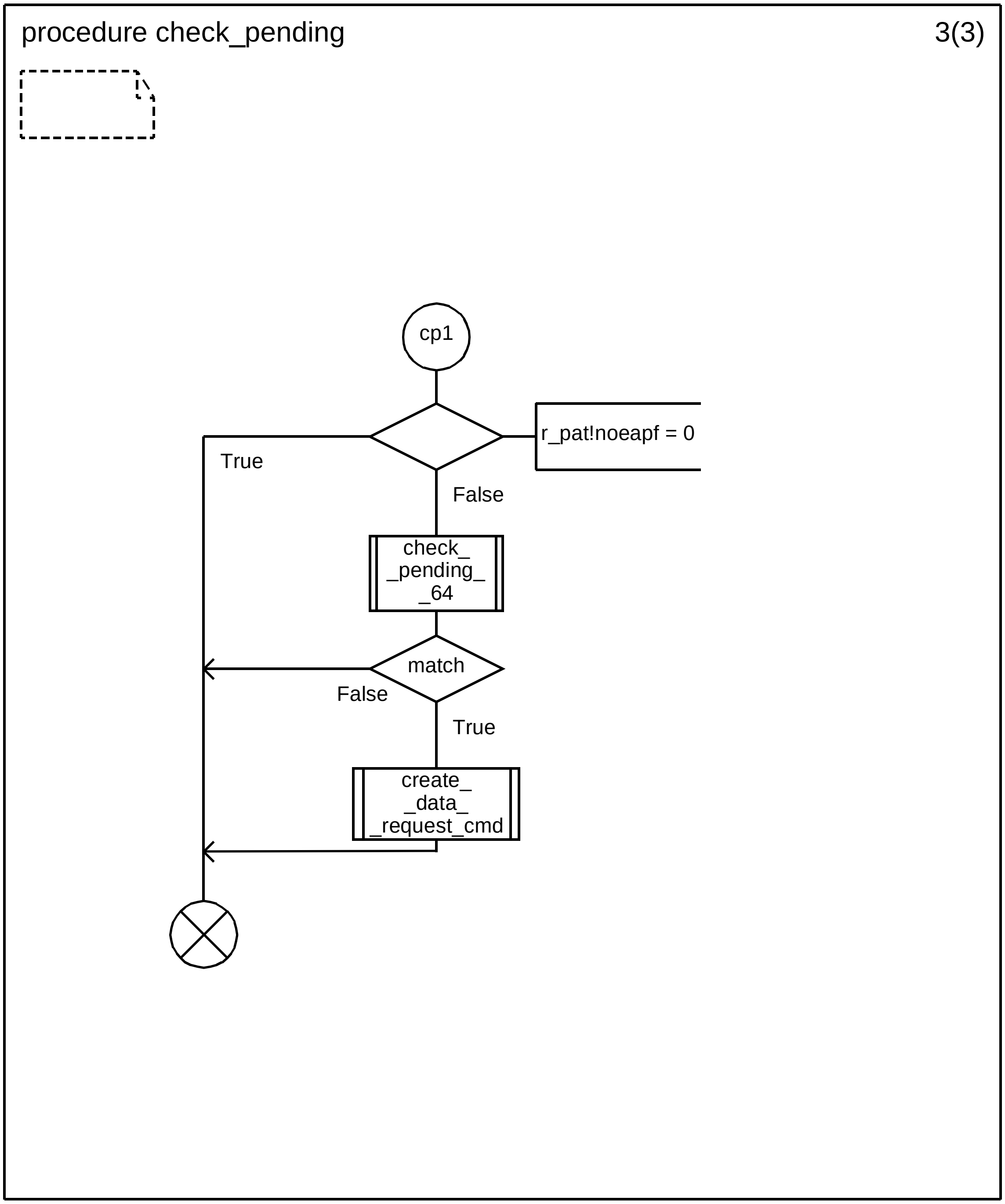
D.3.1.154.82.23 Procedure check_pending (2)

.



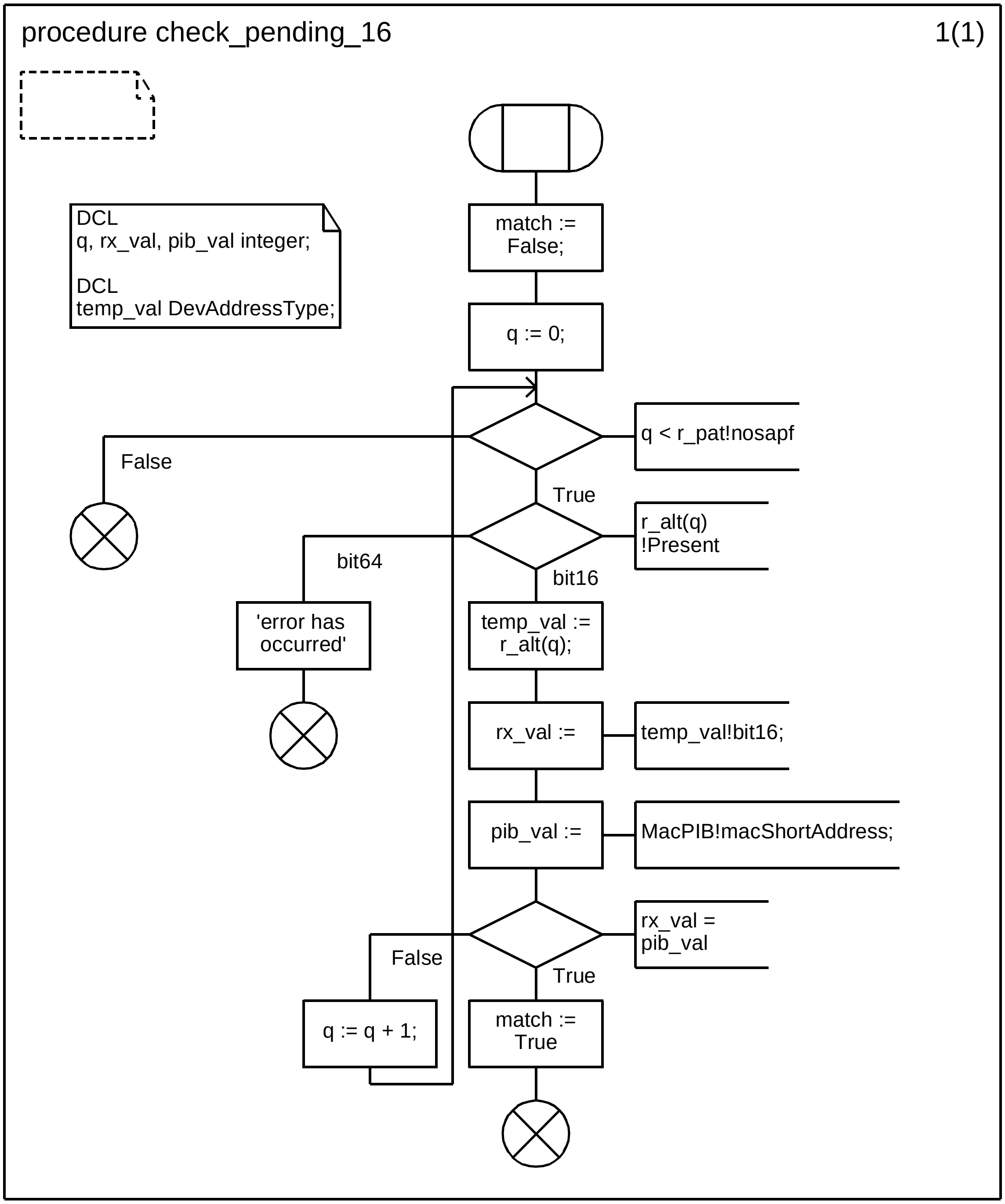
D.3.1.154.82.24 Procedure check_pending (3)

.



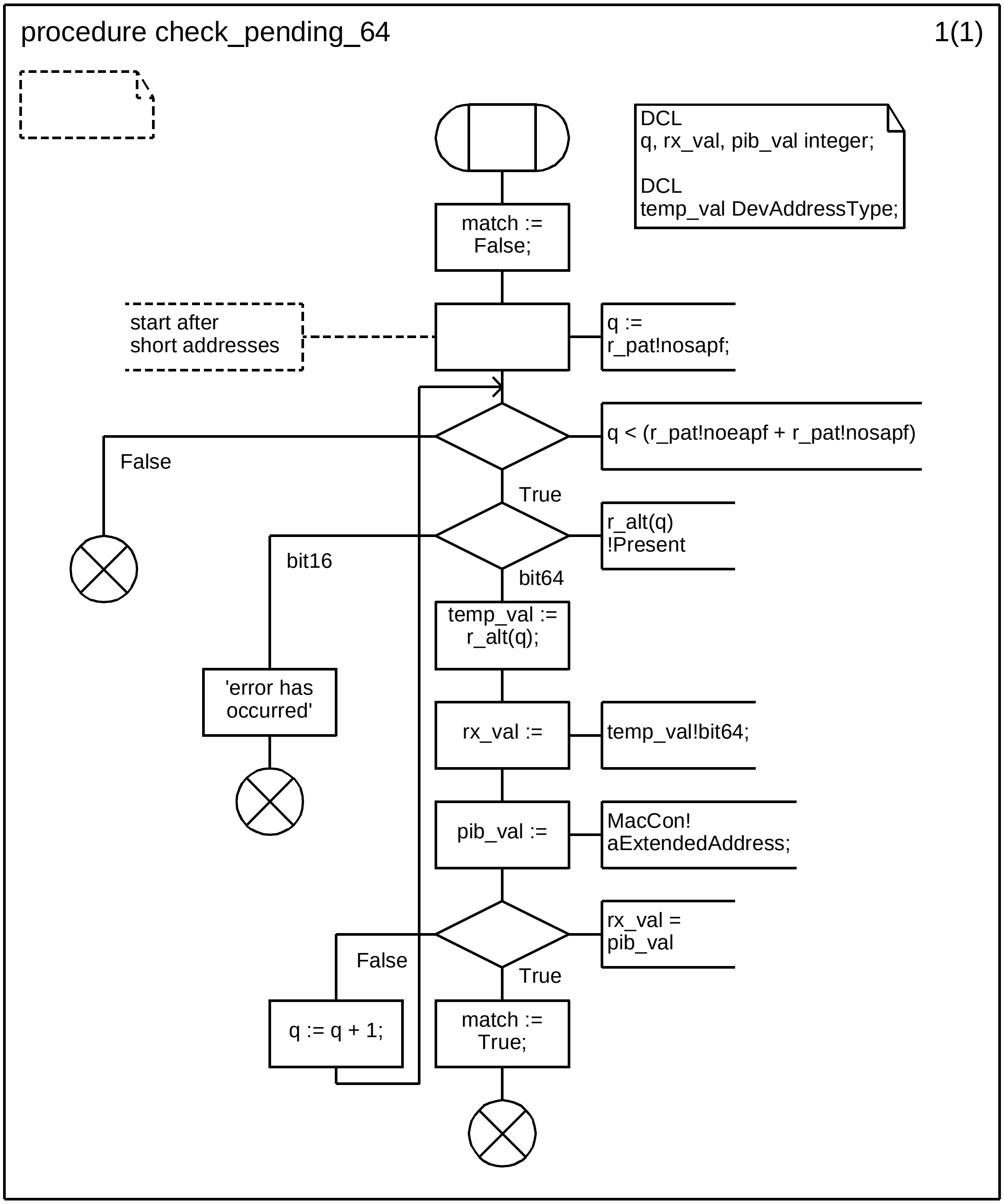
D.3.1.154.82.25 Procedure check_pending_16

.



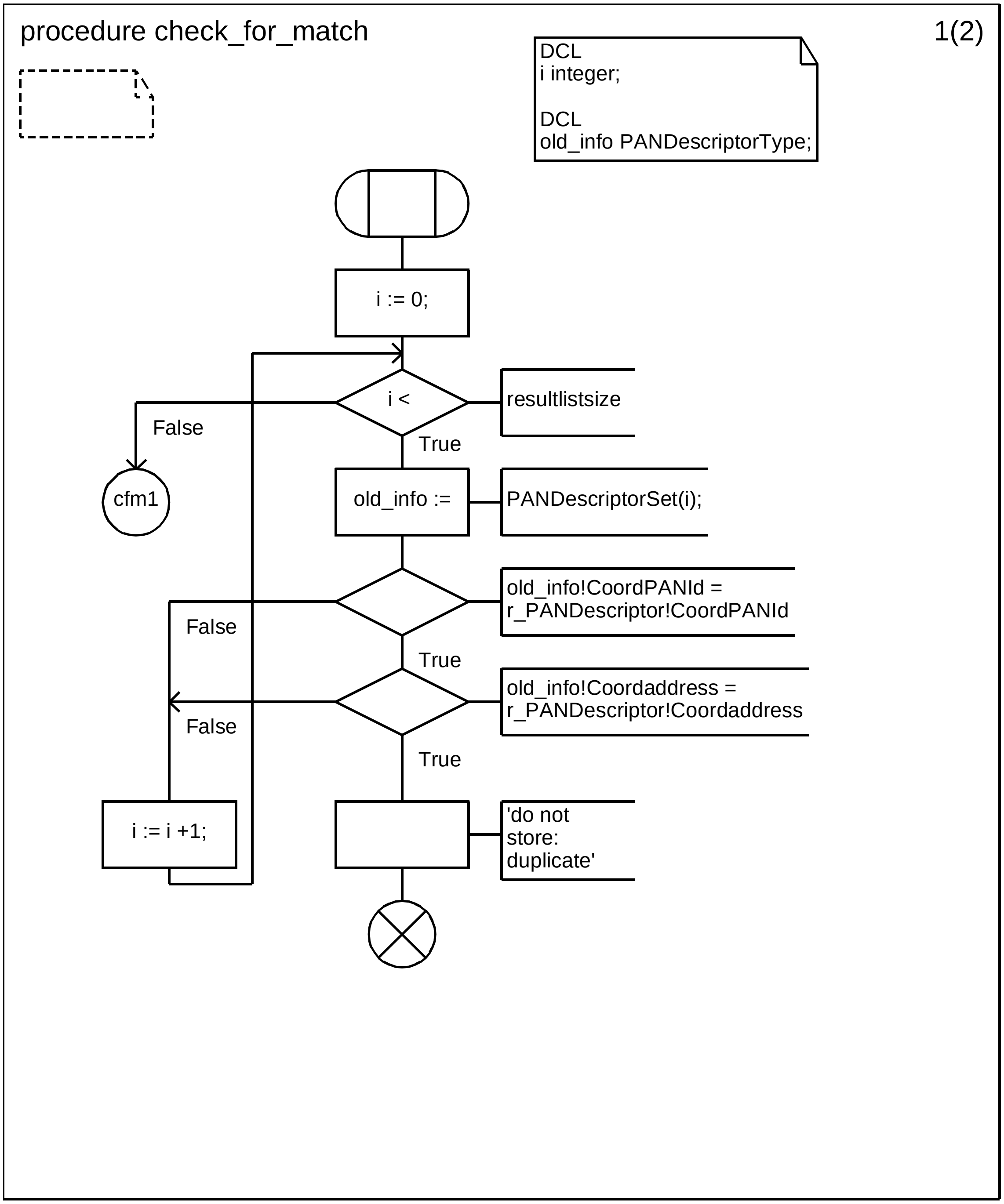
D.3.1.154.82.26 Procedure check_pending_64

.



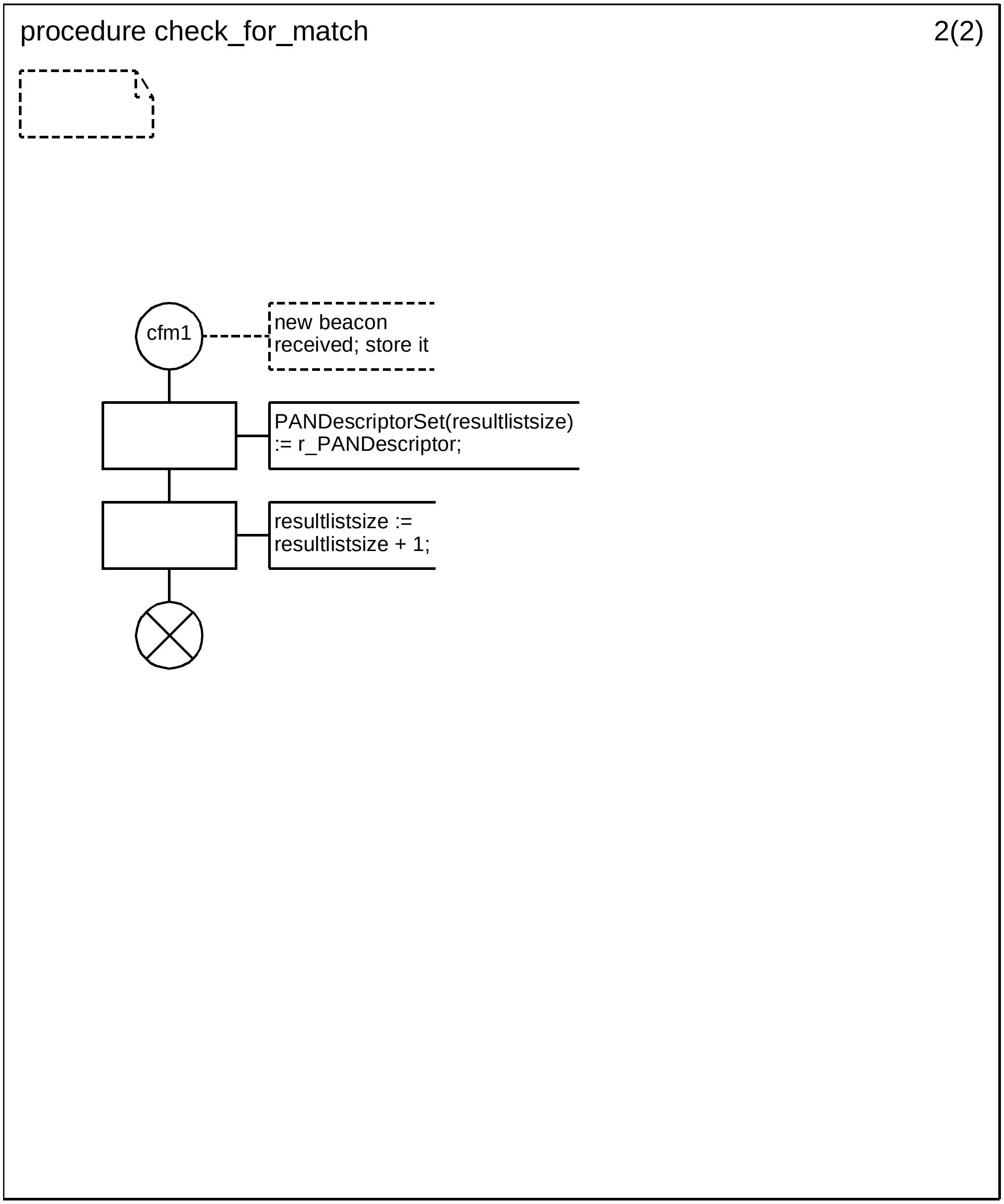
D.3.1.154.82.27 Procedure check_for_match (1)

.



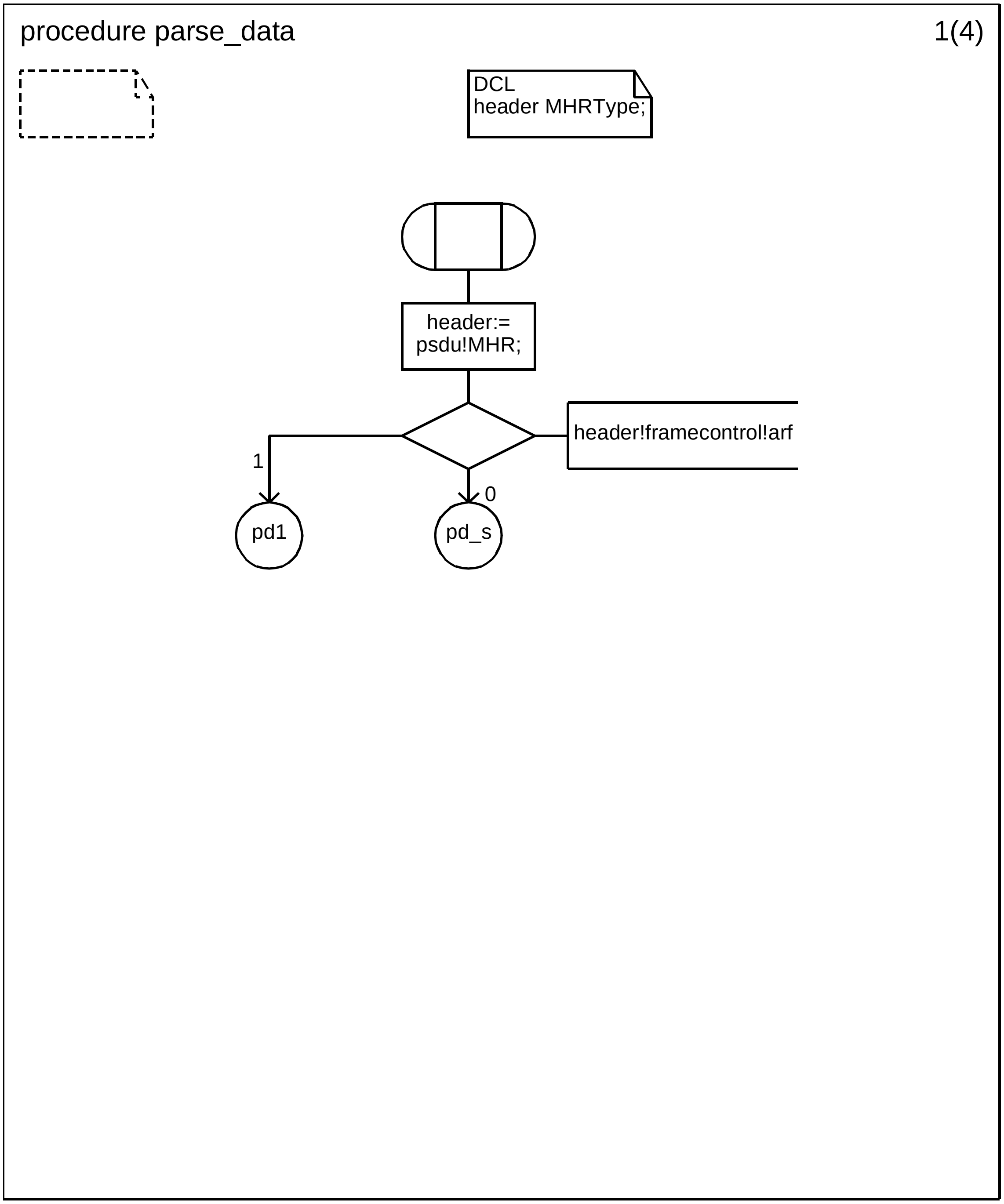
D.3.1.154.82.28 Procedure check_for_match (2)

.



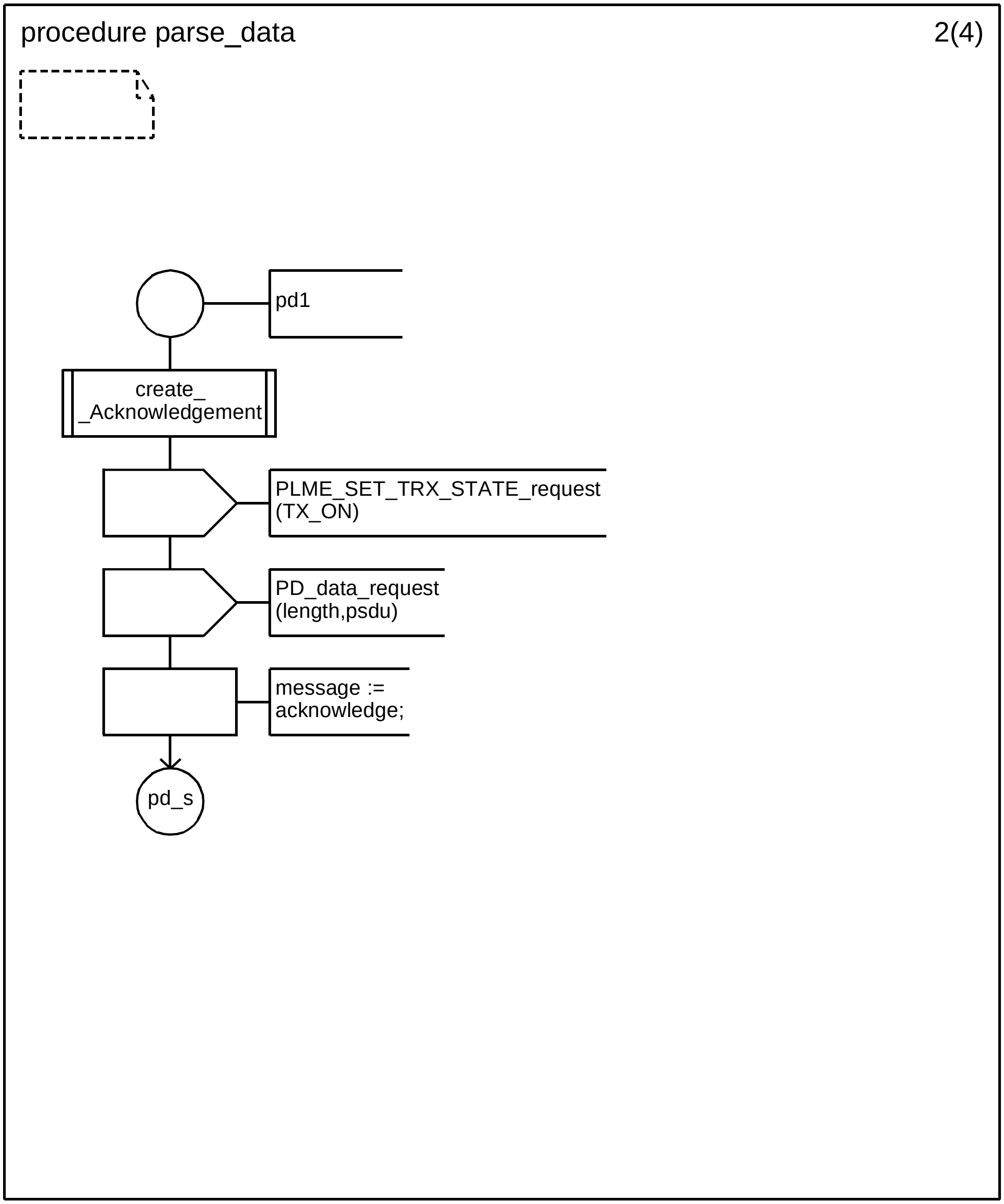
D.3.1.154.82.29 Procedure parse_data (1)

.



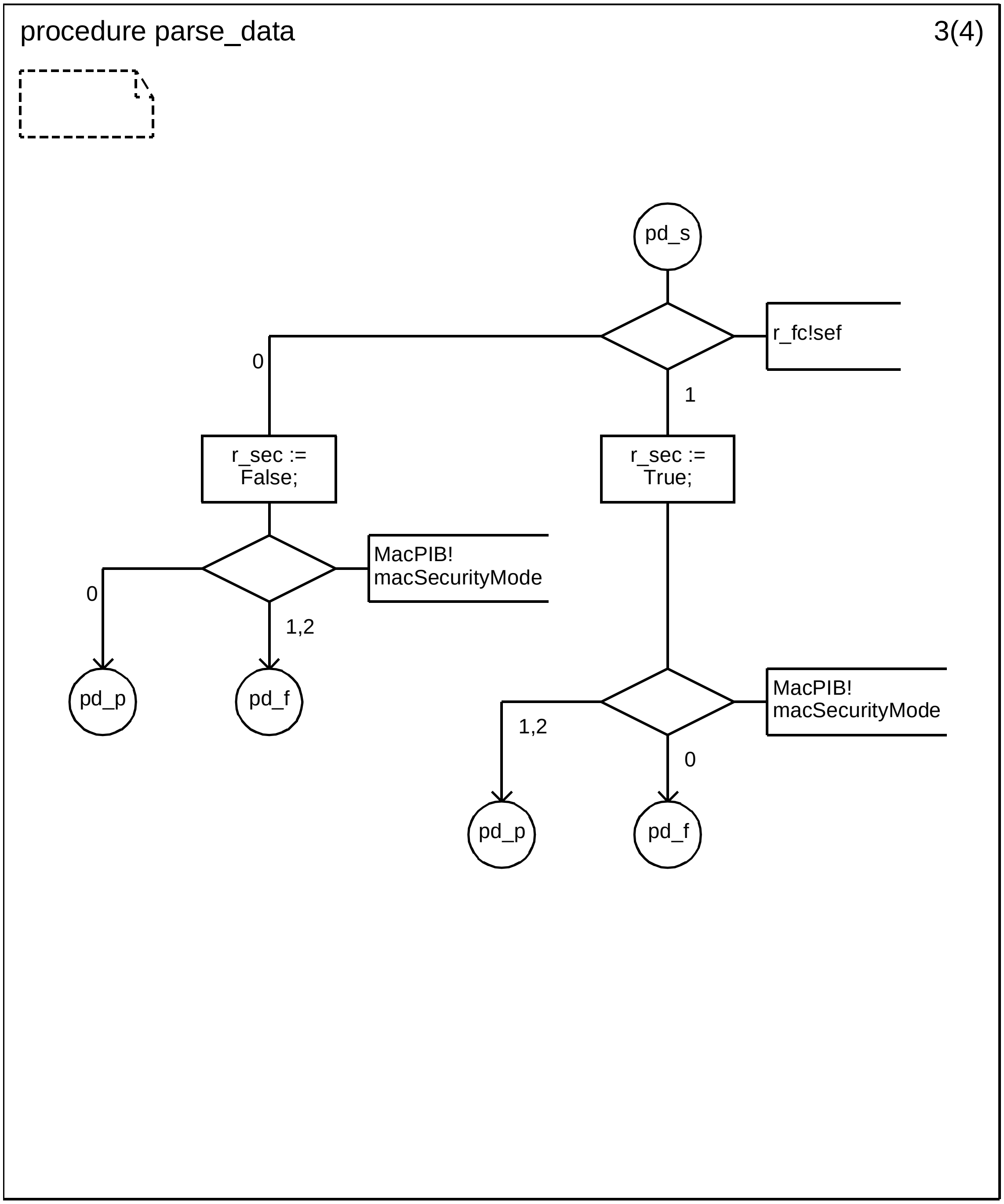
D.3.1.154.82.30 Procedure parse_data (2)

.



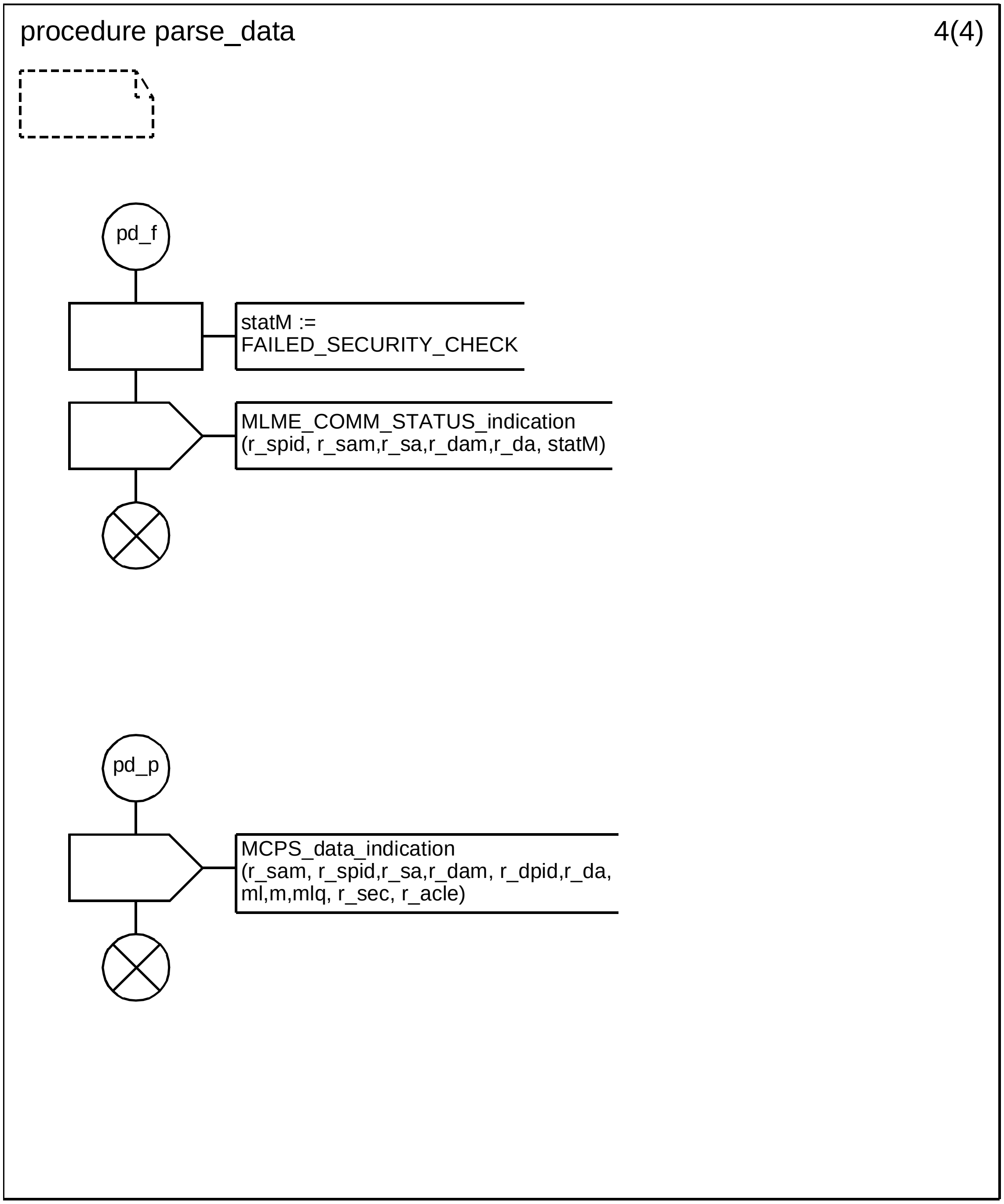
D.3.1.154.82.31 Procedure parse_data (3)

.

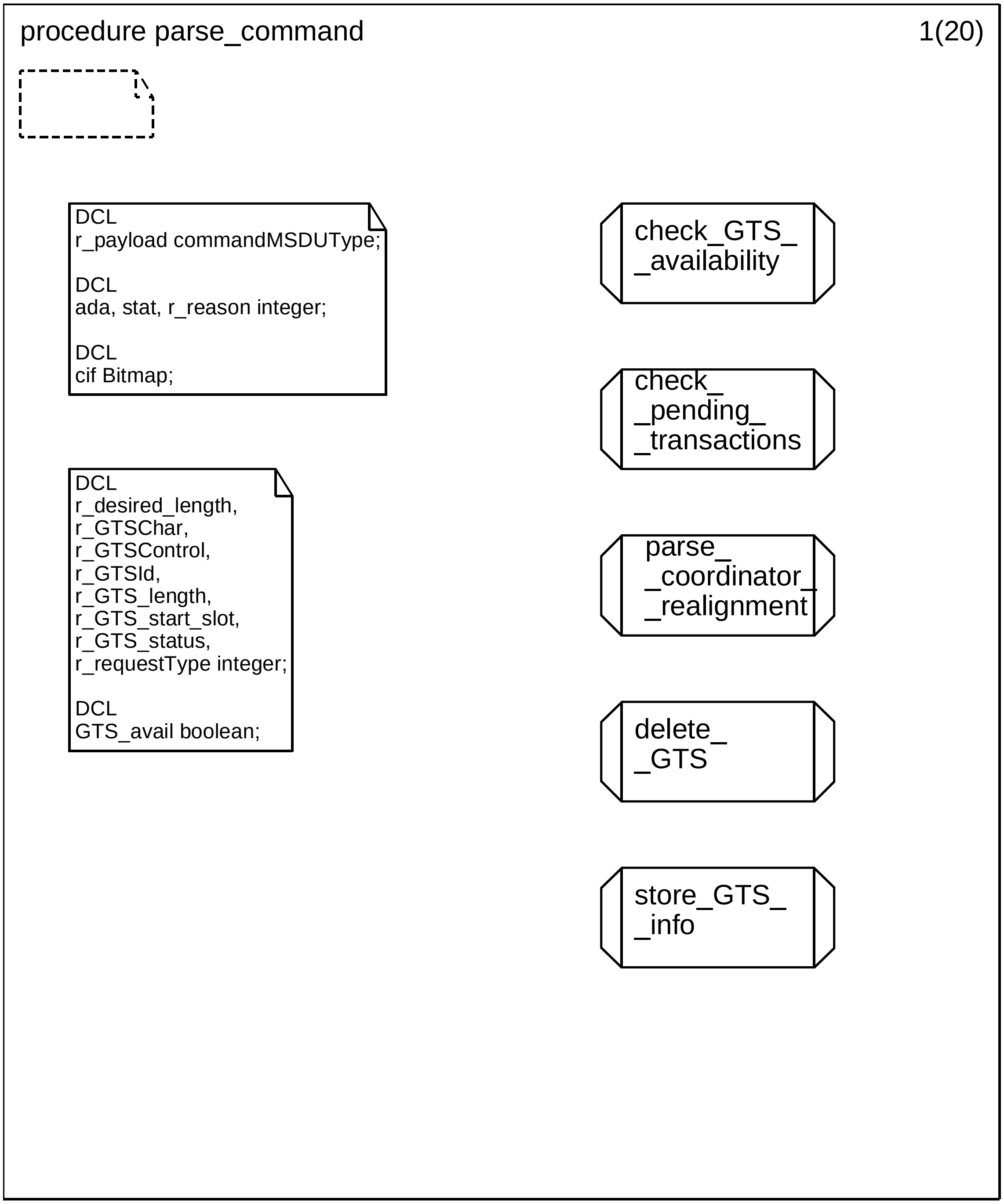


D.3.1.154.82.32 Procedure parse_data (4)

.



D.3.1.154.82.33 Procedure parse_command (1)

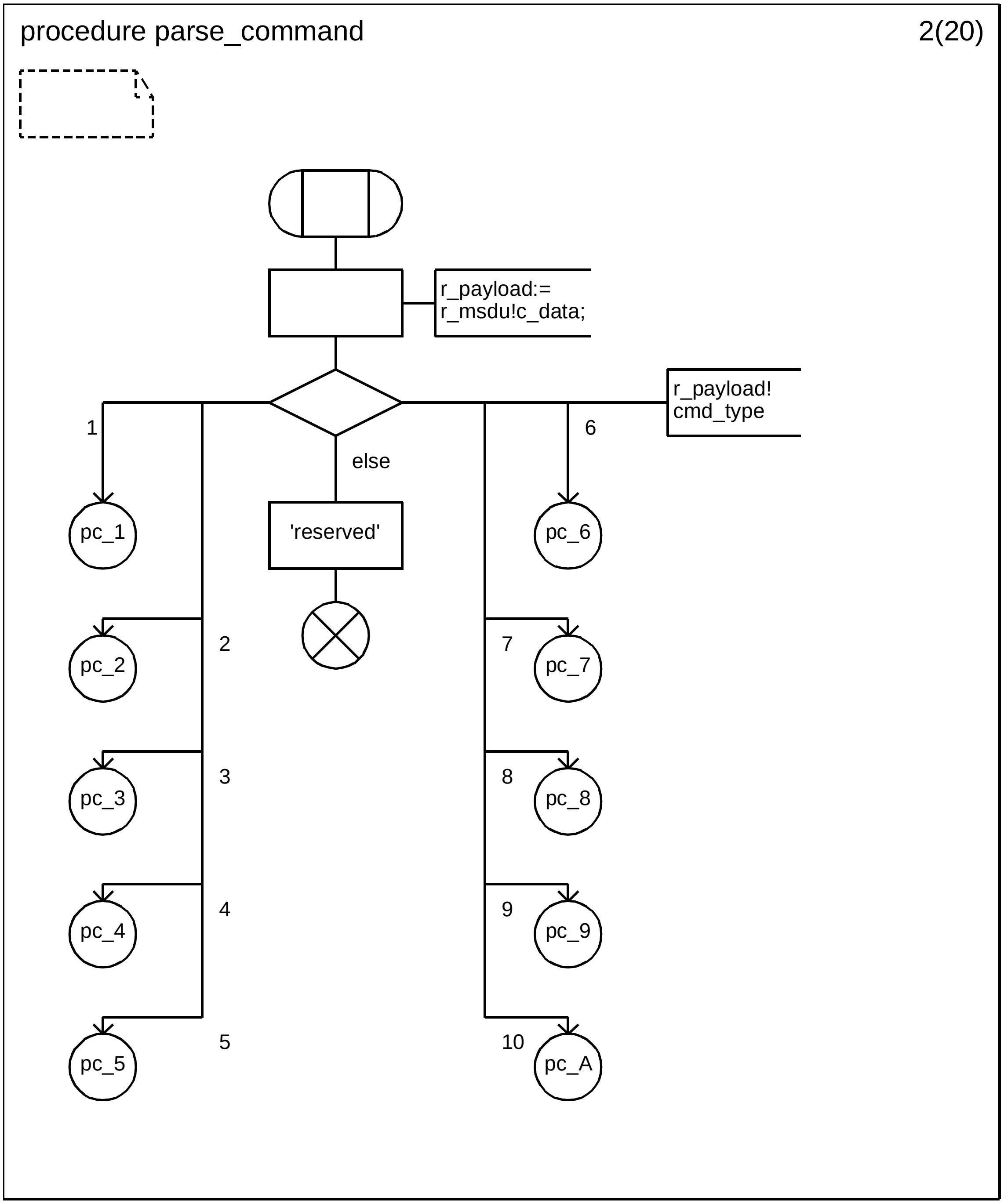


```
DCL
r_payload commandMSDUType;

DCL
ada, stat, r_reason integer;

DCL
cif Bitmap;
```

```
DCL
r_desired_length,
r_GTSChar,
r_GTSControl,
r_GTSId,
r_GTS_length,
r_GTS_start_slot,
r_GTS_status,
r_requestType integer;

DCL
GTS_avail boolean;
```

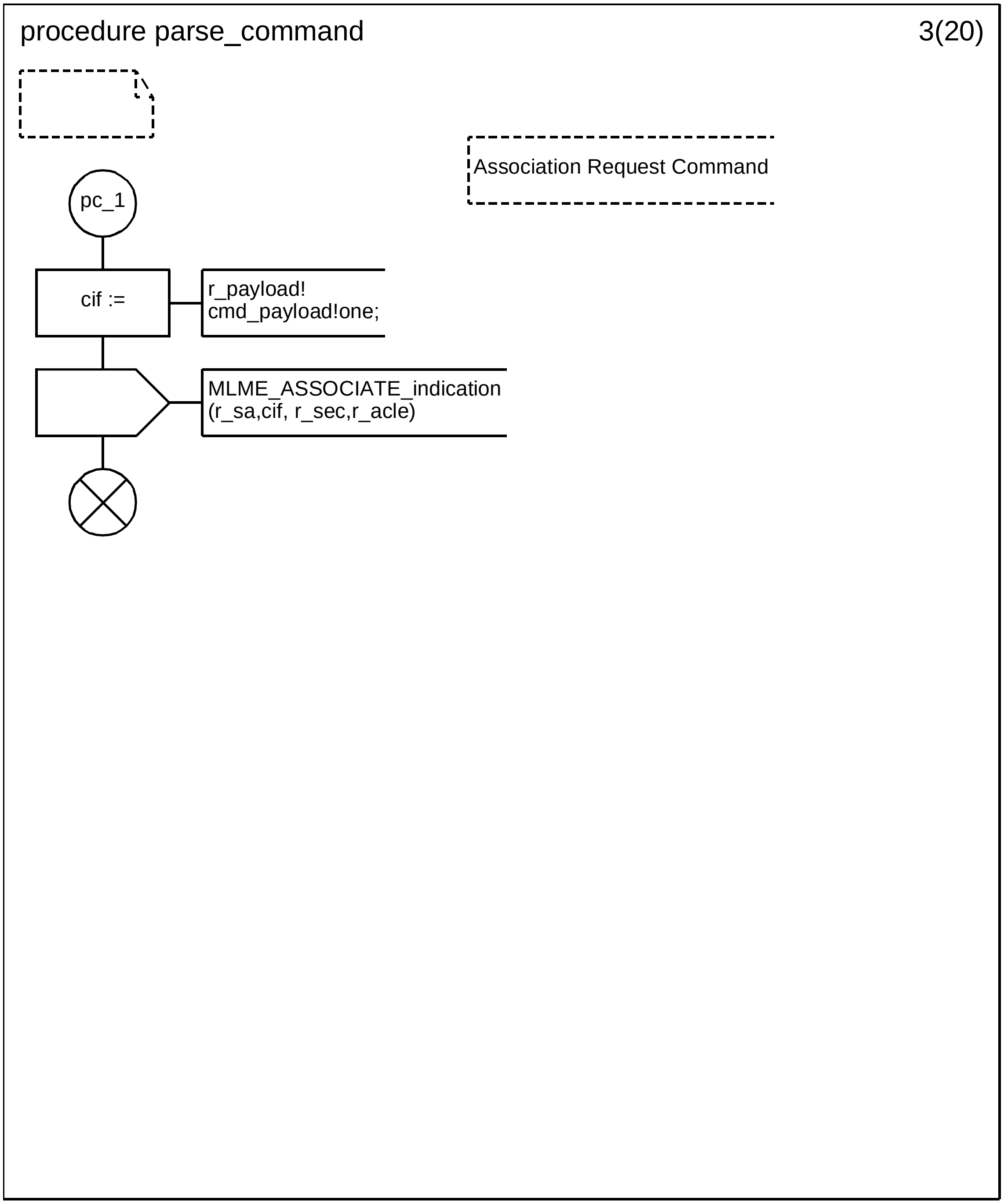
D.3.1.154.82.34 Procedure parse_command (2)

.

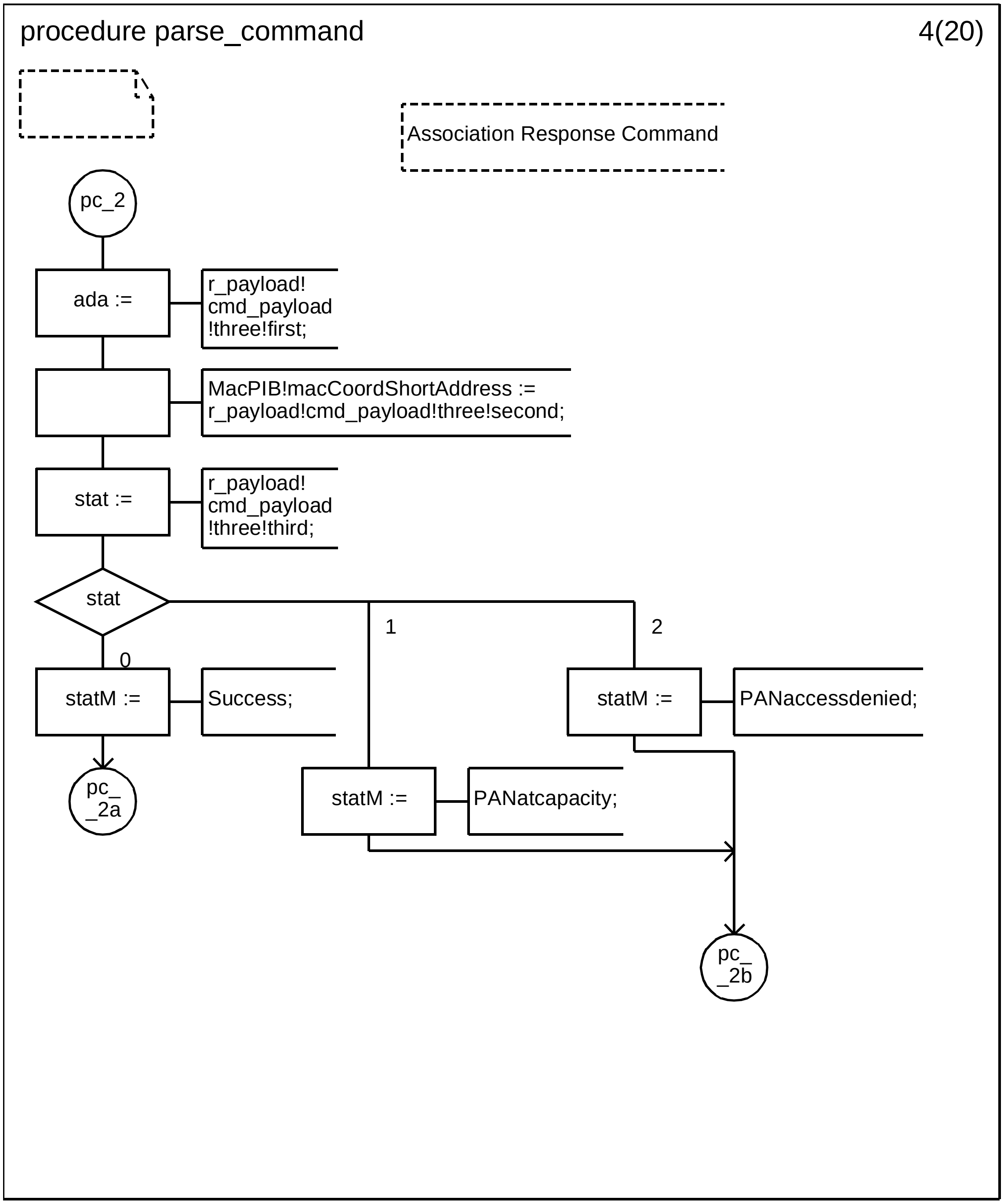


D.3.1.154.82.35 Procedure parse_command (3)

.

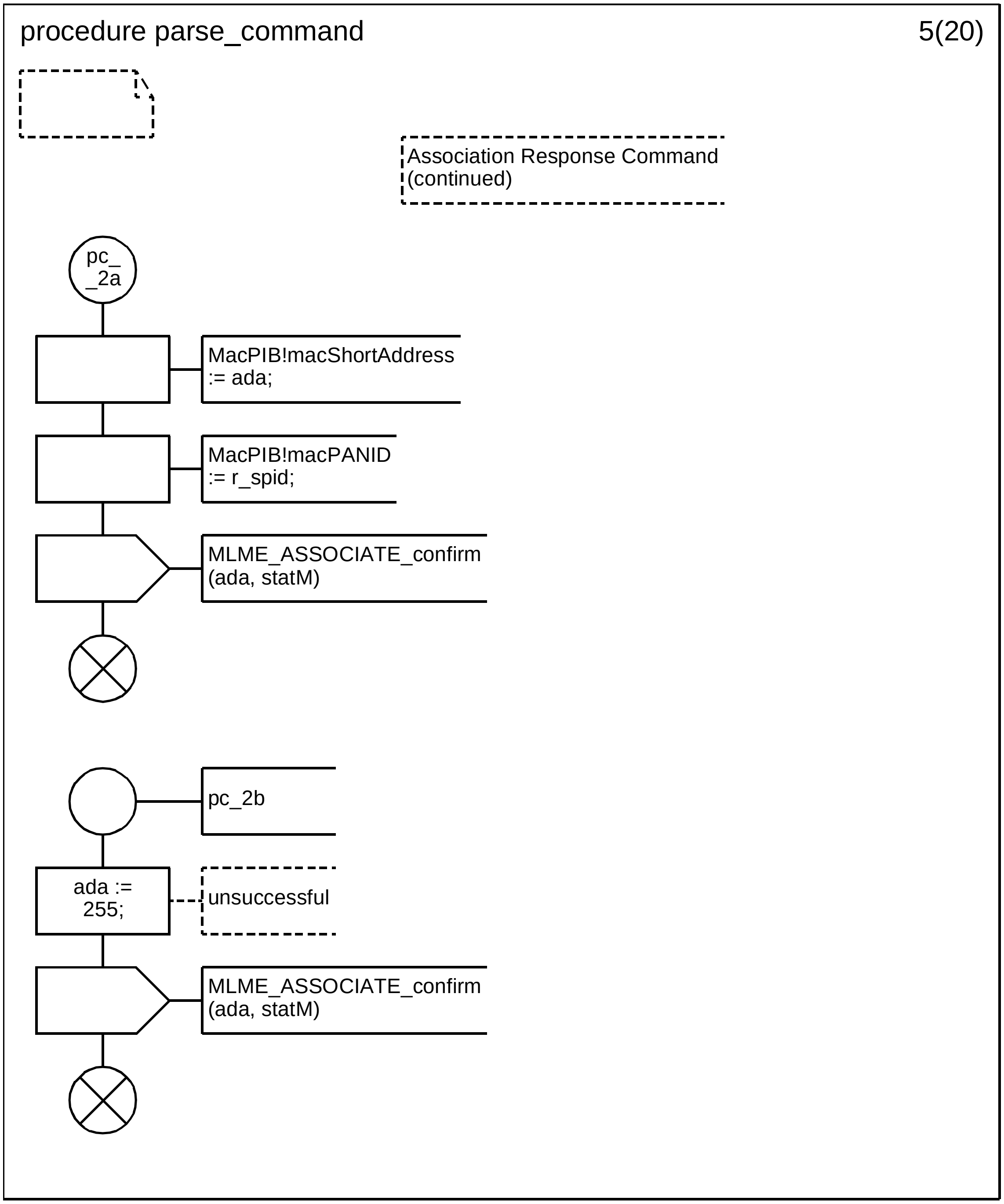


D.3.1.154.82.36 Procedure parse_command (4)



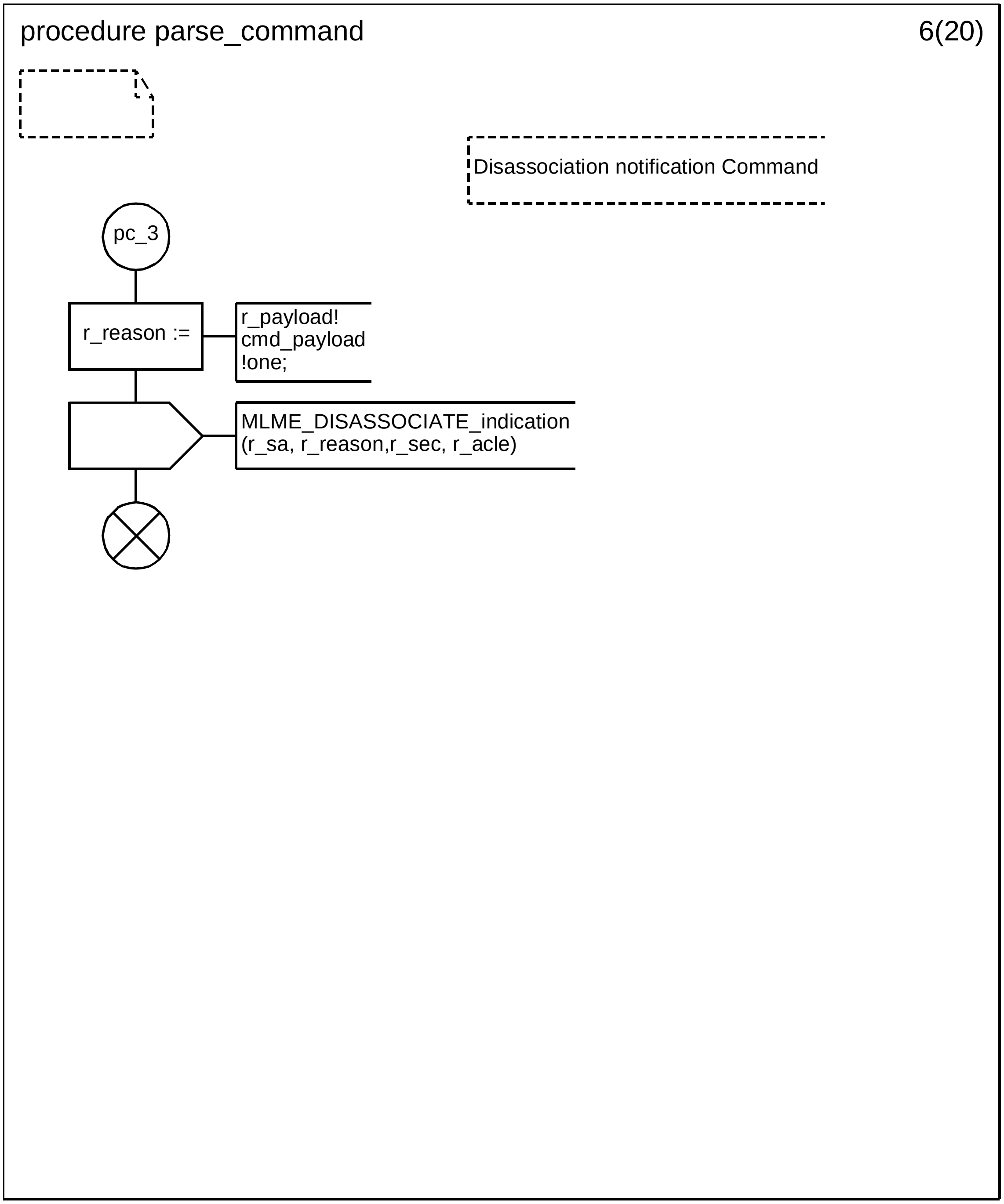
D.3.1.154.82.37 Procedure parse_command (5)

.



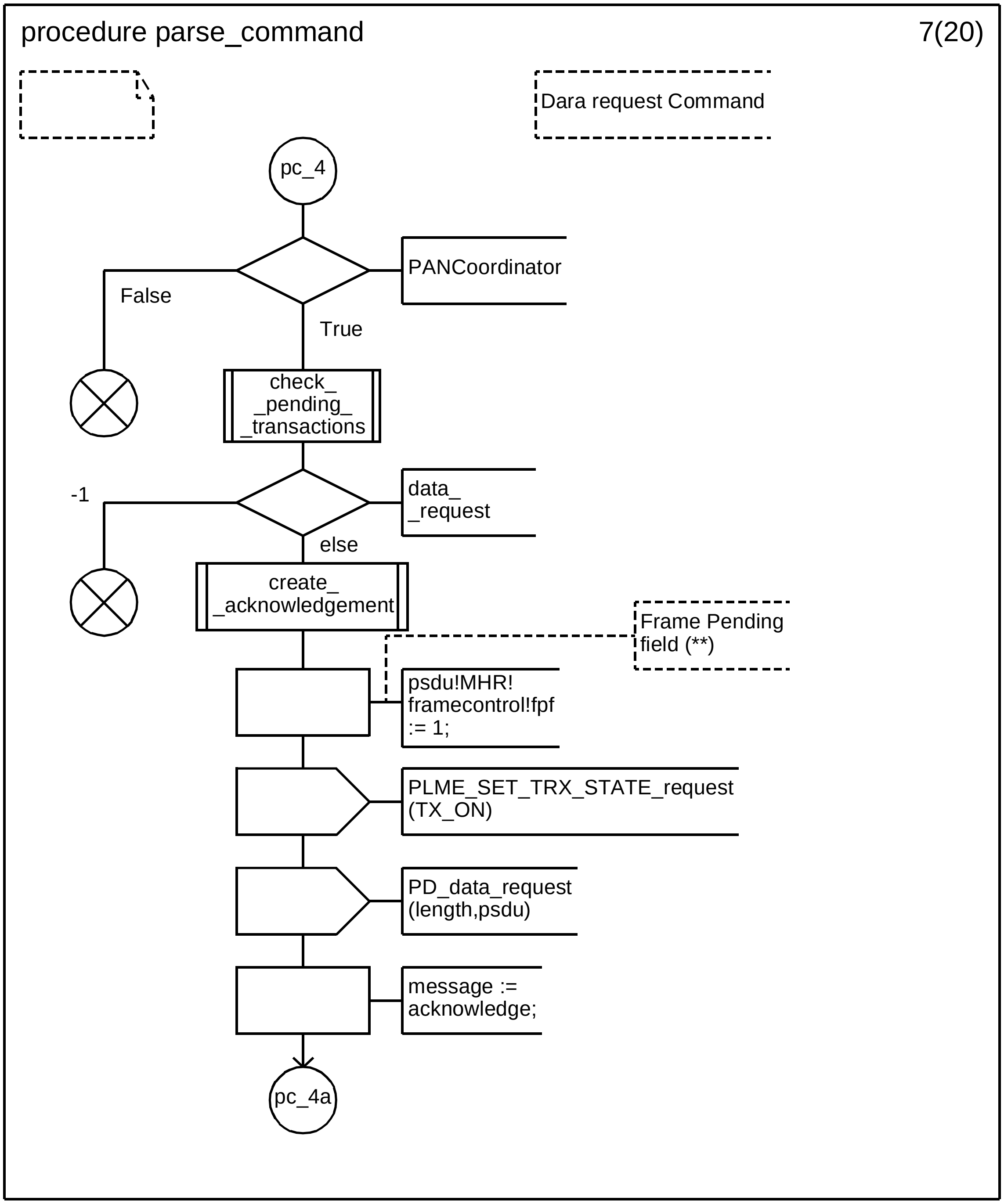
D.3.1.154.82.38 Procedure parse_command (6)

.


procedure parse_command
6(20)
Disassociation notification Command
pc_3
r_reason :=
r_payload!
cmd_payload
!one;
MLME_DISASSOCIATE_indication
(r_sa, r_reason,r_sec, r_acle)

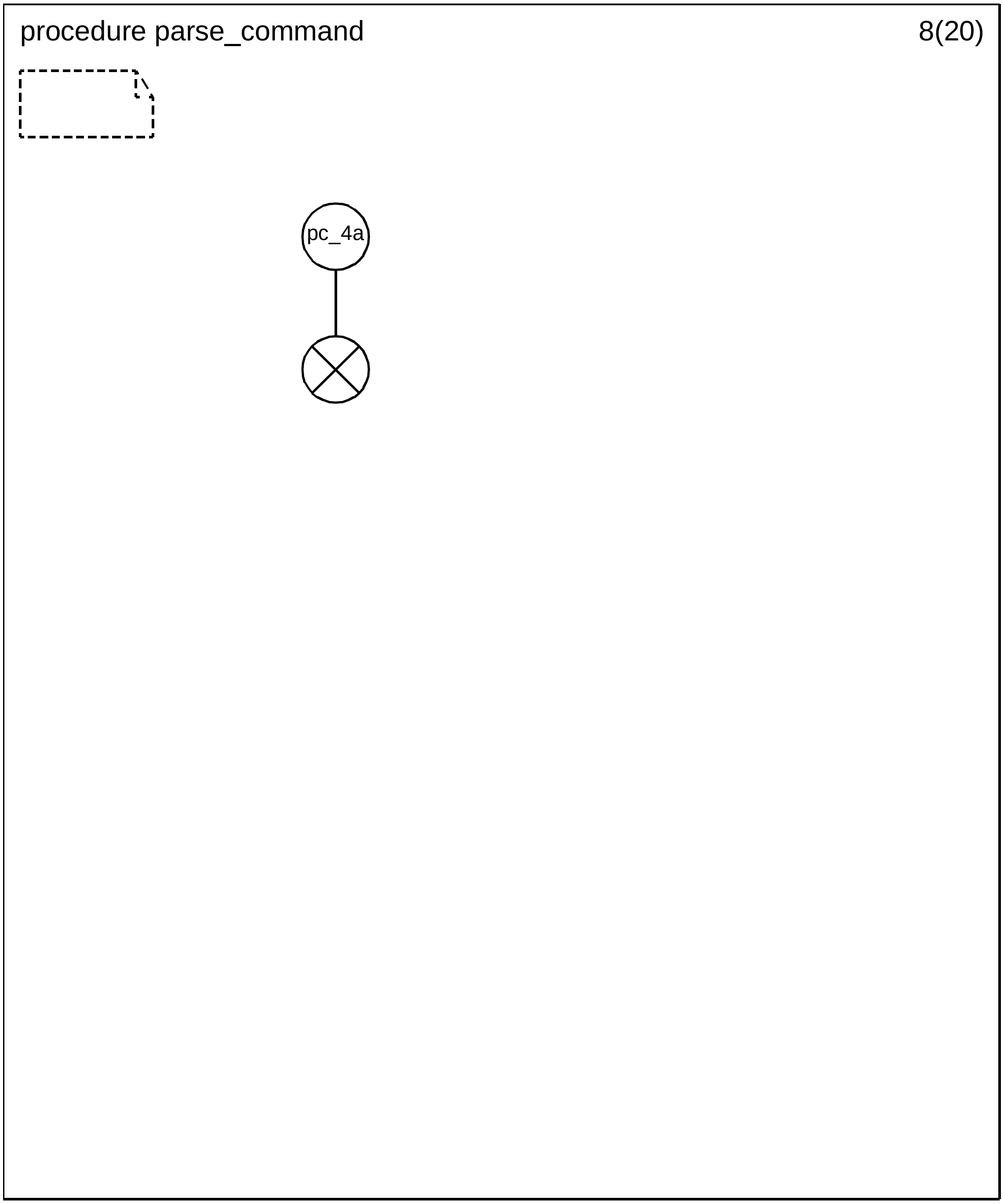
D.3.1.154.82.39 Procedure parse_command (7)

.



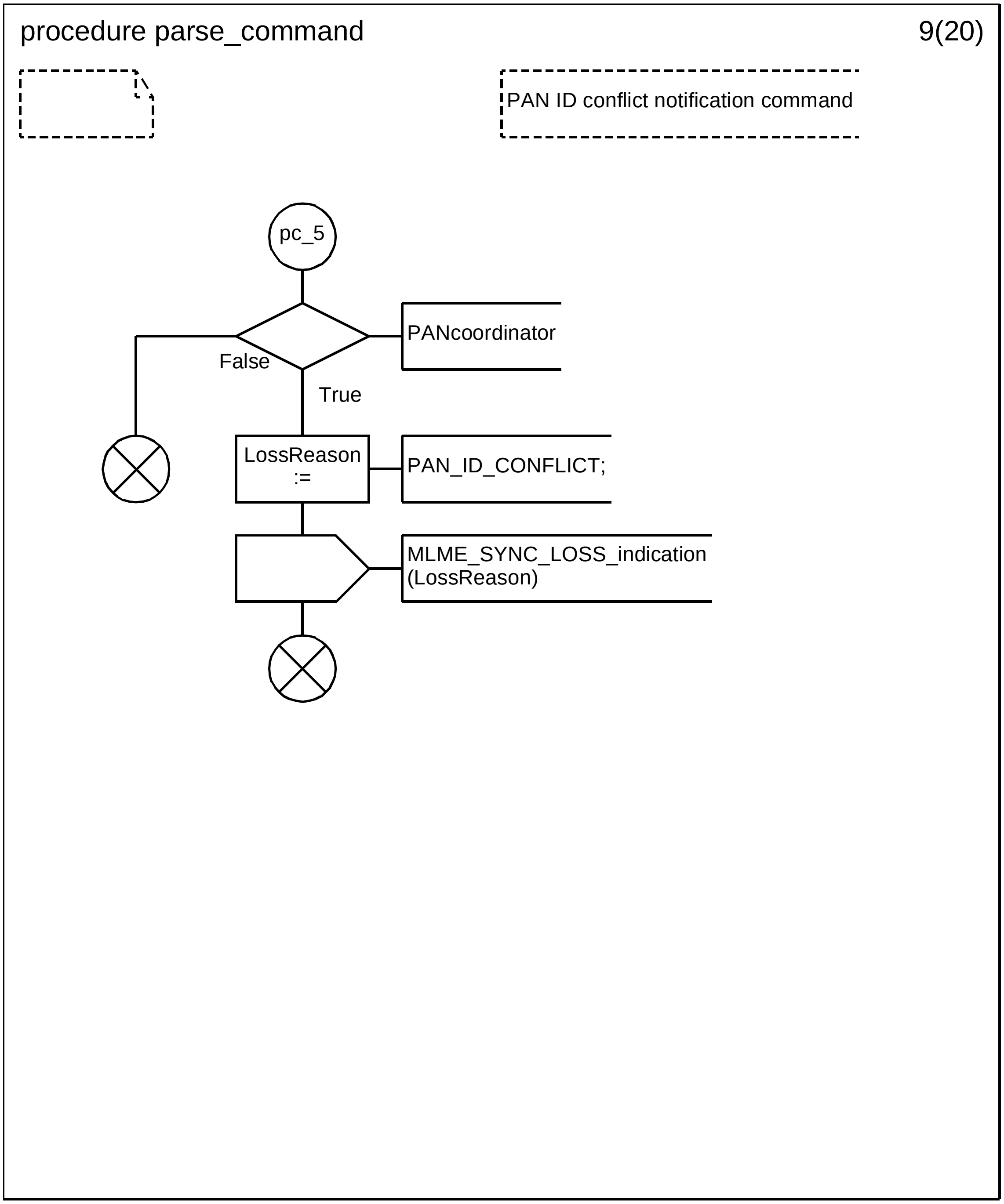
D.3.1.154.82.40 Procedure parse_command (8)

.



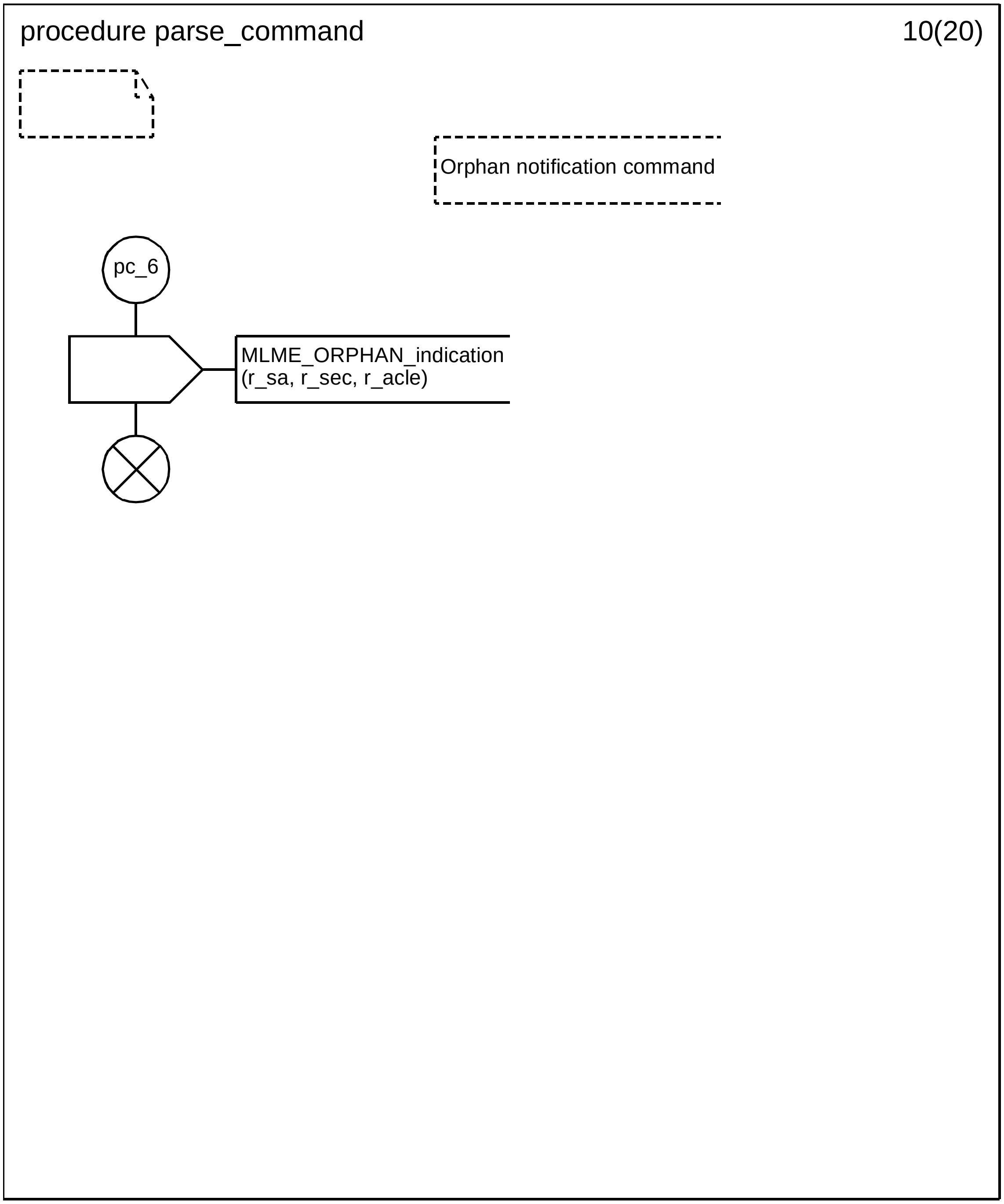
D.3.1.154.82.41 Procedure parse_command (9)

.



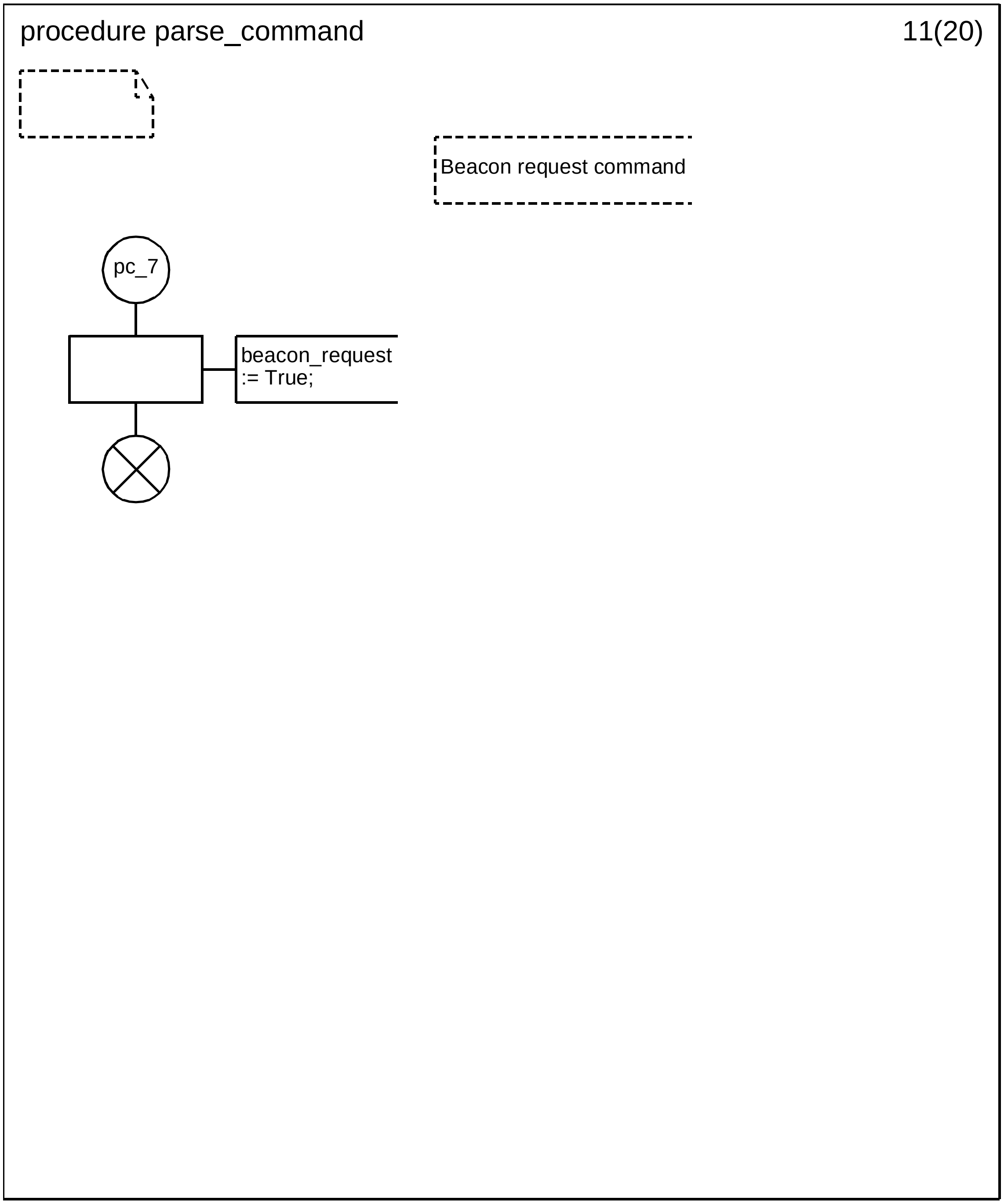
D.3.1.154.82.42 Procedure parse_command (10)

.

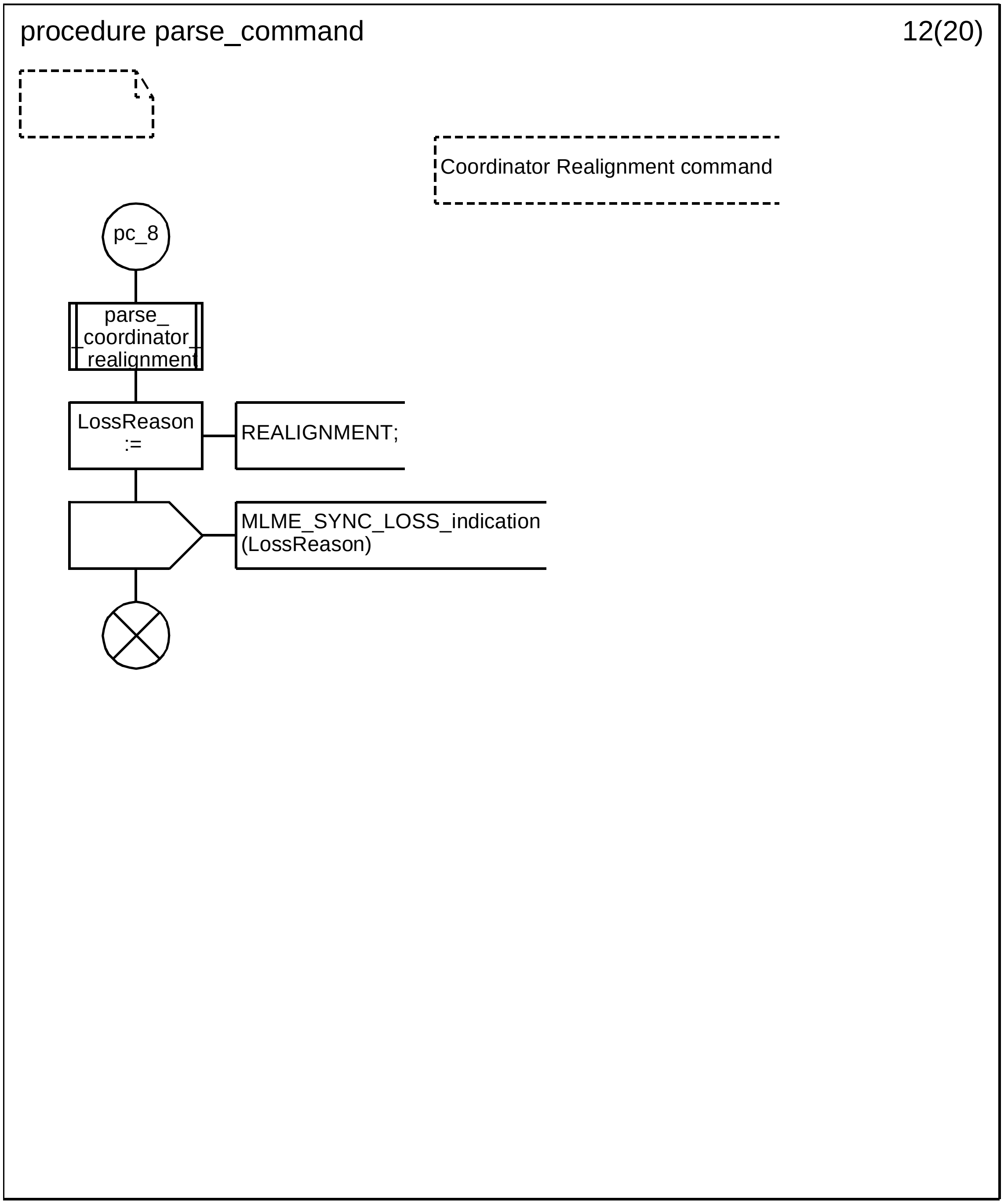


D.3.1.154.82.43 Procedure parse_command (11)

.

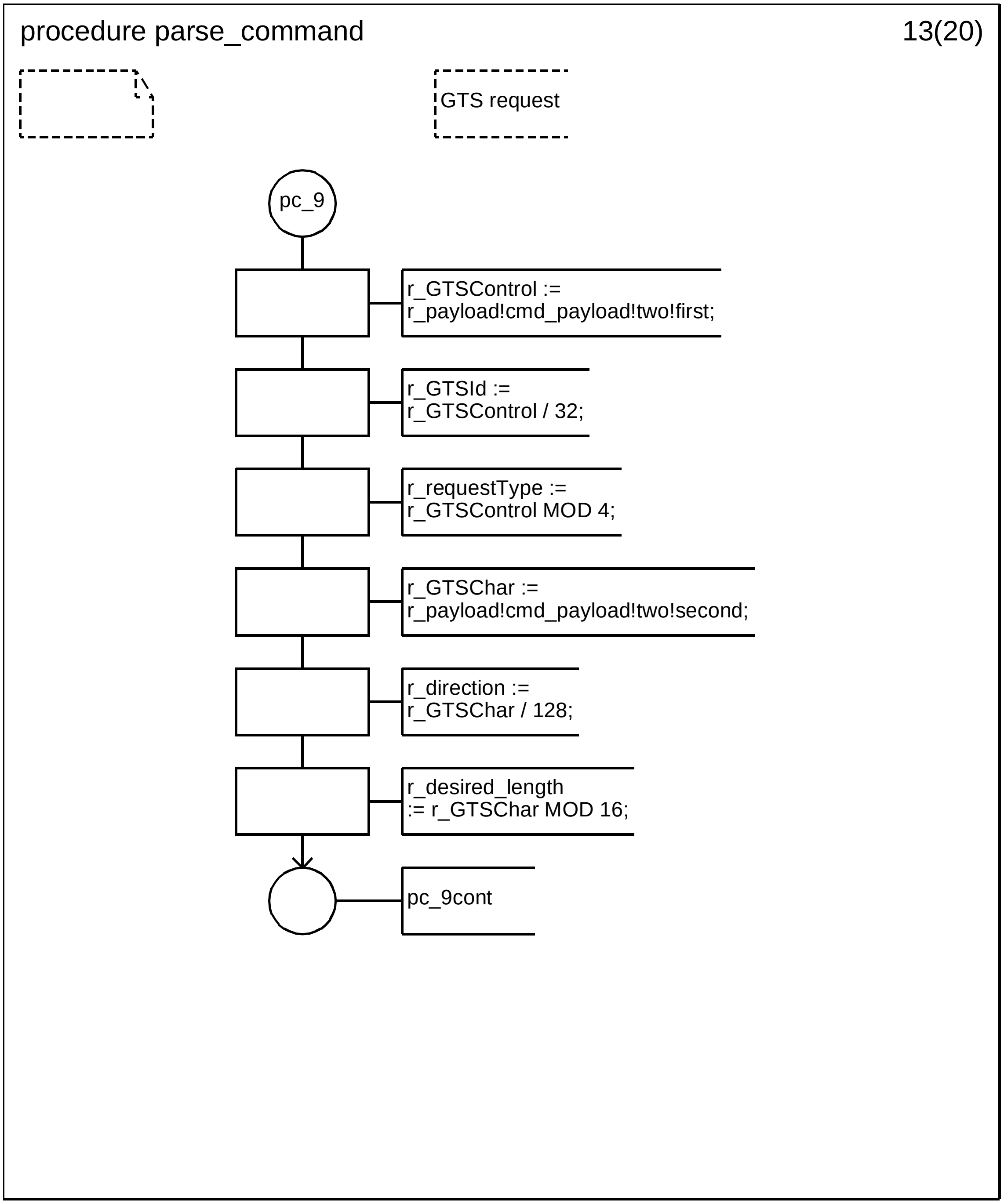


D.3.1.154.82.44 Procedure parse_command (12)



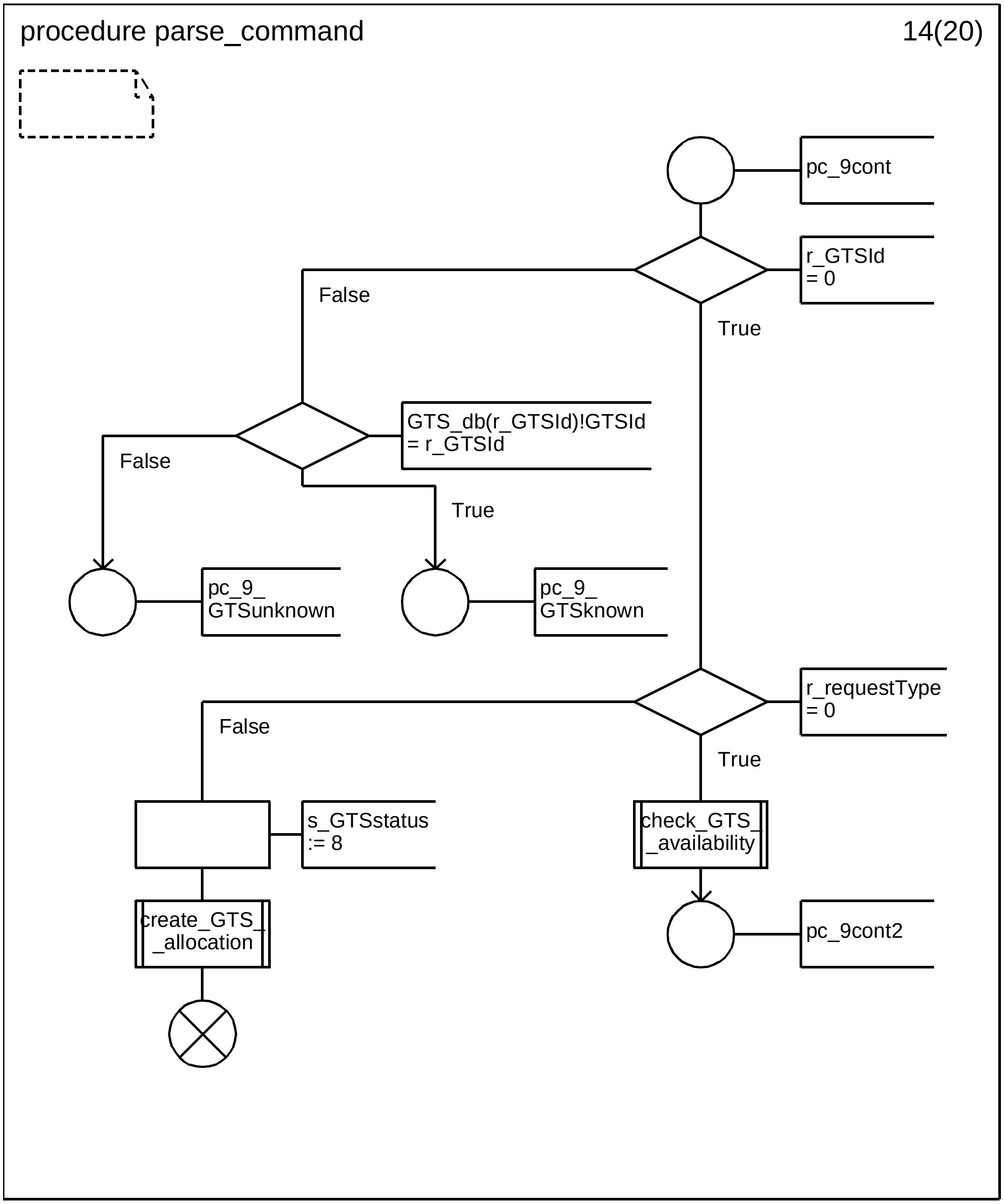
D.3.1.154.82.45 Procedure parse_command (13)

.



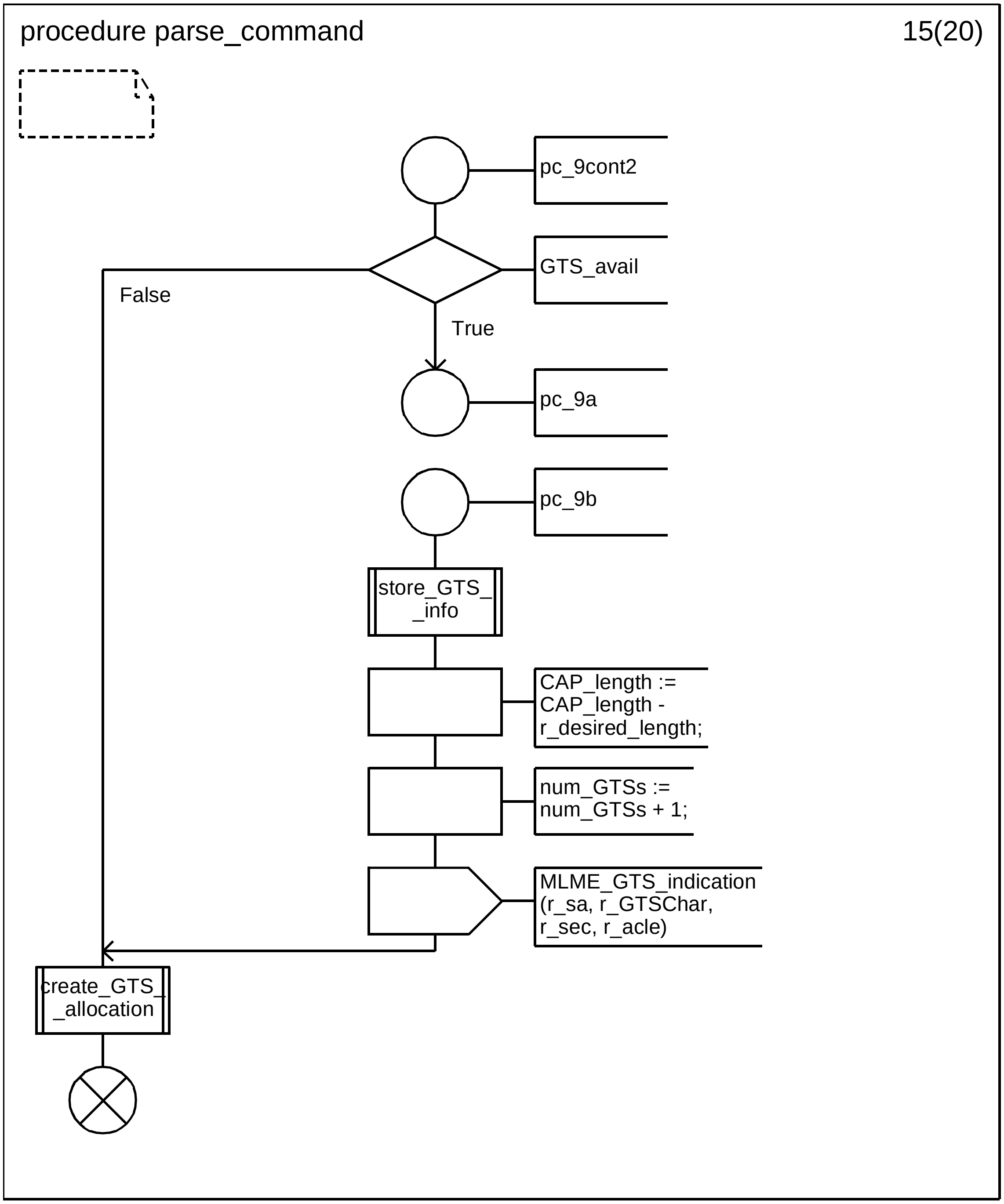
D.3.1.154.82.46 Procedure parse_command (14)

.



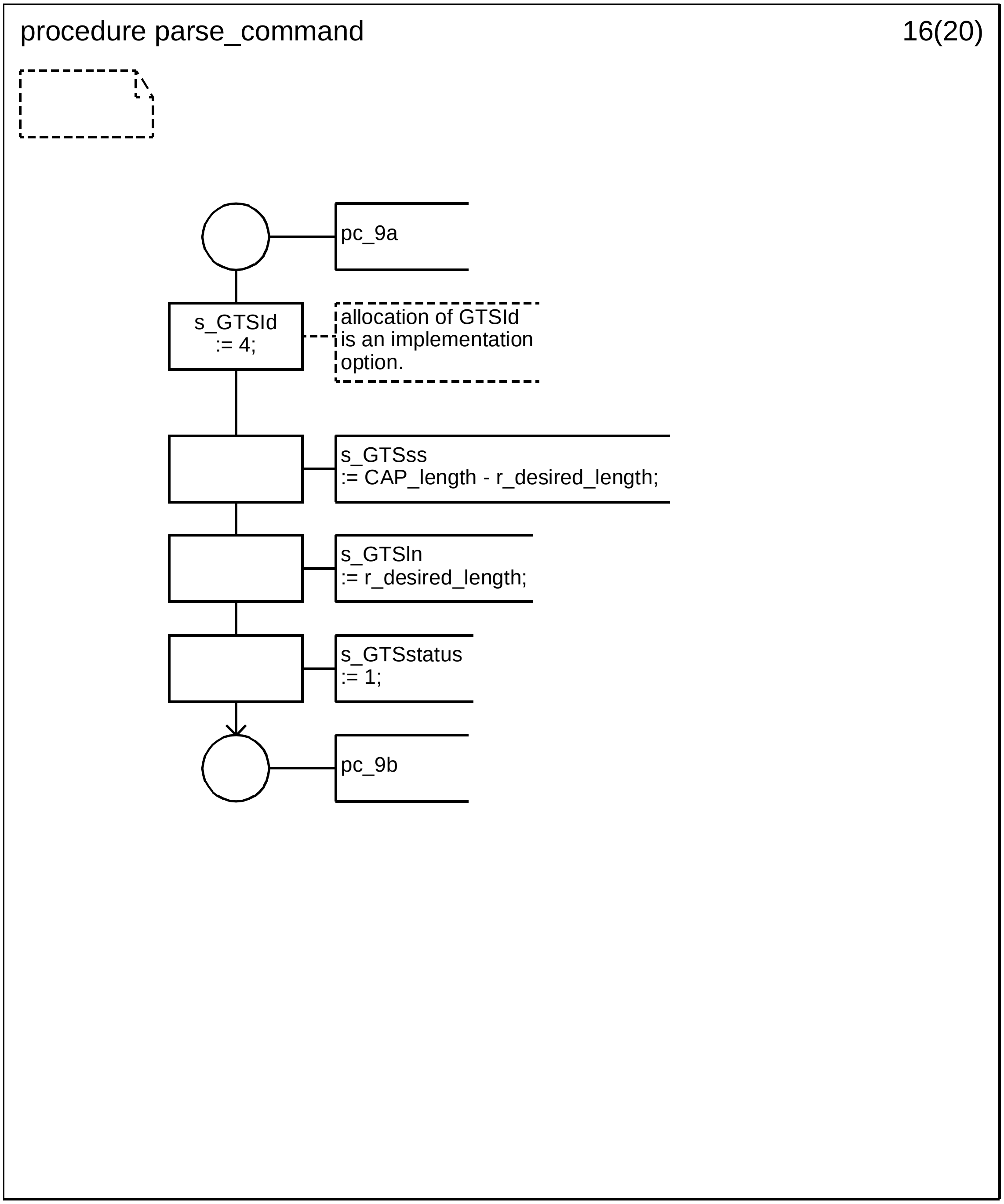
D.3.1.154.82.47 Procedure parse_command (15)

.



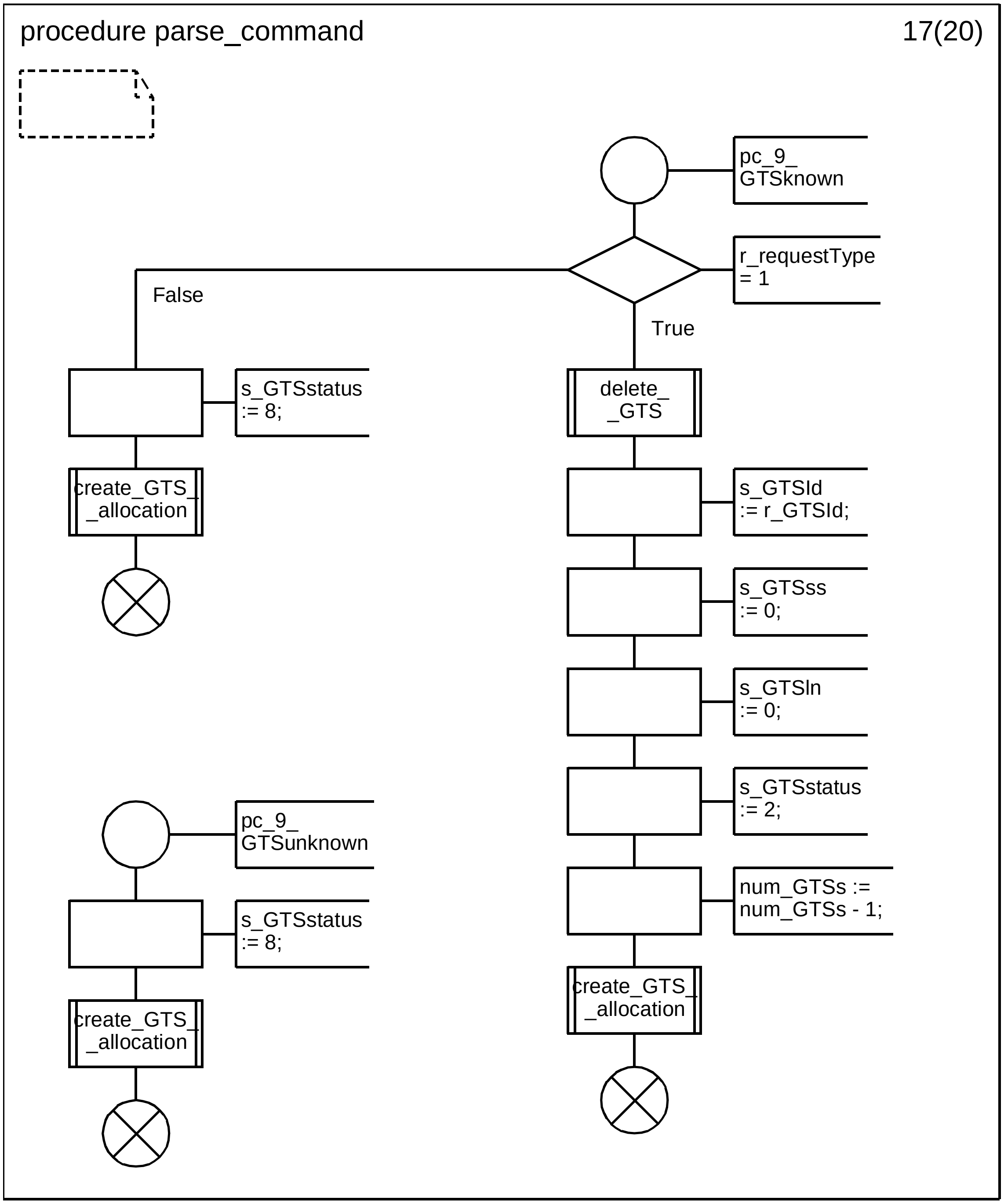
D.3.1.154.82.48 Procedure parse_command (16)

.



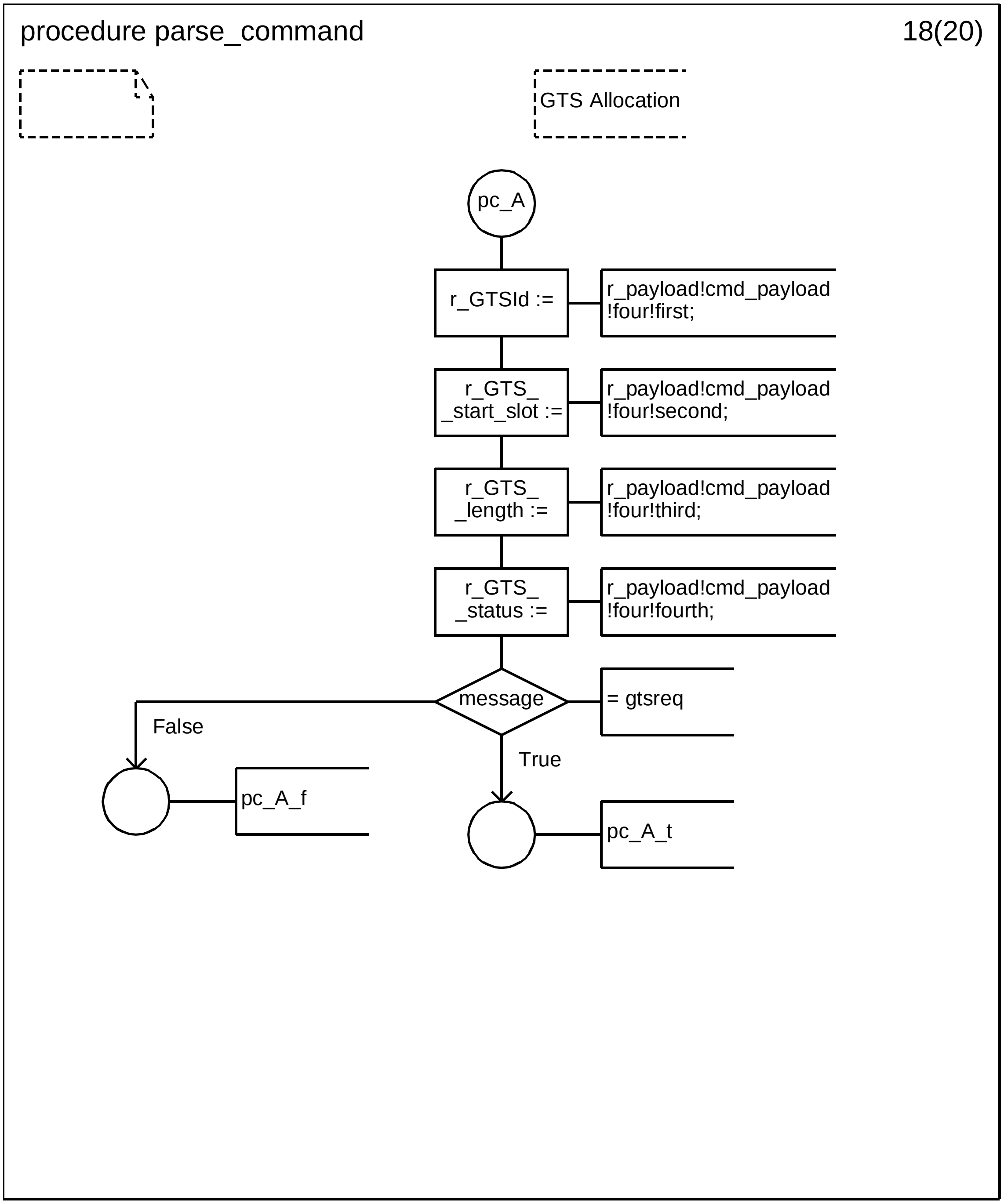
D.3.1.154.82.49 Procedure parse_command (17)

.



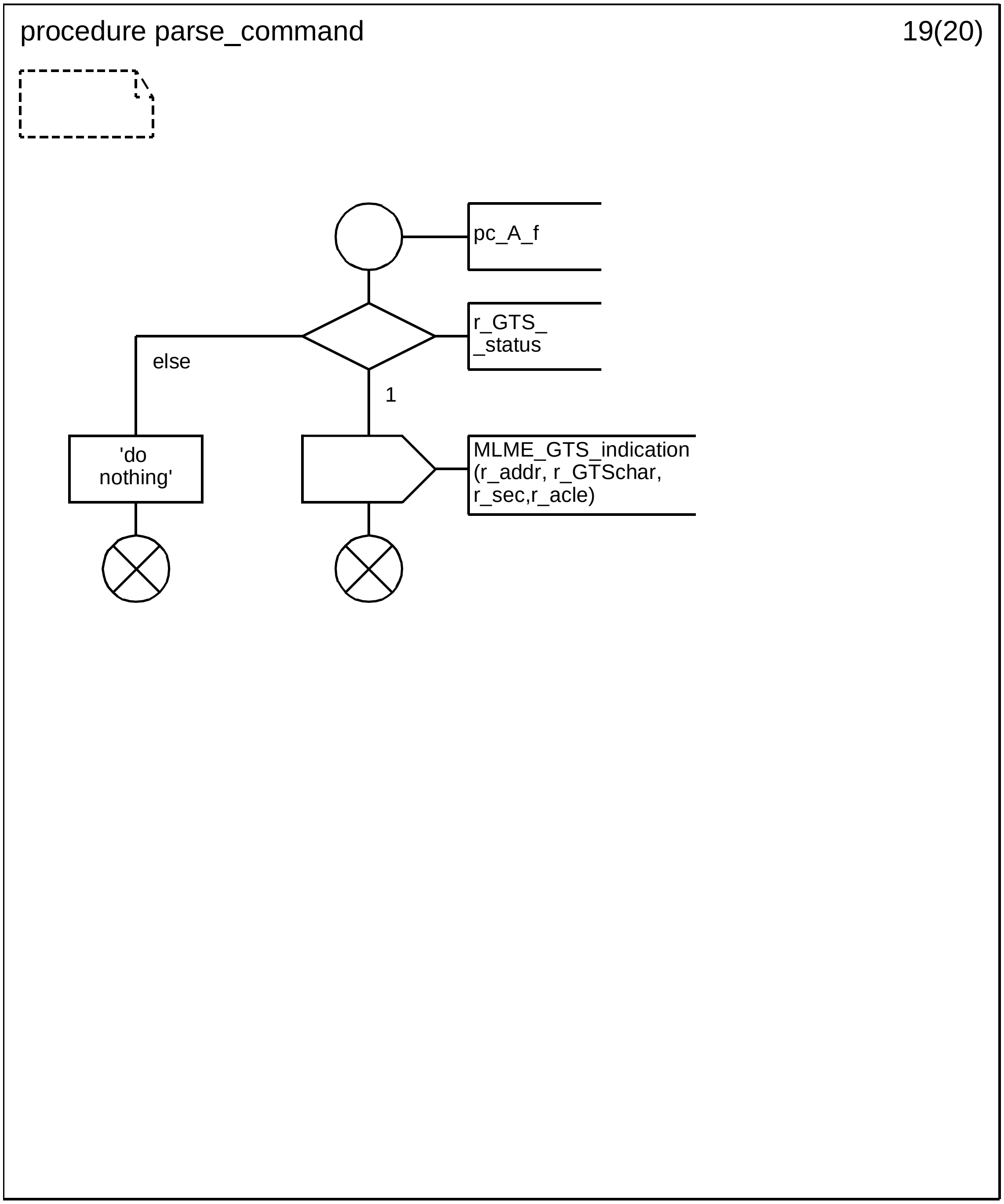
D.3.1.154.82.50 Procedure parse_command (18)

.



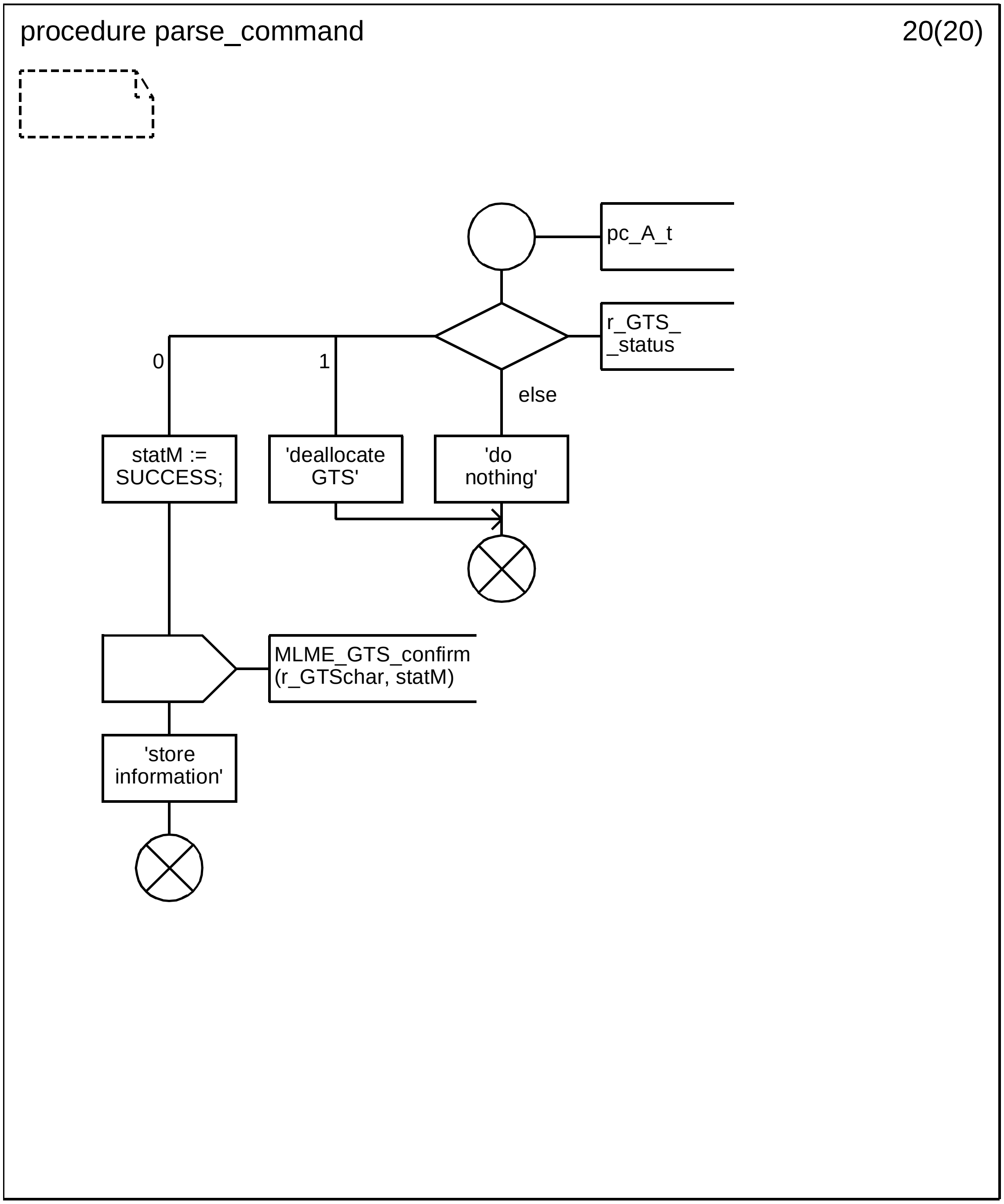
D.3.1.154.82.51 Procedure parse_command (19)

.



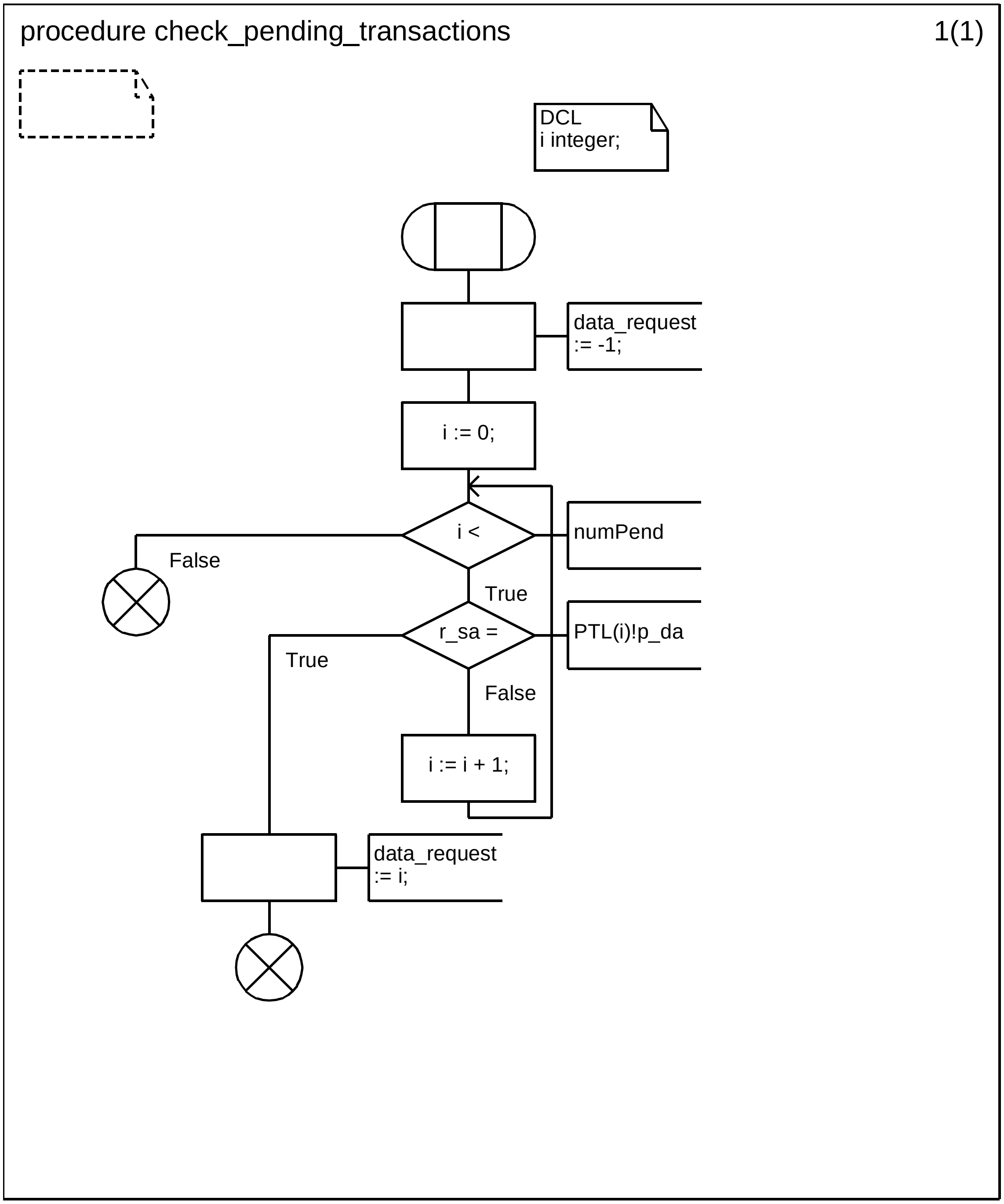
D.3.1.154.82.52 Procedure parse_command (20)

.



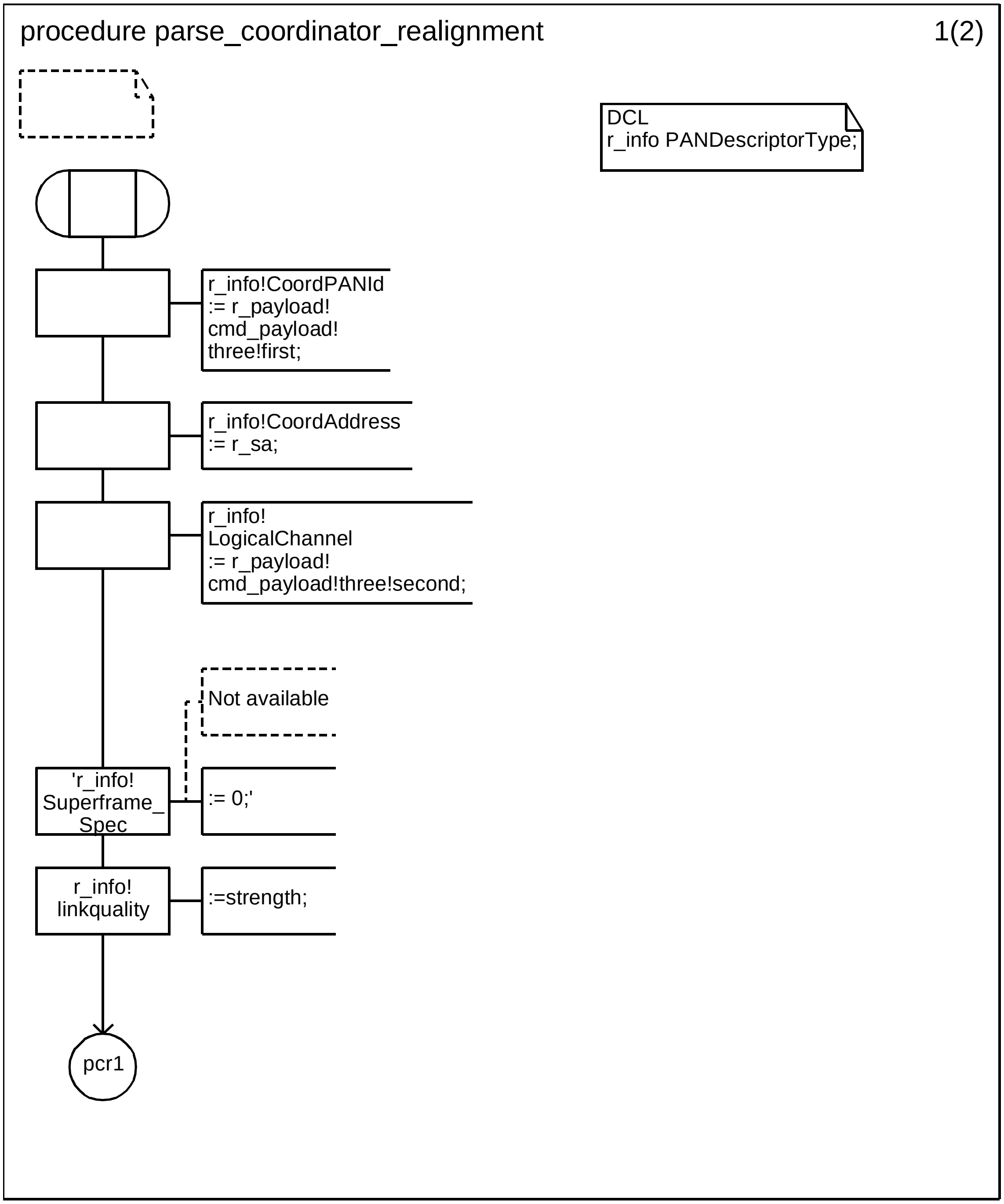
D.3.1.154.82.53 Procedure check_pending_transactions

.



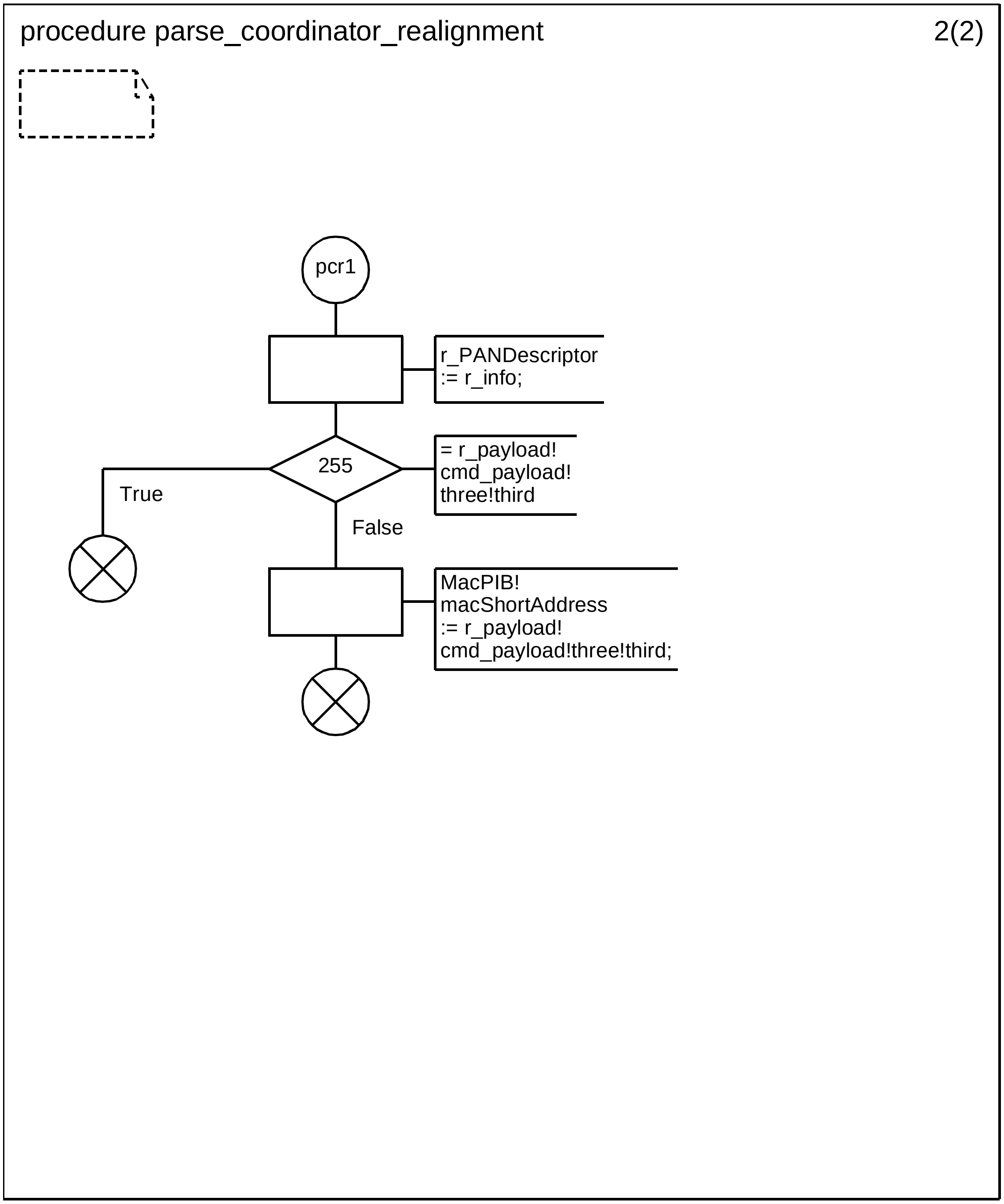
D.3.1.154.82.54 Procedure parse_coordinator_realignment (1)

.



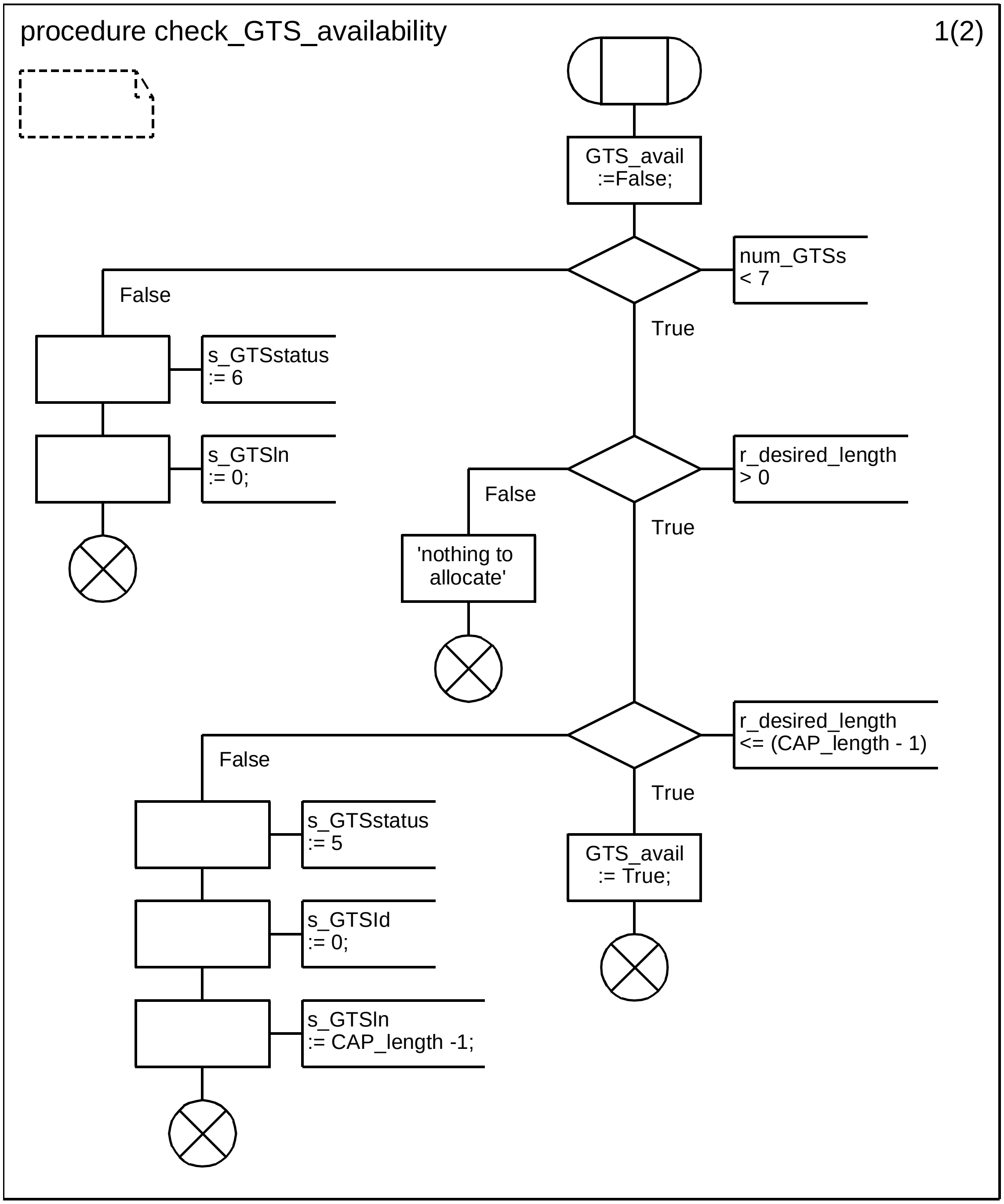
D.3.1.154.82.55 Procedure parse_coordinator_realignment (2)

.



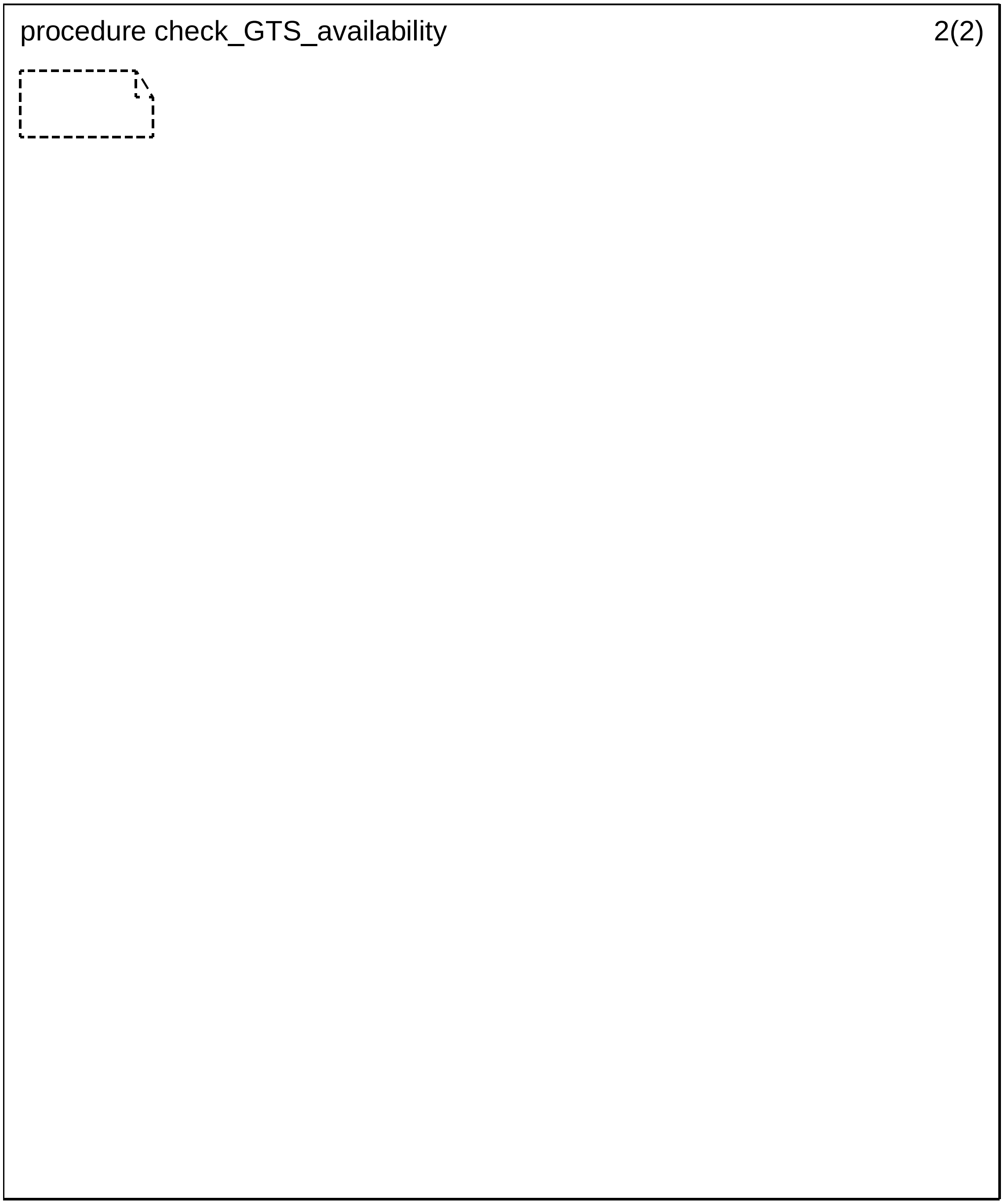
D.3.1.154.82.56 Procedure check_GTS_availability (1)

.



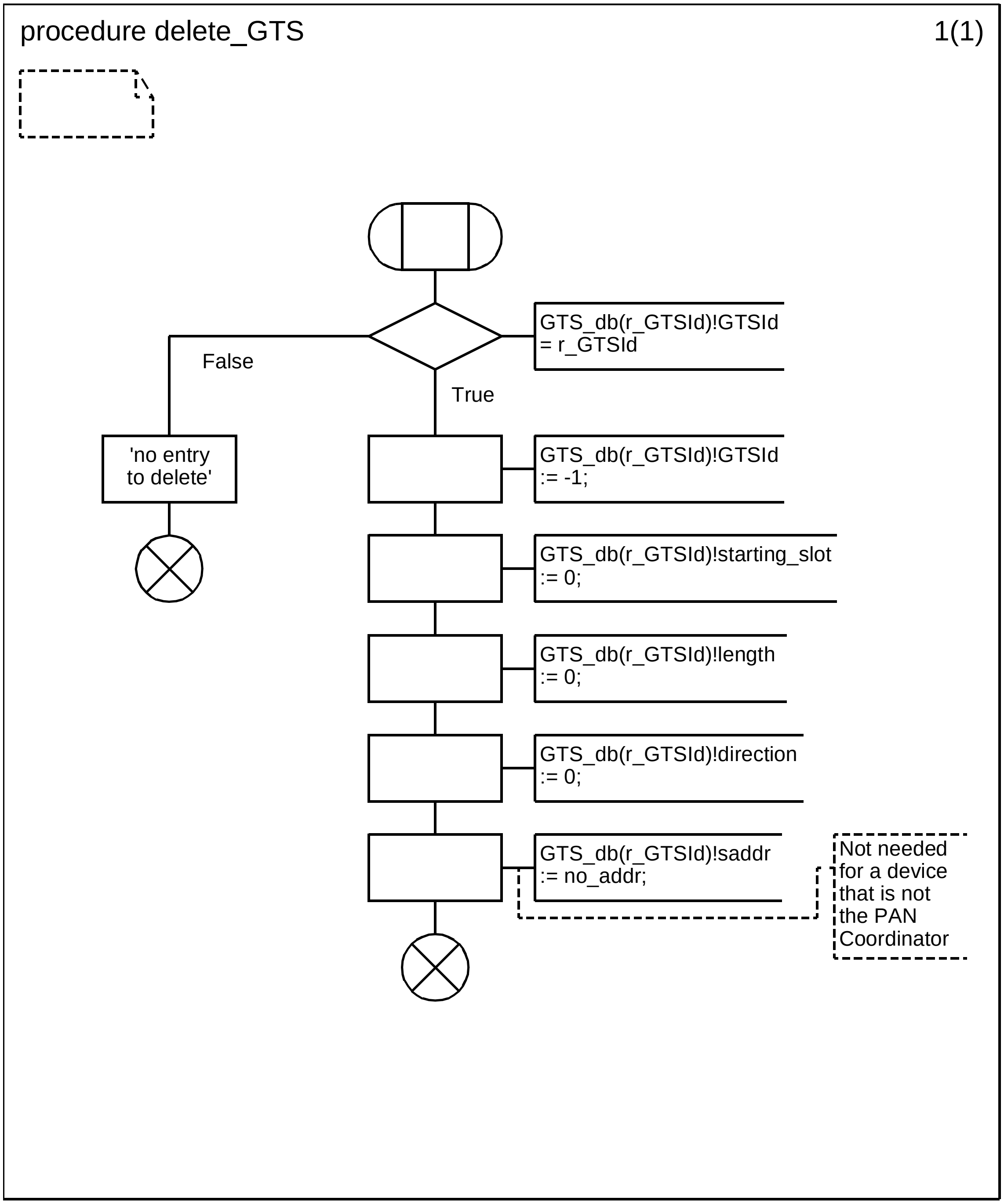
D.3.1.154.82.57 Procedure check_GTS_availability (2)

.



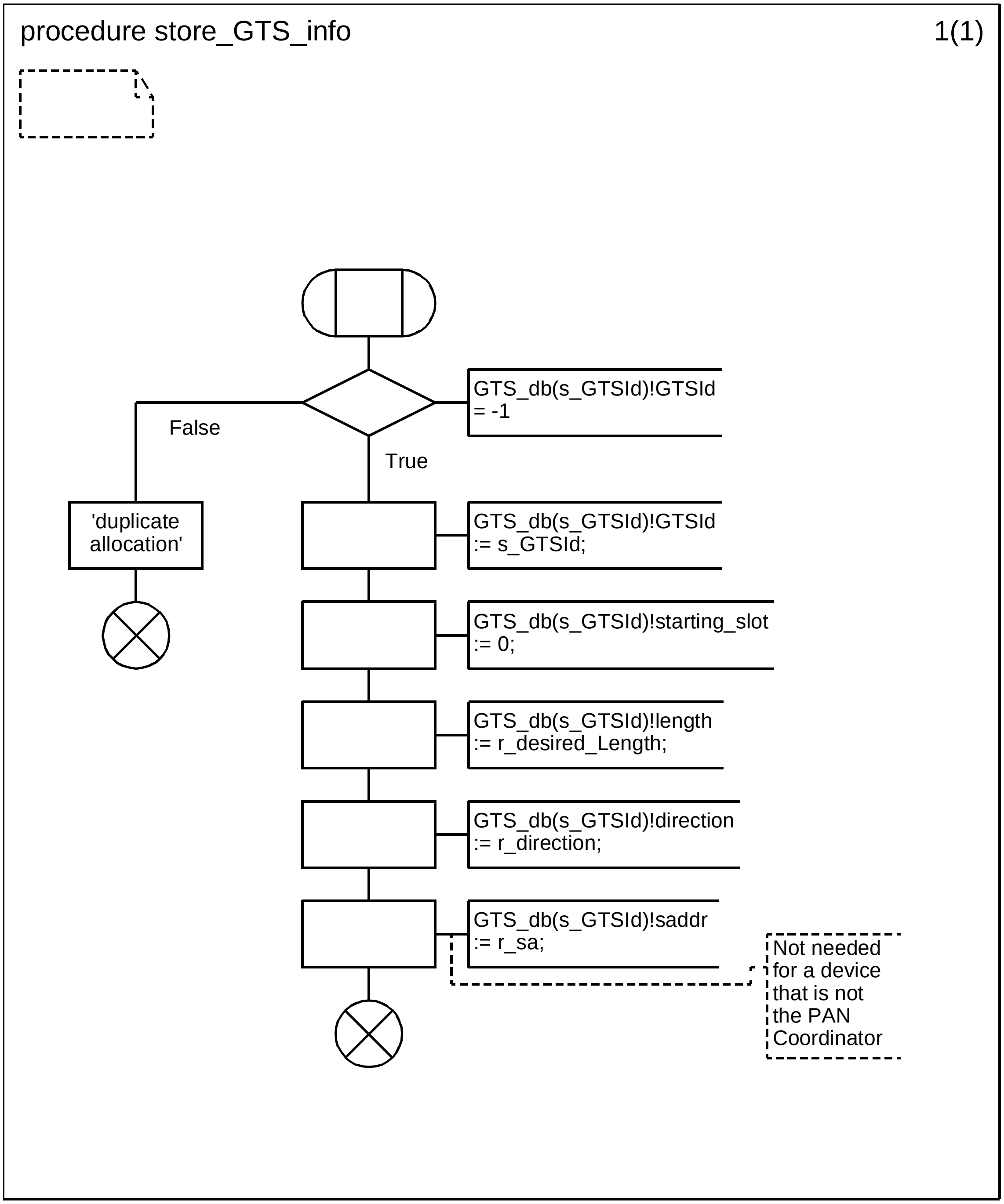
D.3.1.154.82.58 Procedure delete_GTS

.



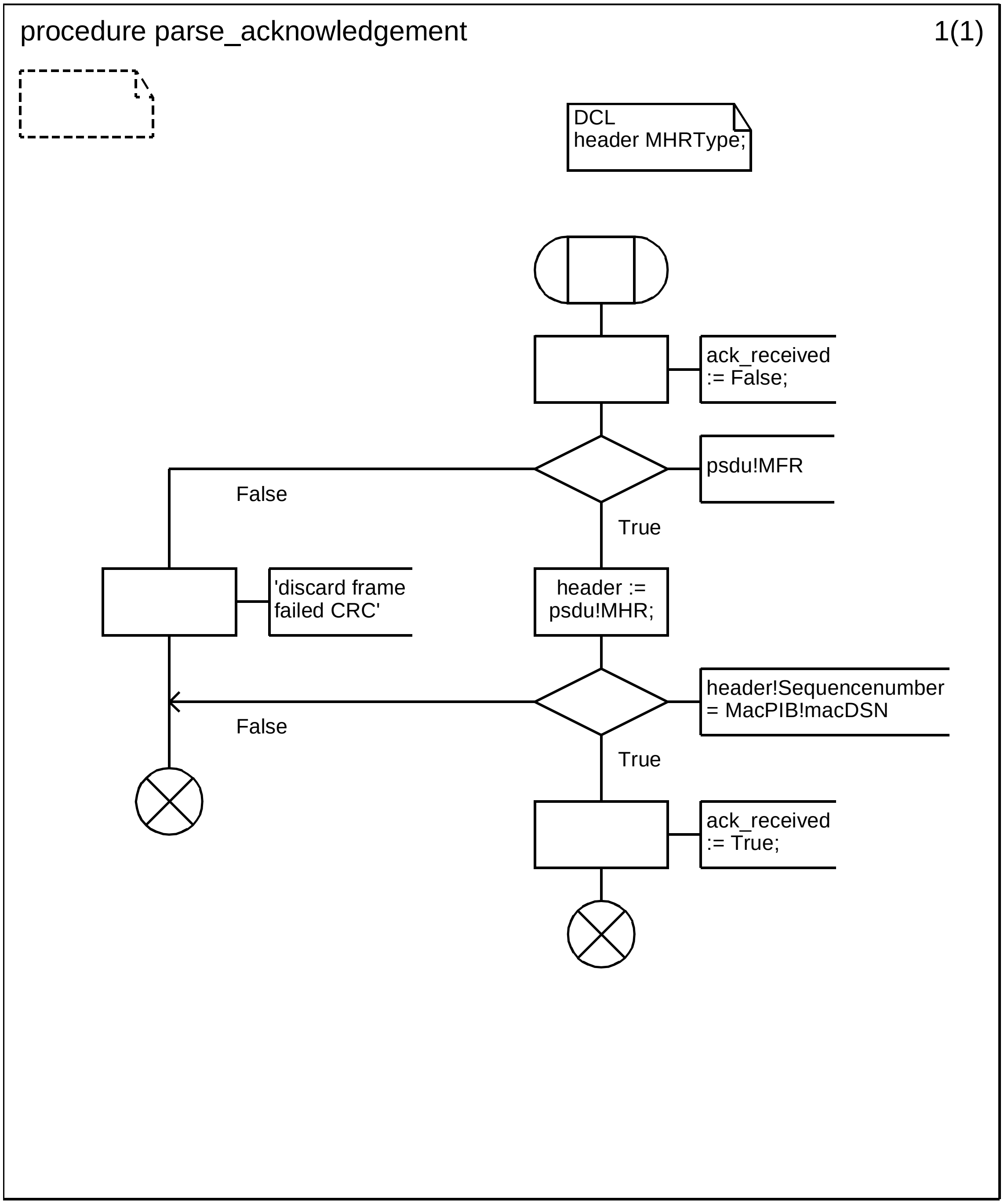
D.3.1.154.82.59 Procedure store_GTS_info

.



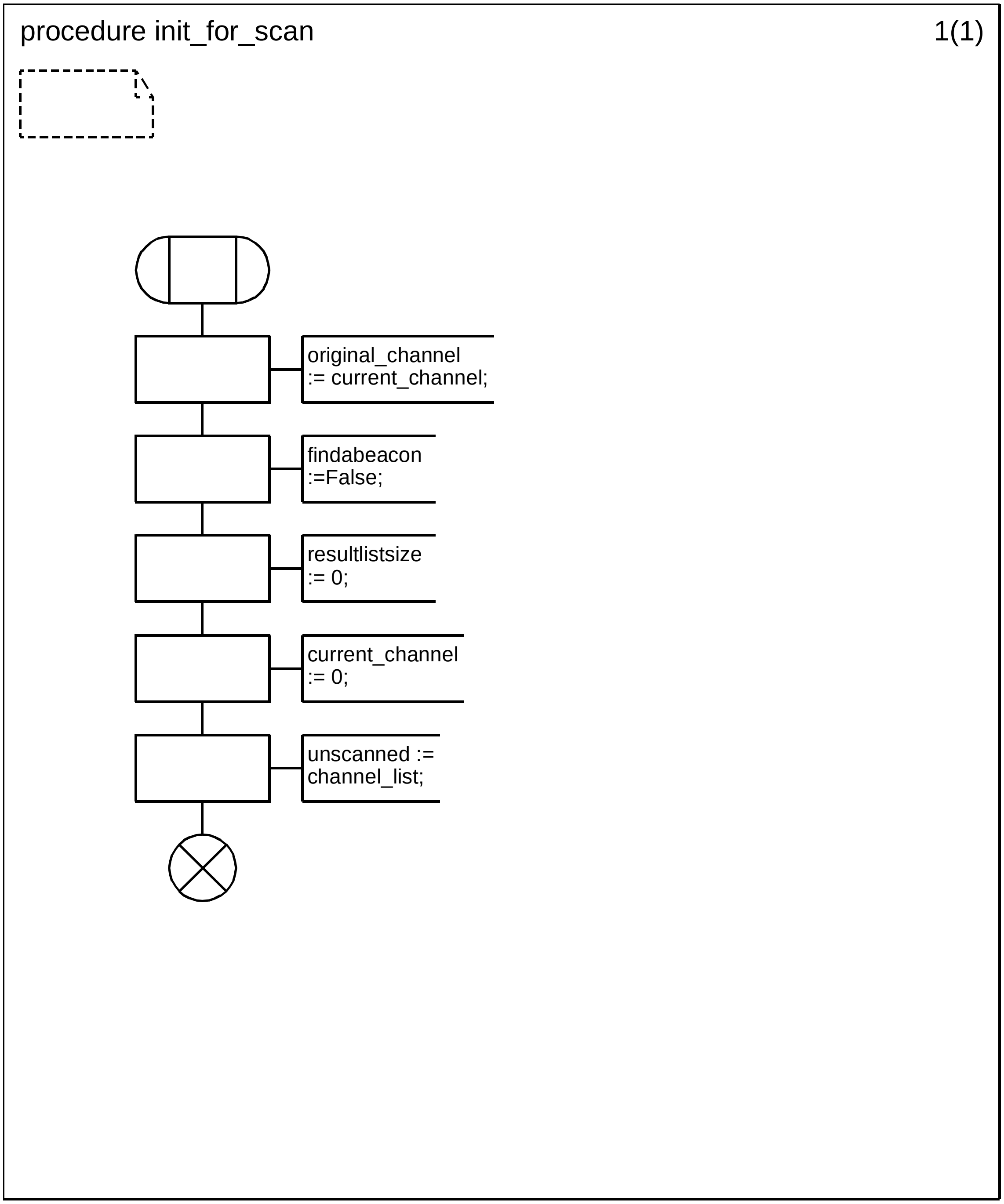
D.3.1.154.82.60 Procedure parse_acknowledgement

.



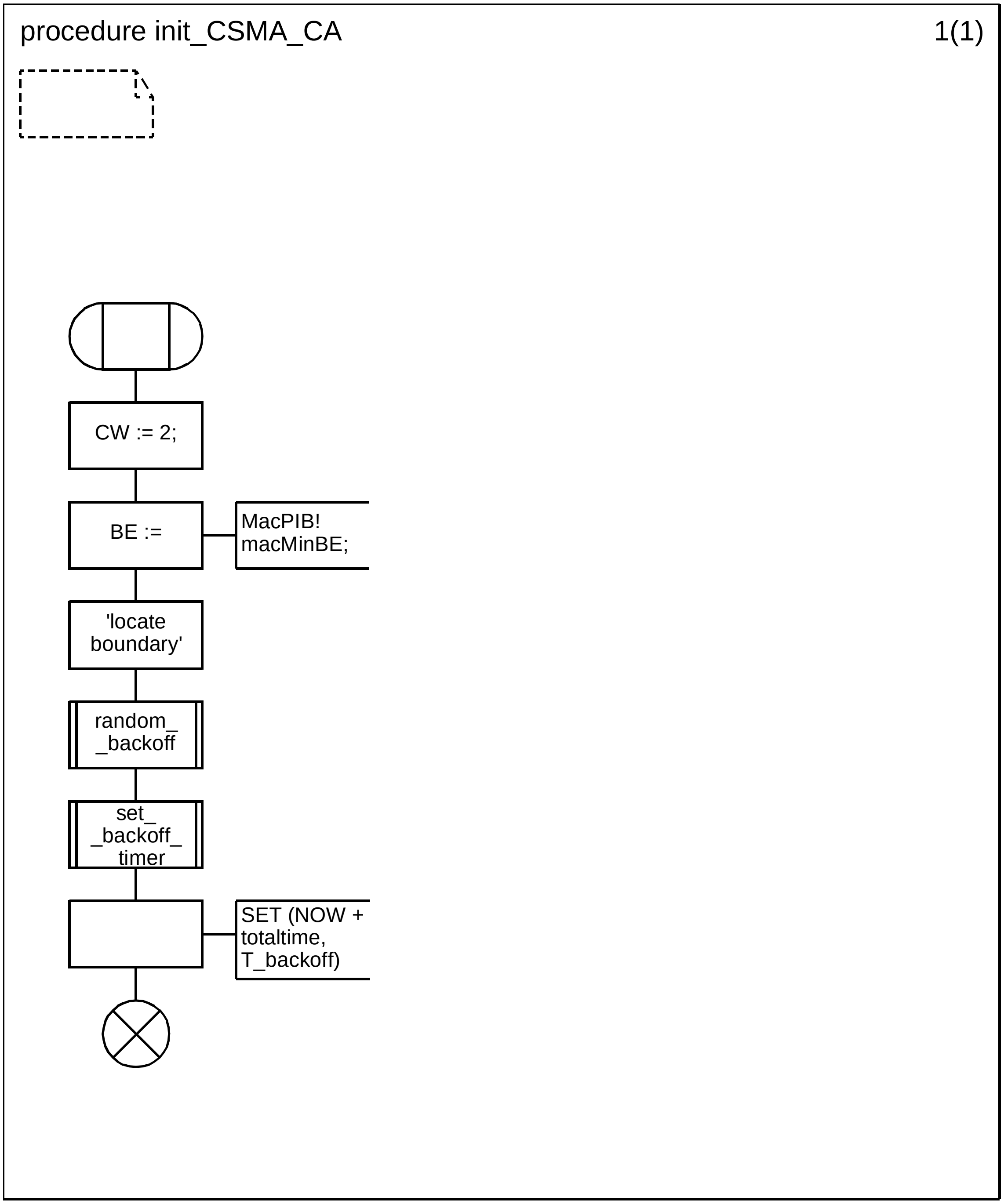
D.3.1.154.83 Procedure init_for_scan

.



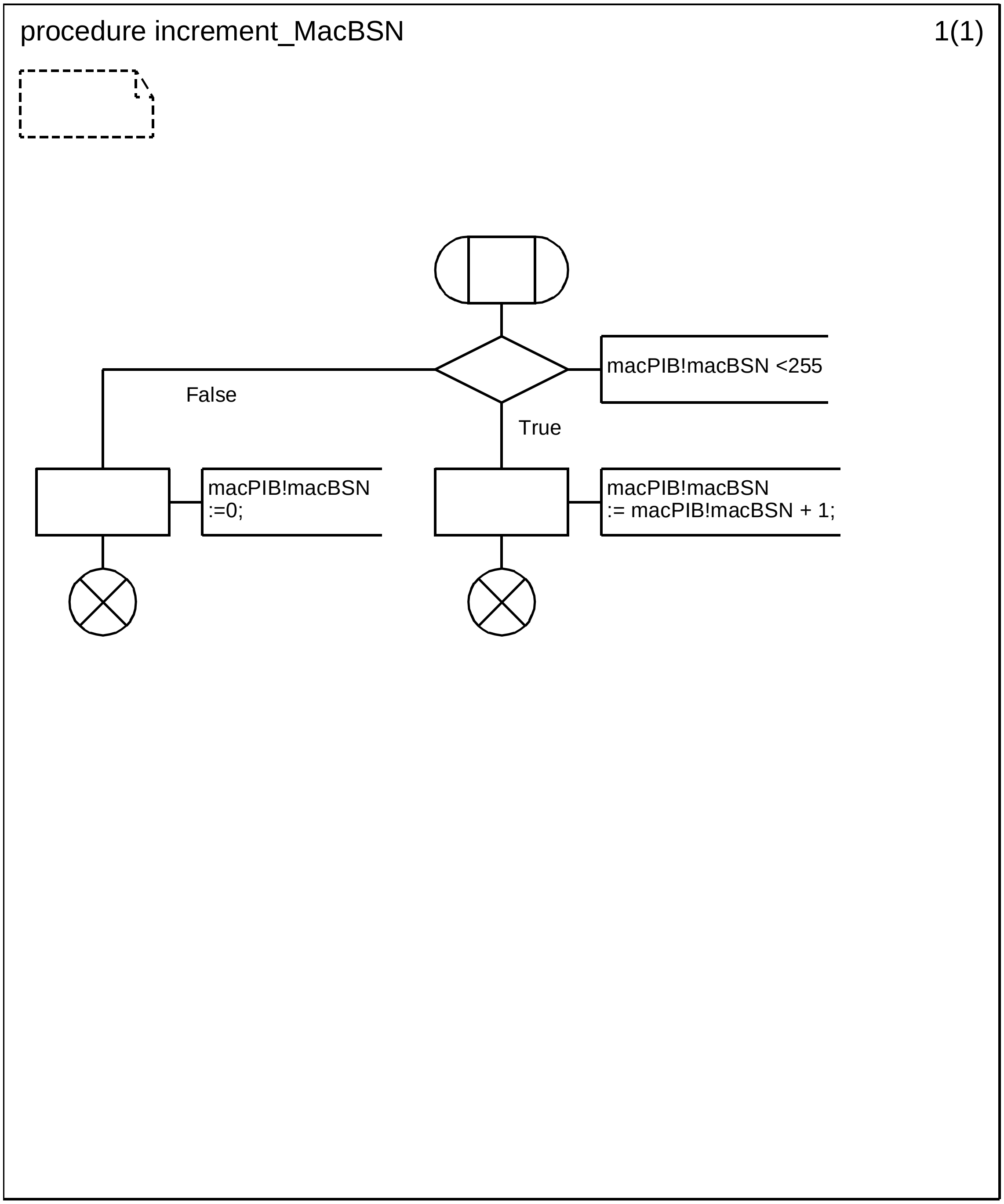
D.3.1.154.84 Procedure init_CSMA_CA

.



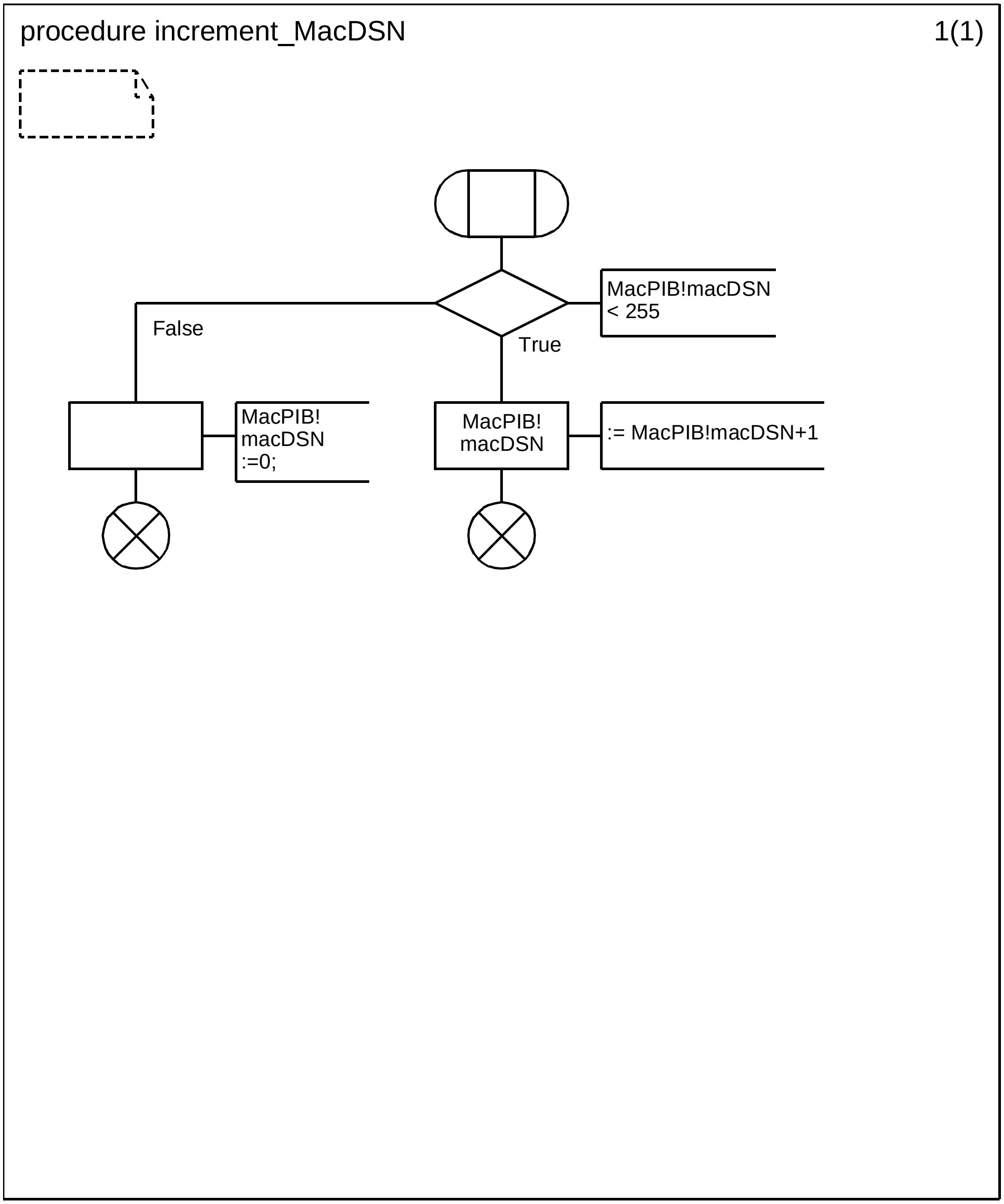
D.3.1.154.85 Procedure increment_MacBSN

.



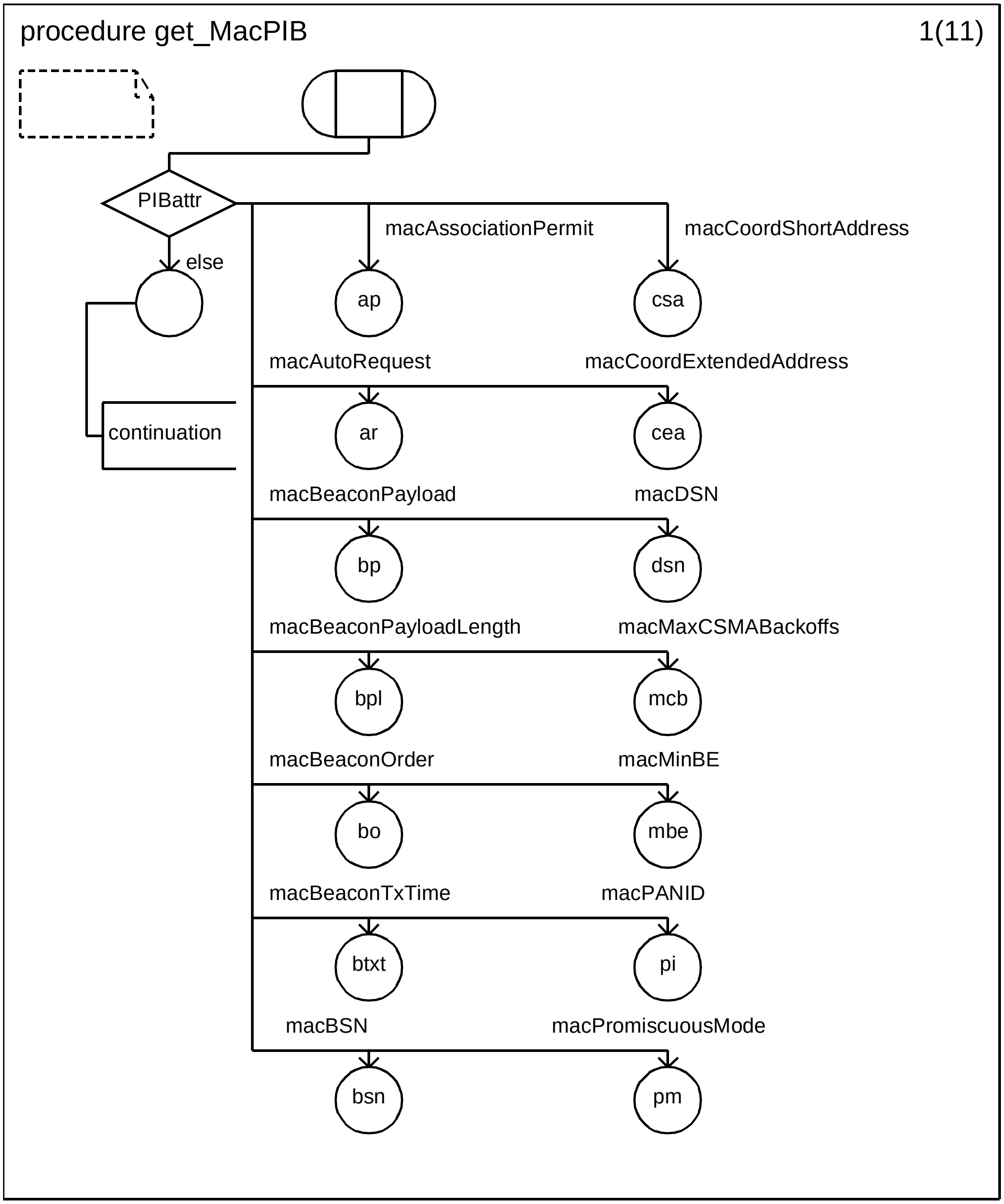
D.3.1.154.86 Procedure increment_MacDSN

.



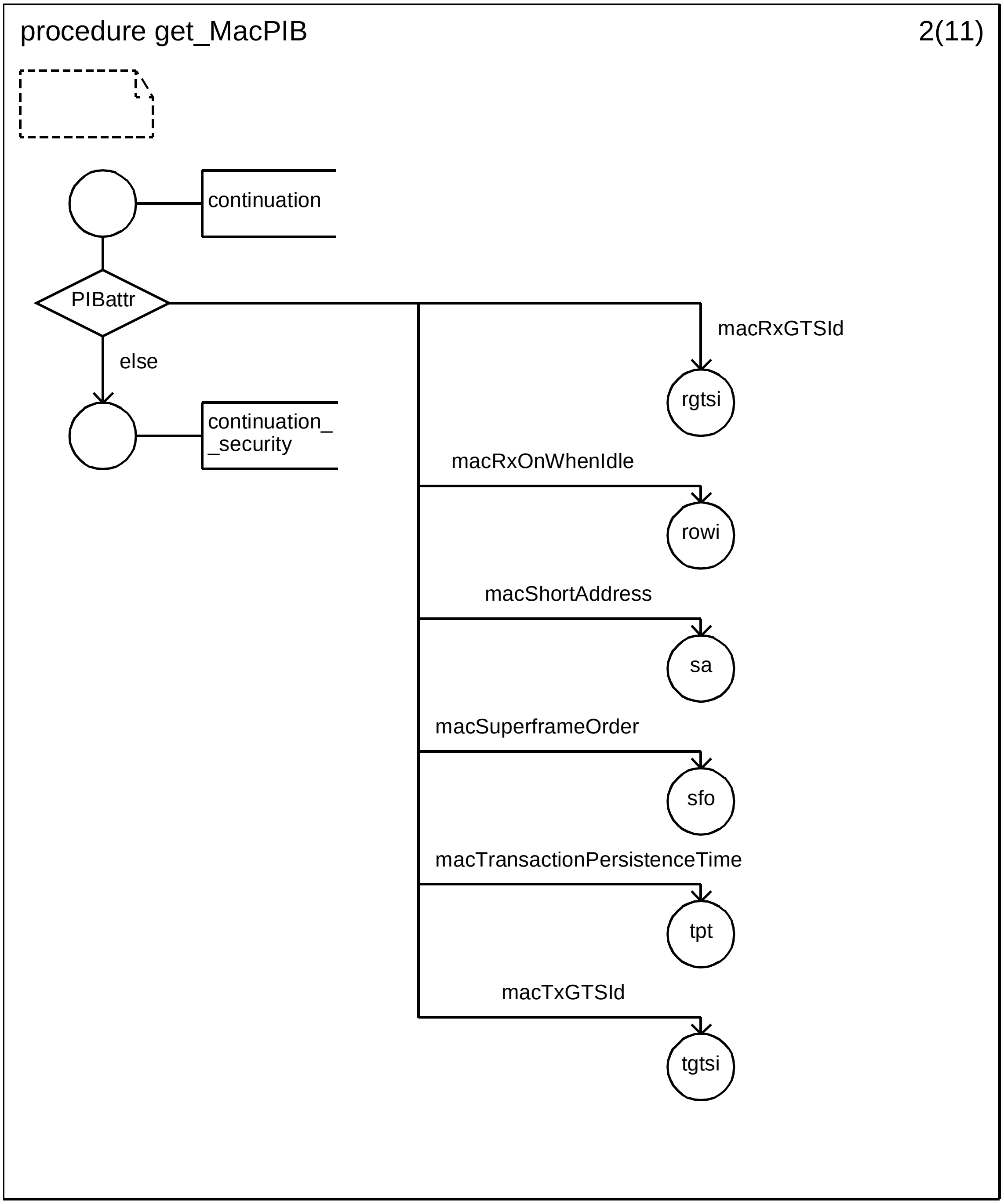
D.3.1.154.87 Procedure get_MACPIB (1)

.



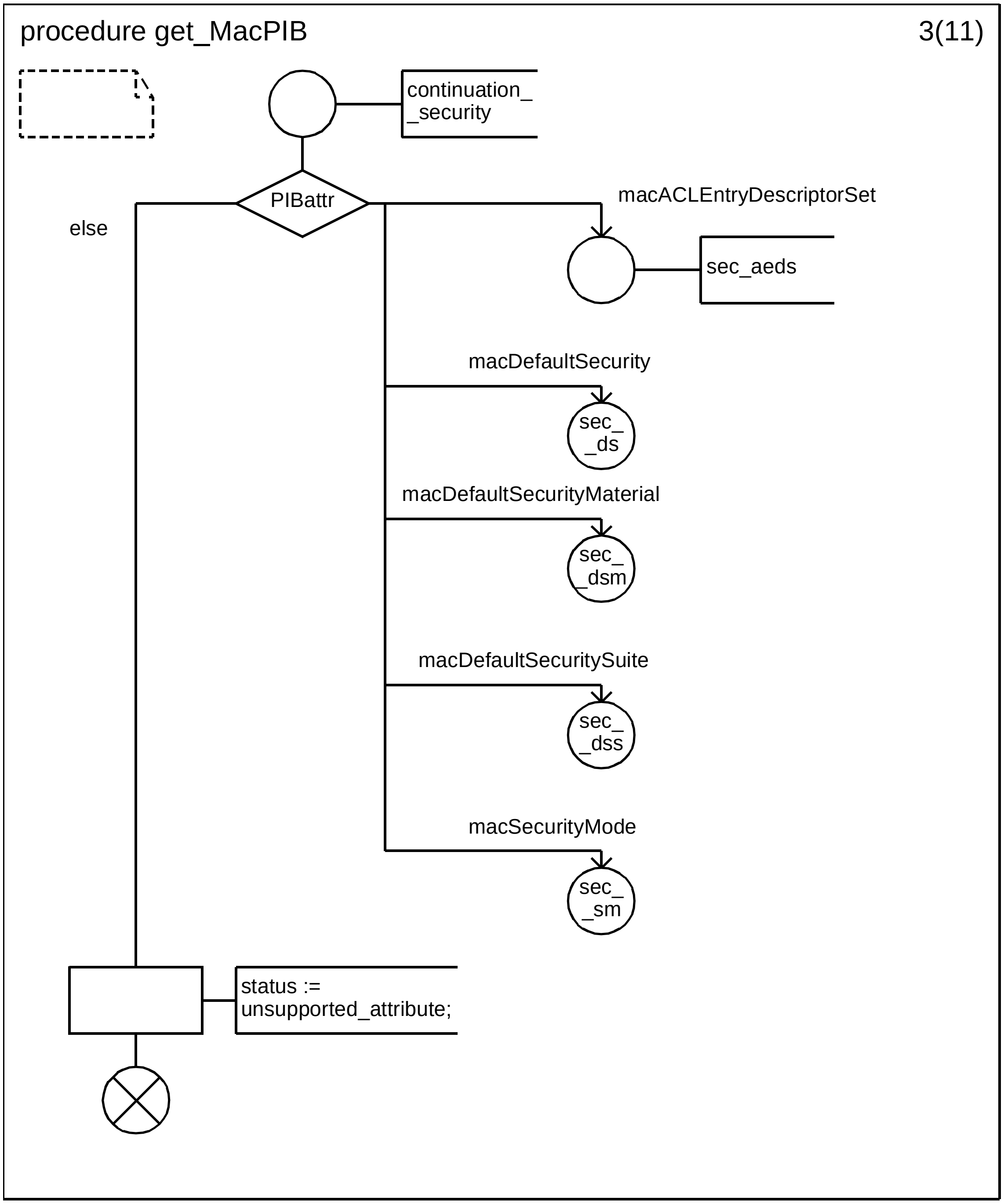
D.3.1.154.88 Procedure get_MACPIB (2)

.



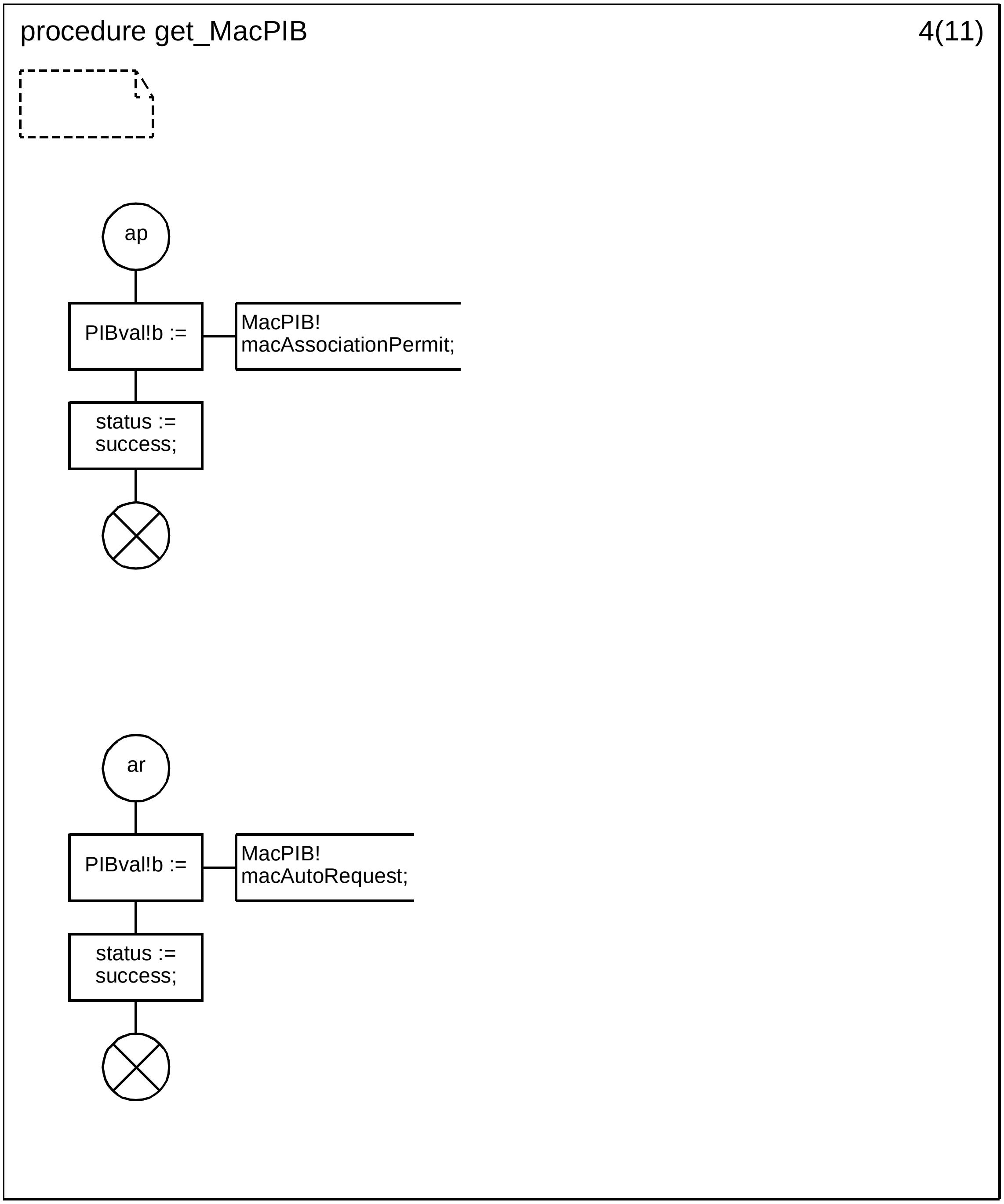
D.3.1.154.89 Procedure get_MACPIB (3)

.



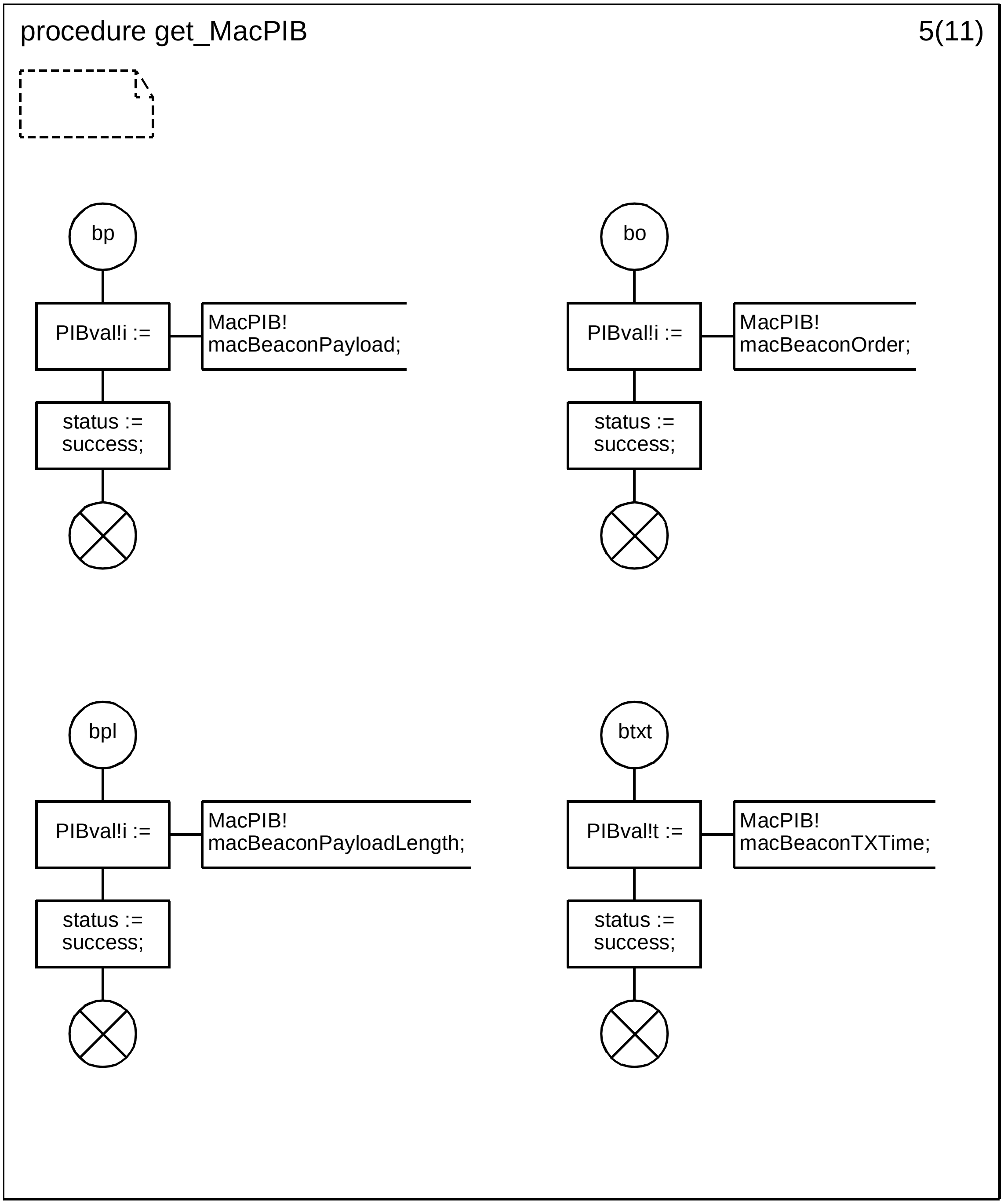
D.3.1.154.90 Procedure get_MACPIB (4)

.



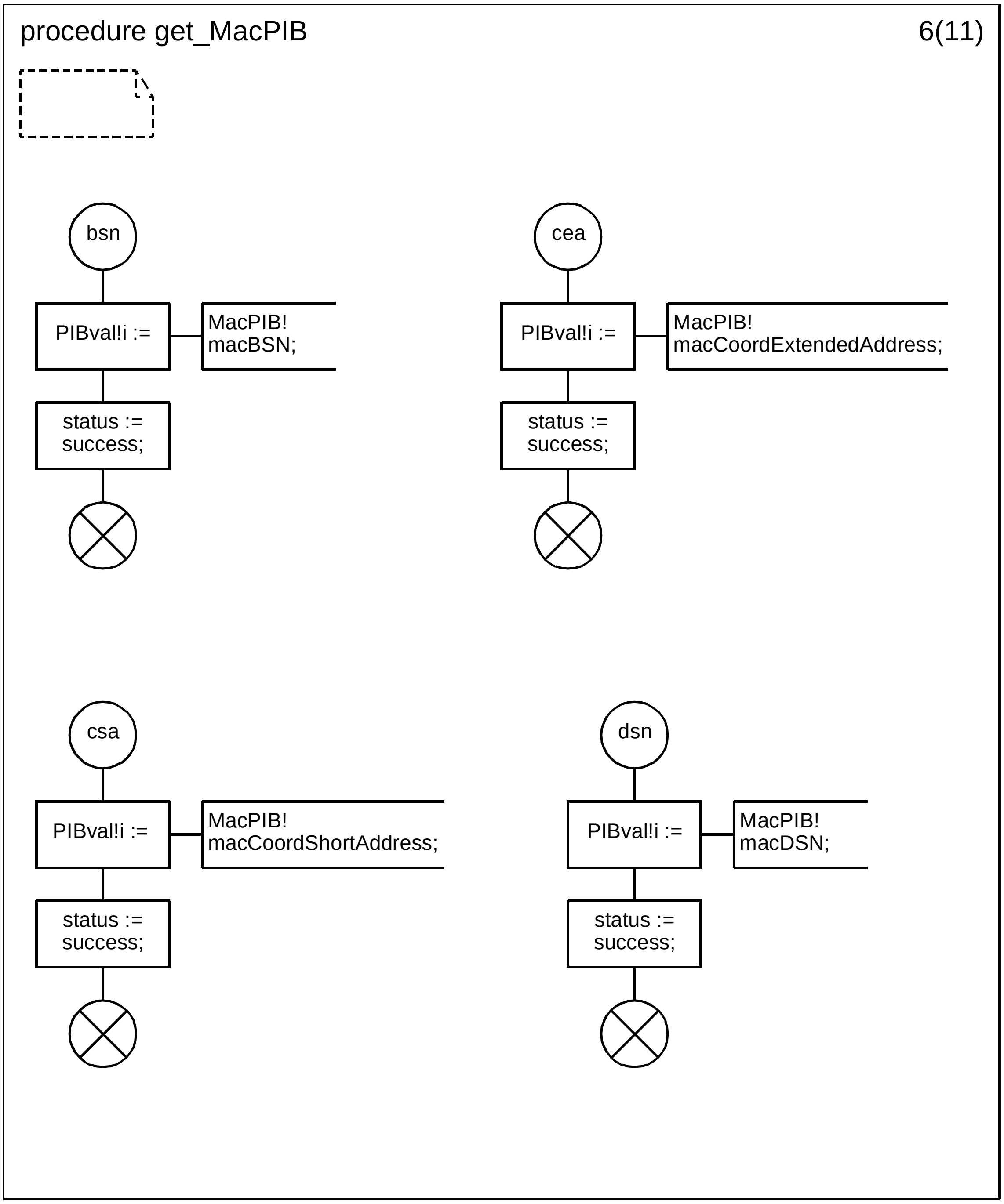
D.3.1.154.91 Procedure get_MACPIB (5)

.



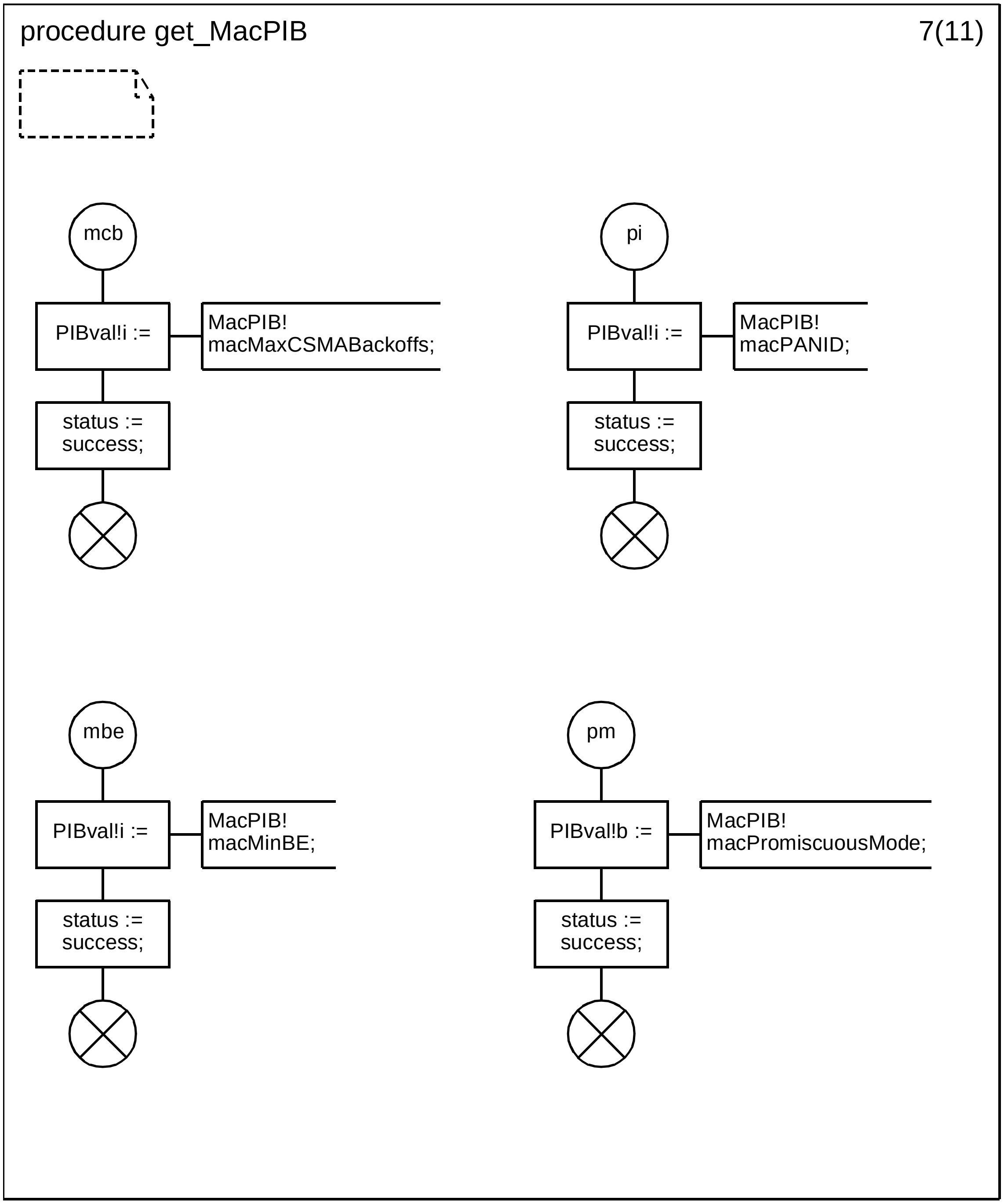
D.3.1.154.92 Procedure get_MACPIB (6)

.



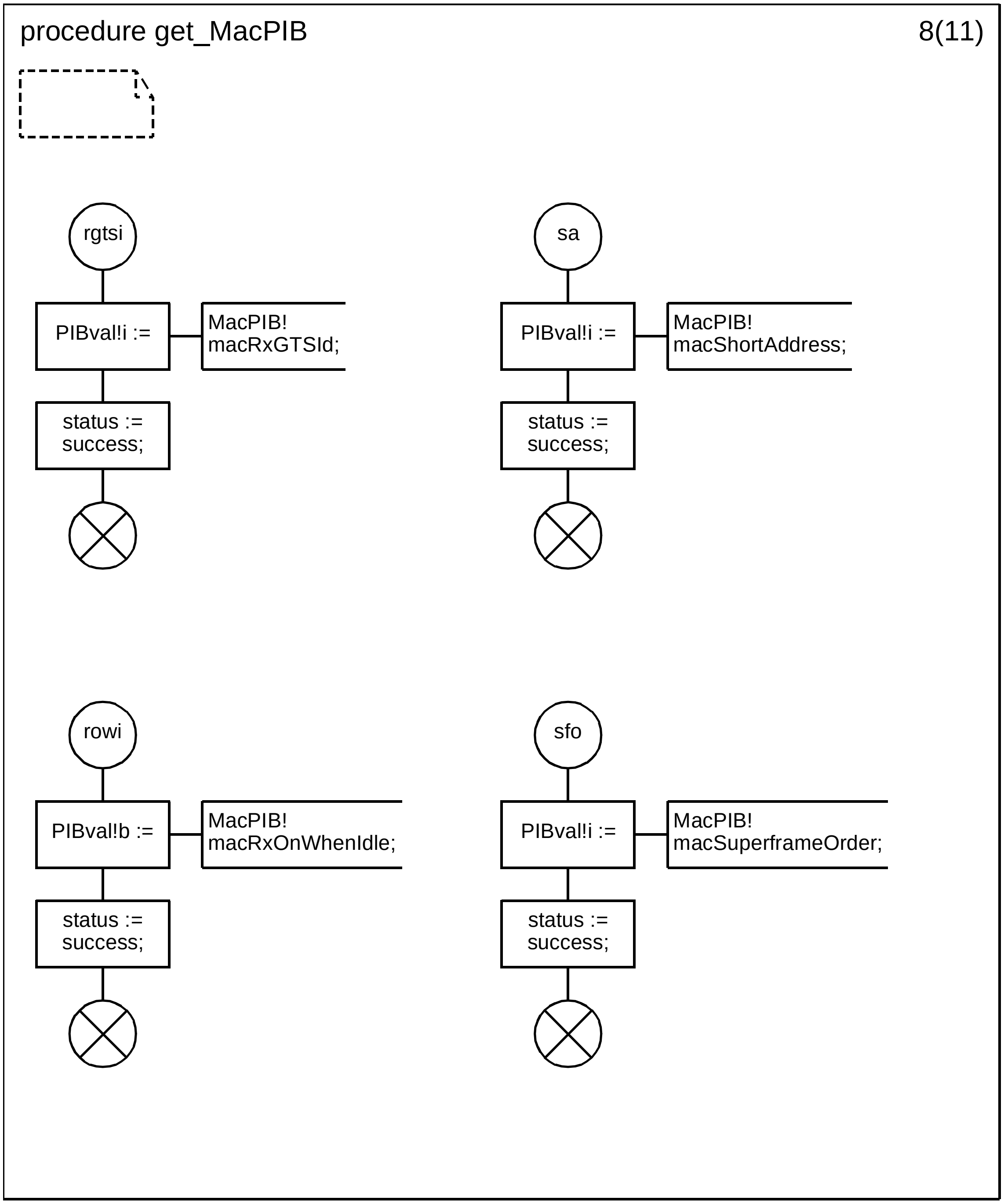
D.3.1.154.93 Procedure get_MACPIB (7)

.



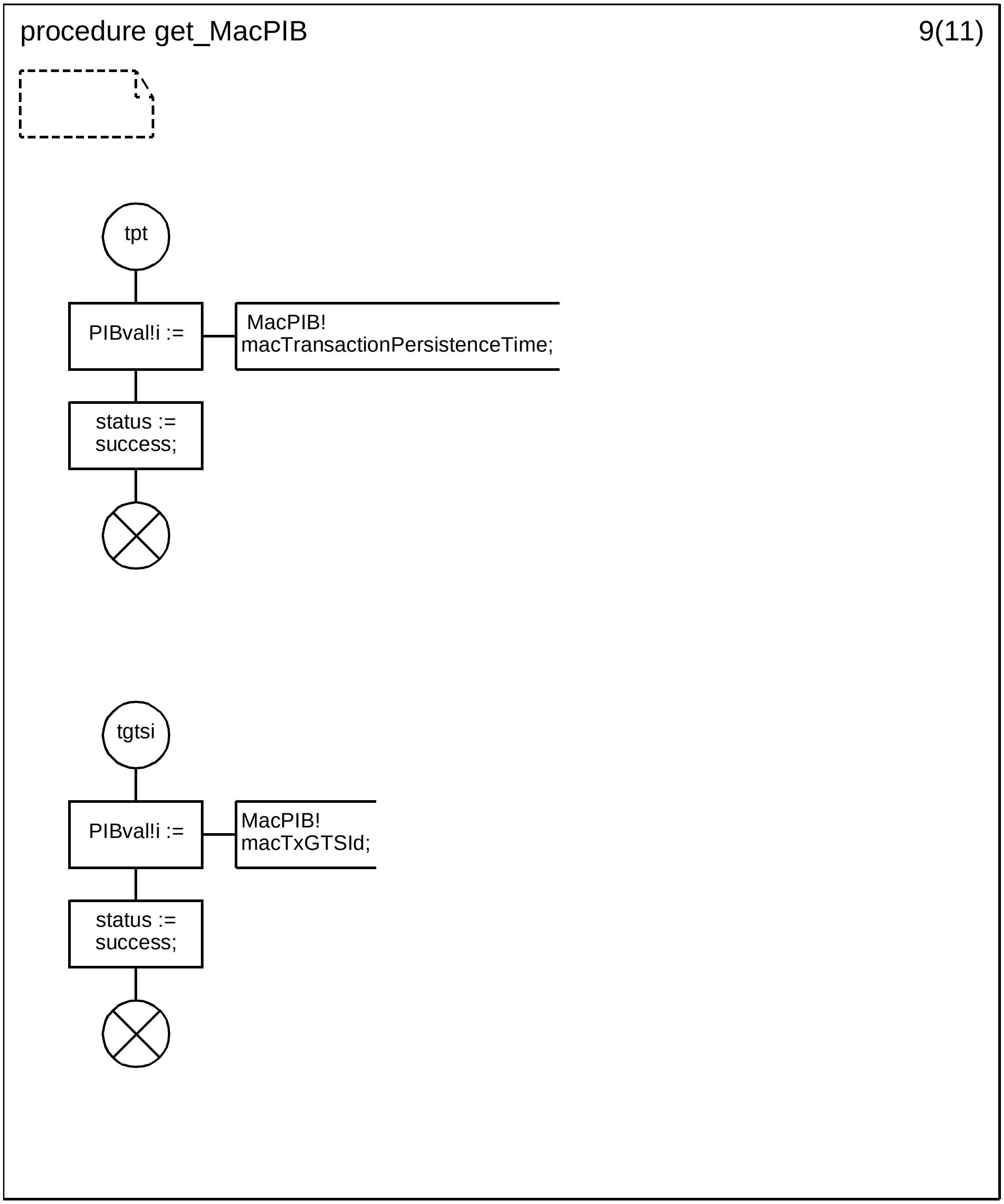
D.3.1.154.94 Procedure get_MACPIB (8)

.



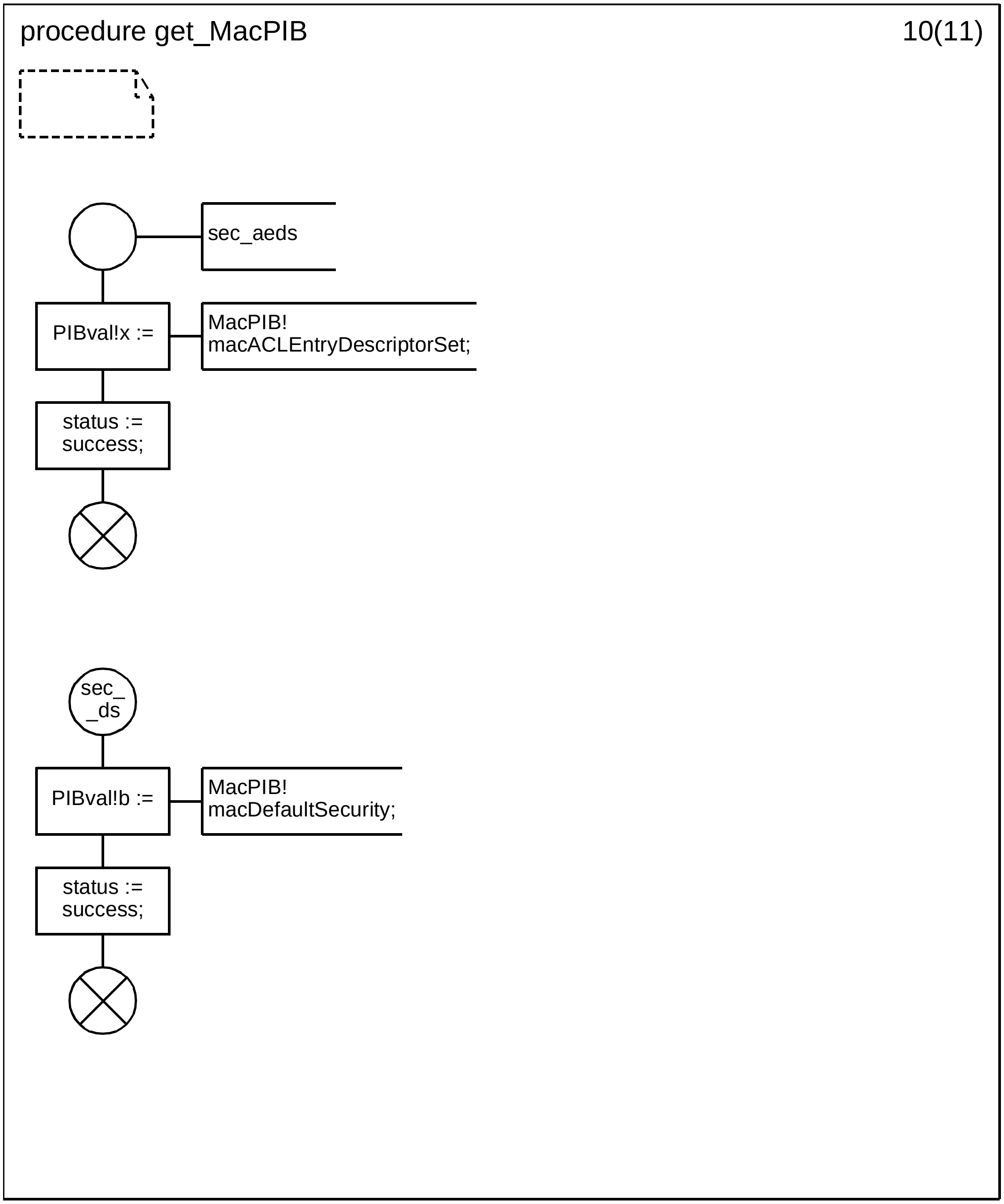
D.3.1.154.95 Procedure get_MACPIB (9)

.



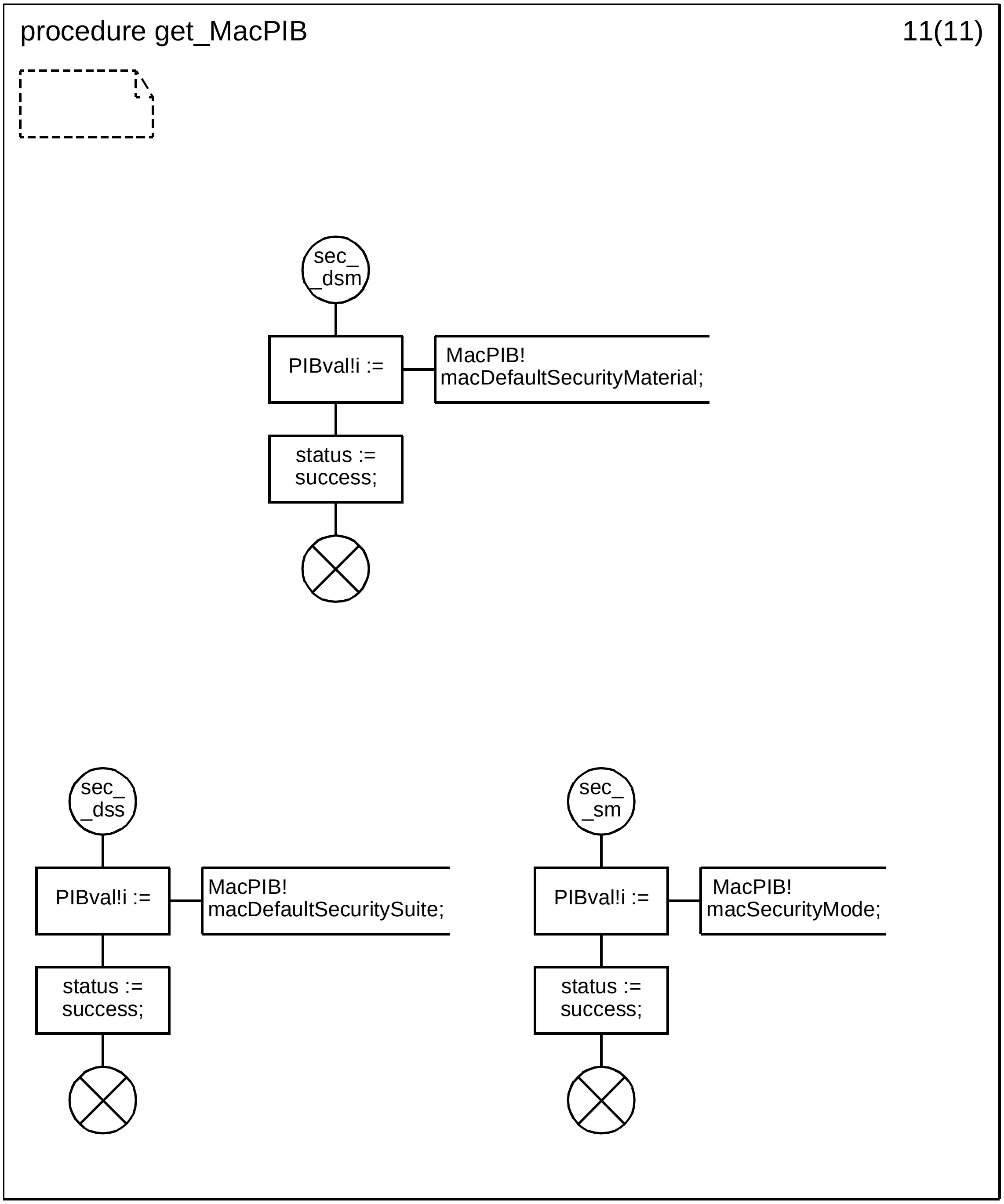
D.3.1.154.96 Procedure get_MACPIB (10)

.



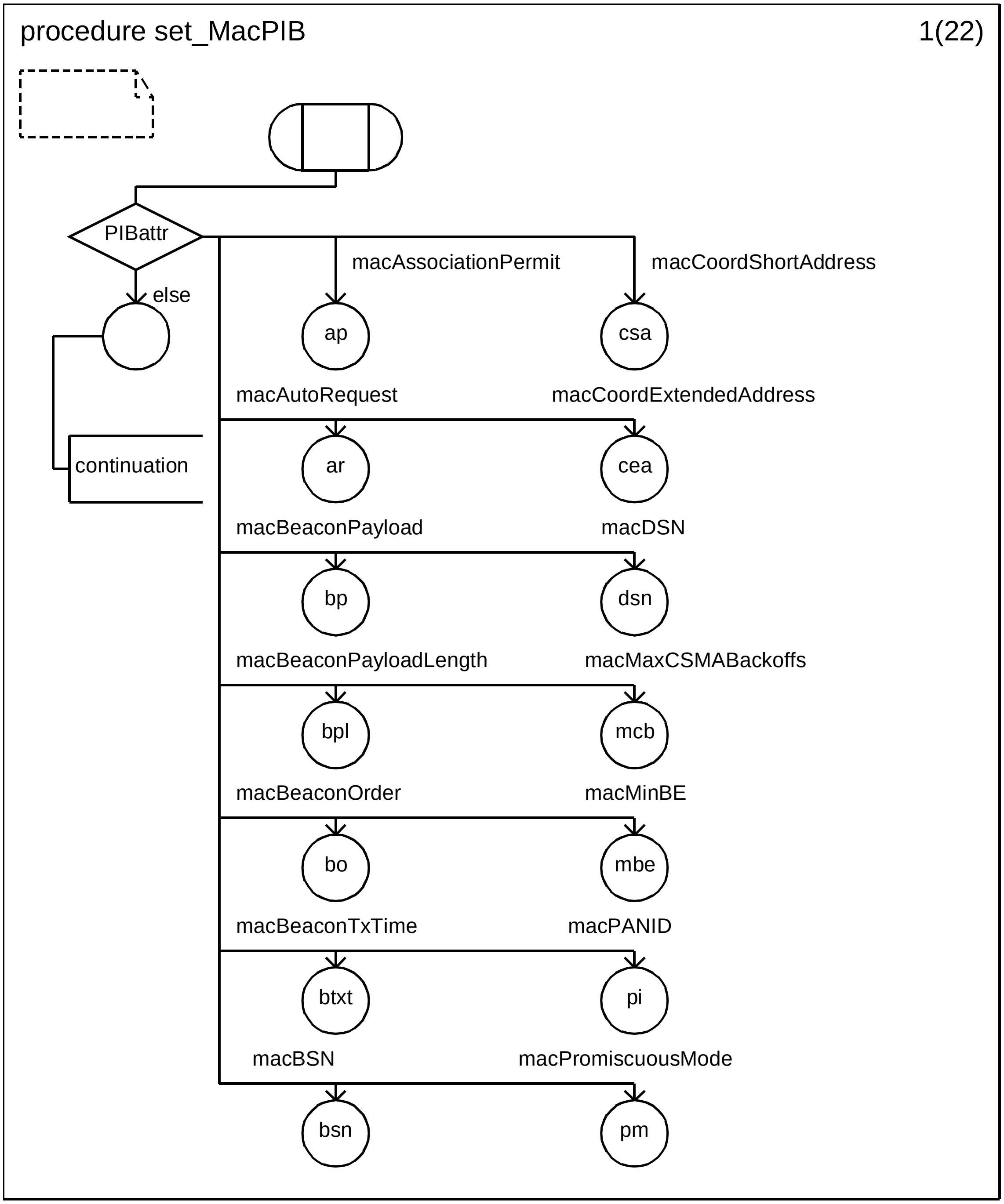
D.3.1.154.97 Procedure get_MACPIB (11)

.



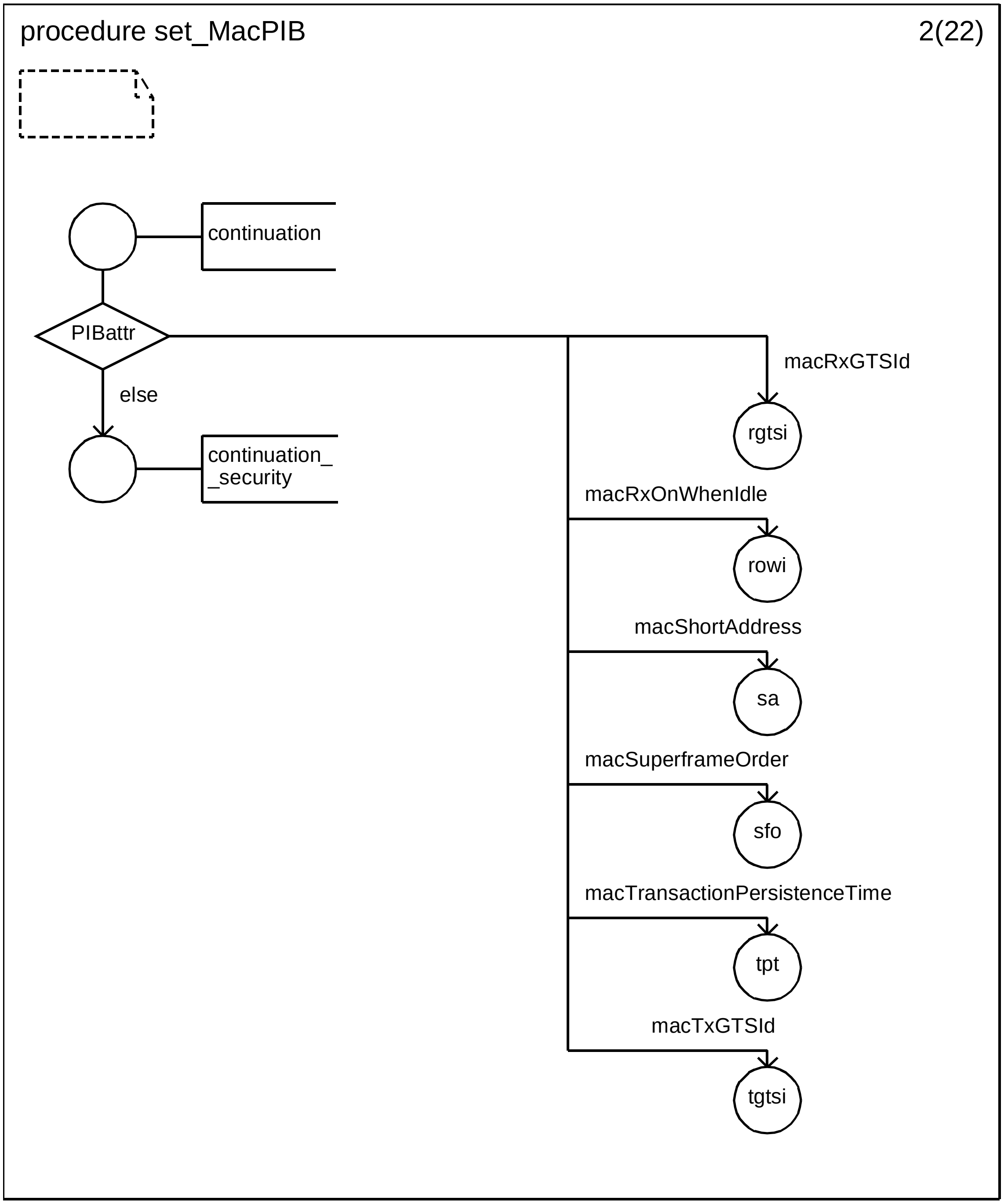
D.3.1.154.98 Procedure set_MacPIB (1)

.



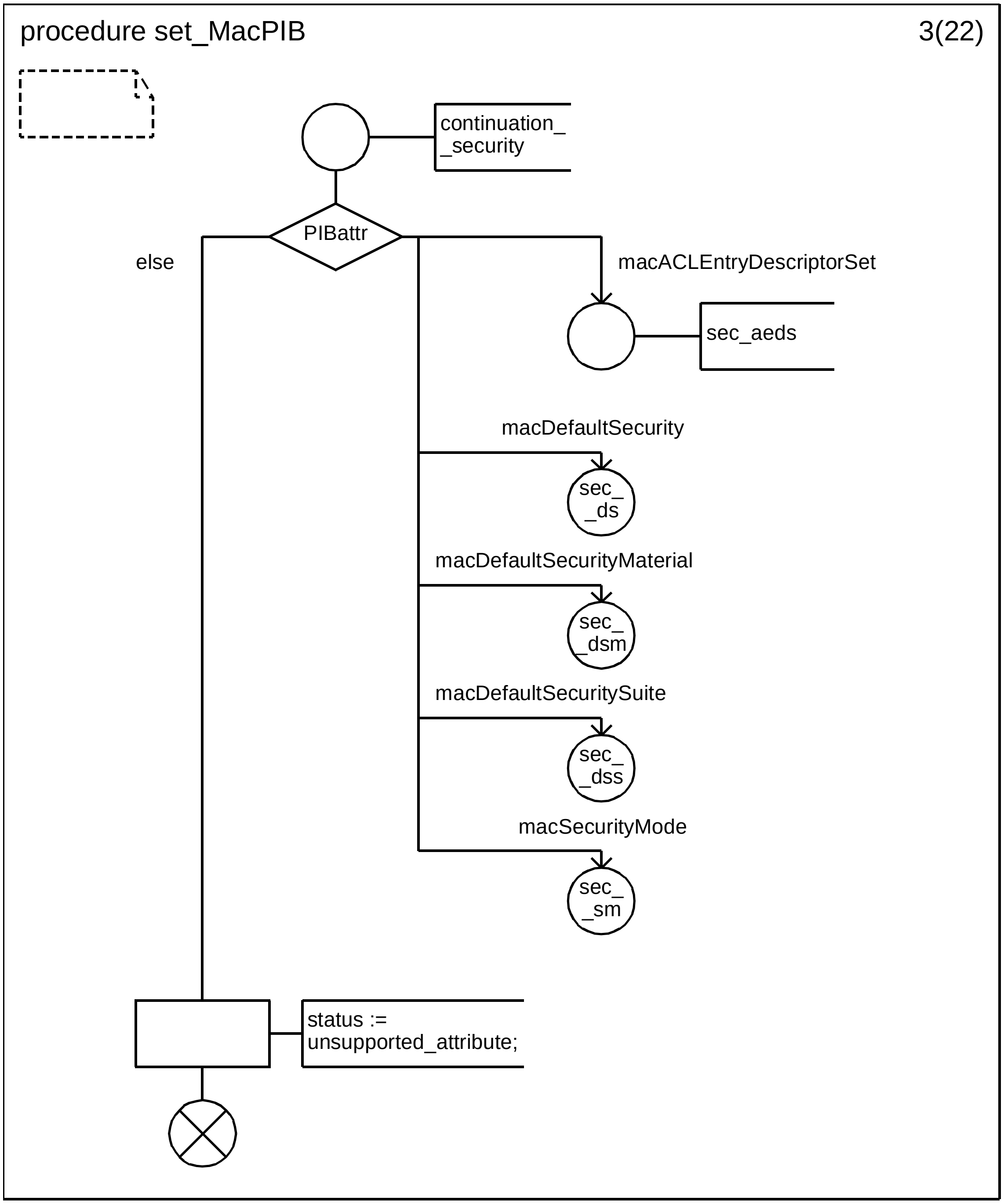
D.3.1.154.99 Procedure set_MacPIB (2)

.



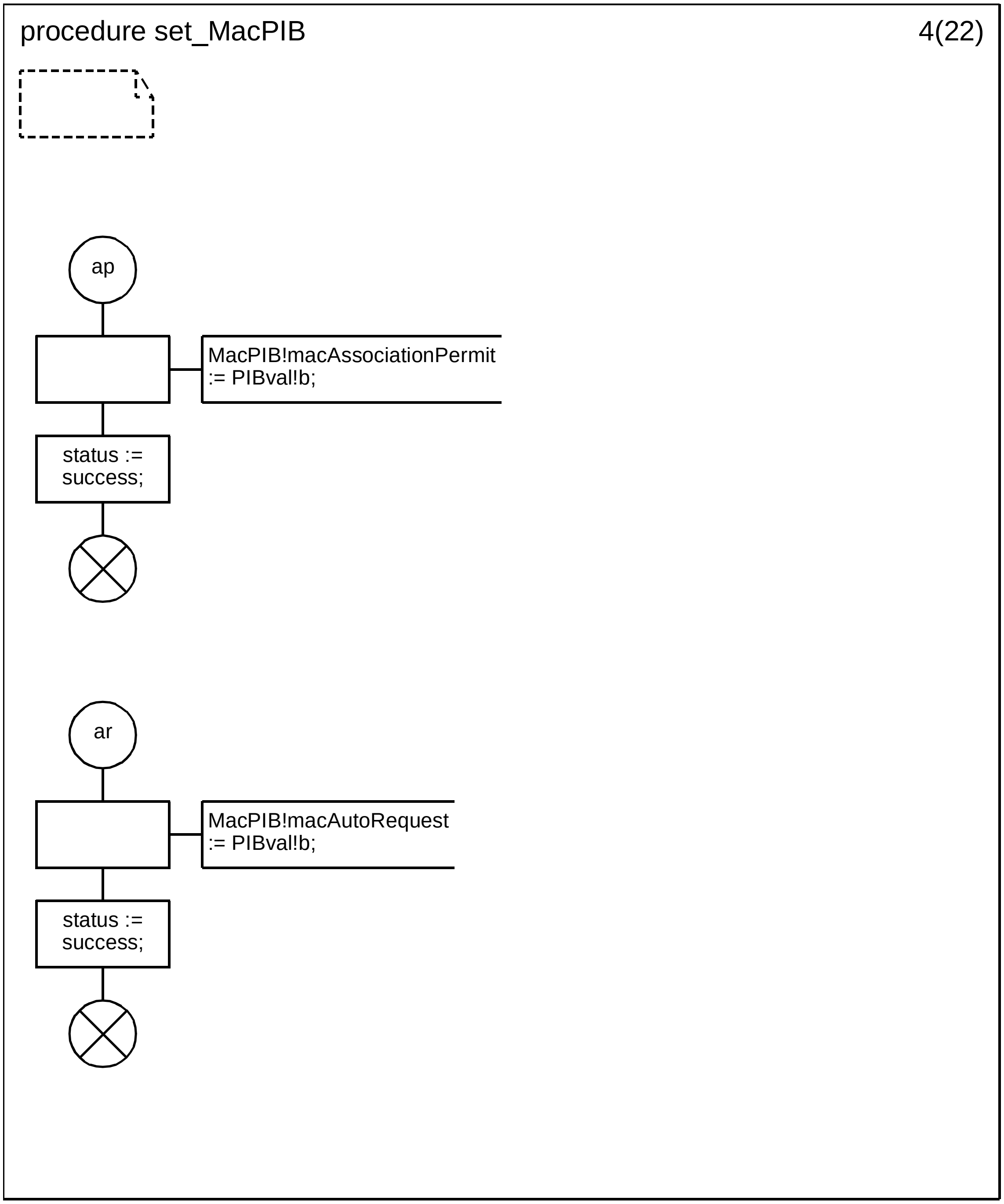
D.3.1.154.100 Procedure set_MacPIB (3)

.



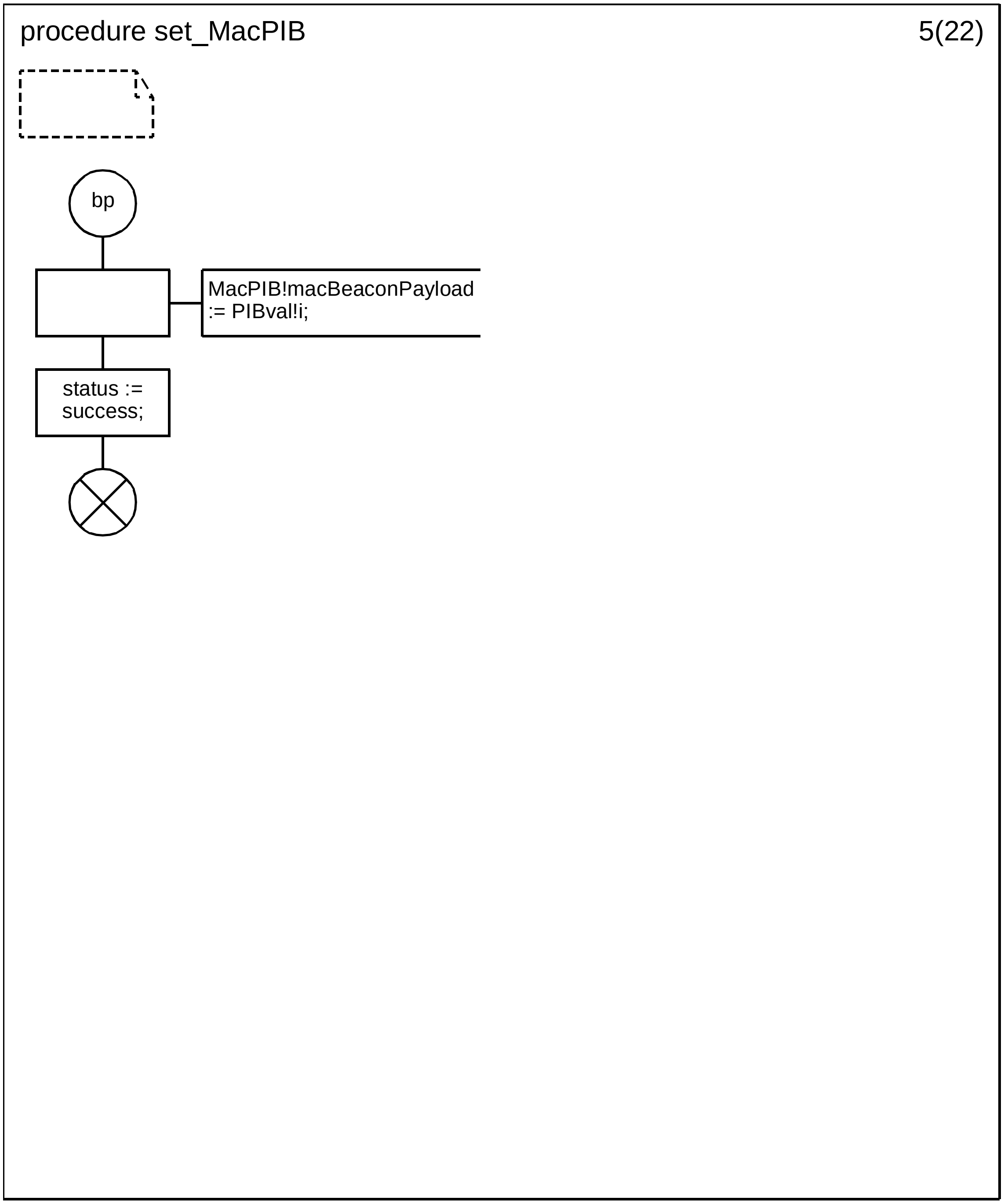
D.3.1.154.101 Procedure set_MacPIB (4)

.



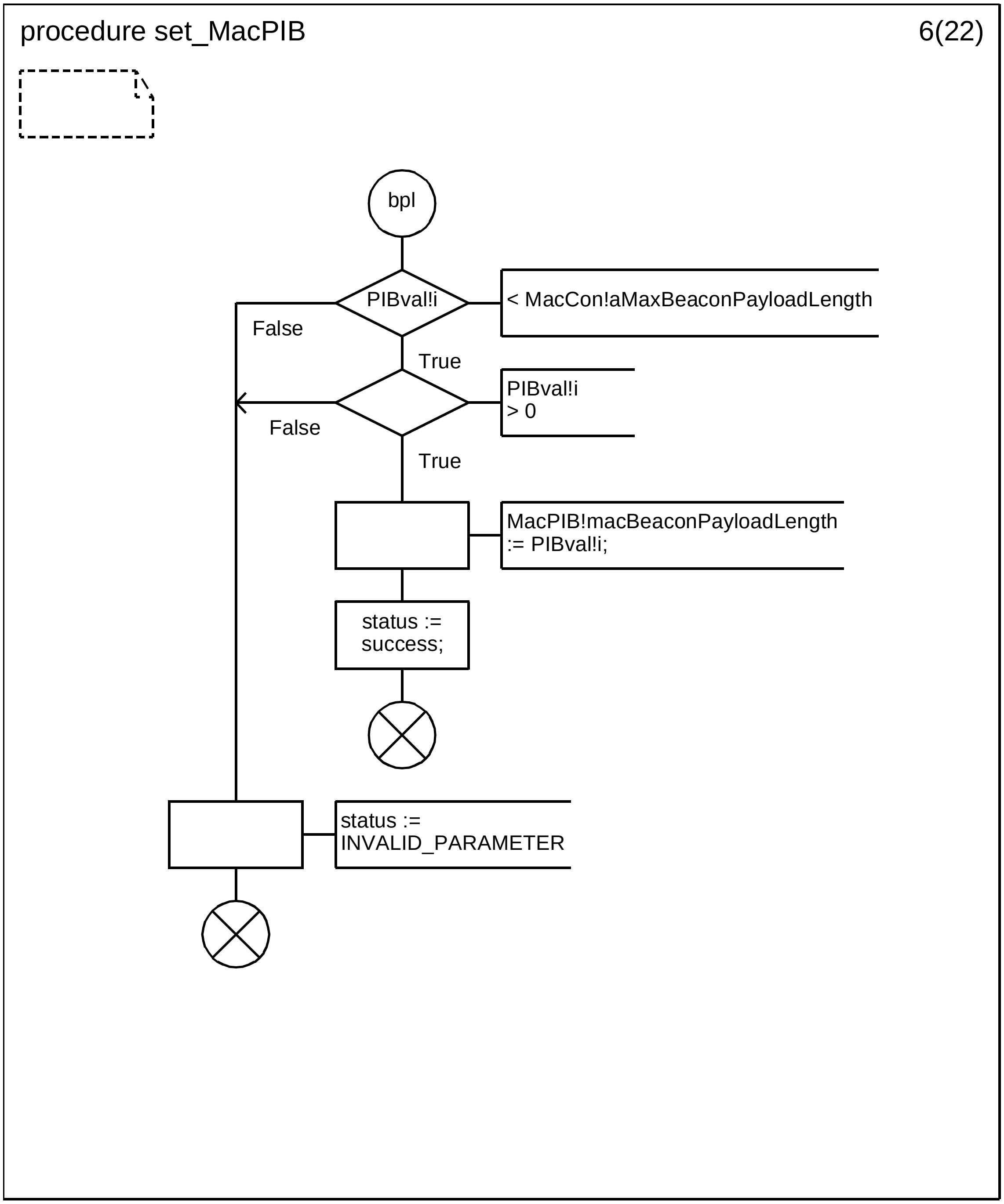
D.3.1.154.102 Procedure set_MacPIB (5)

.



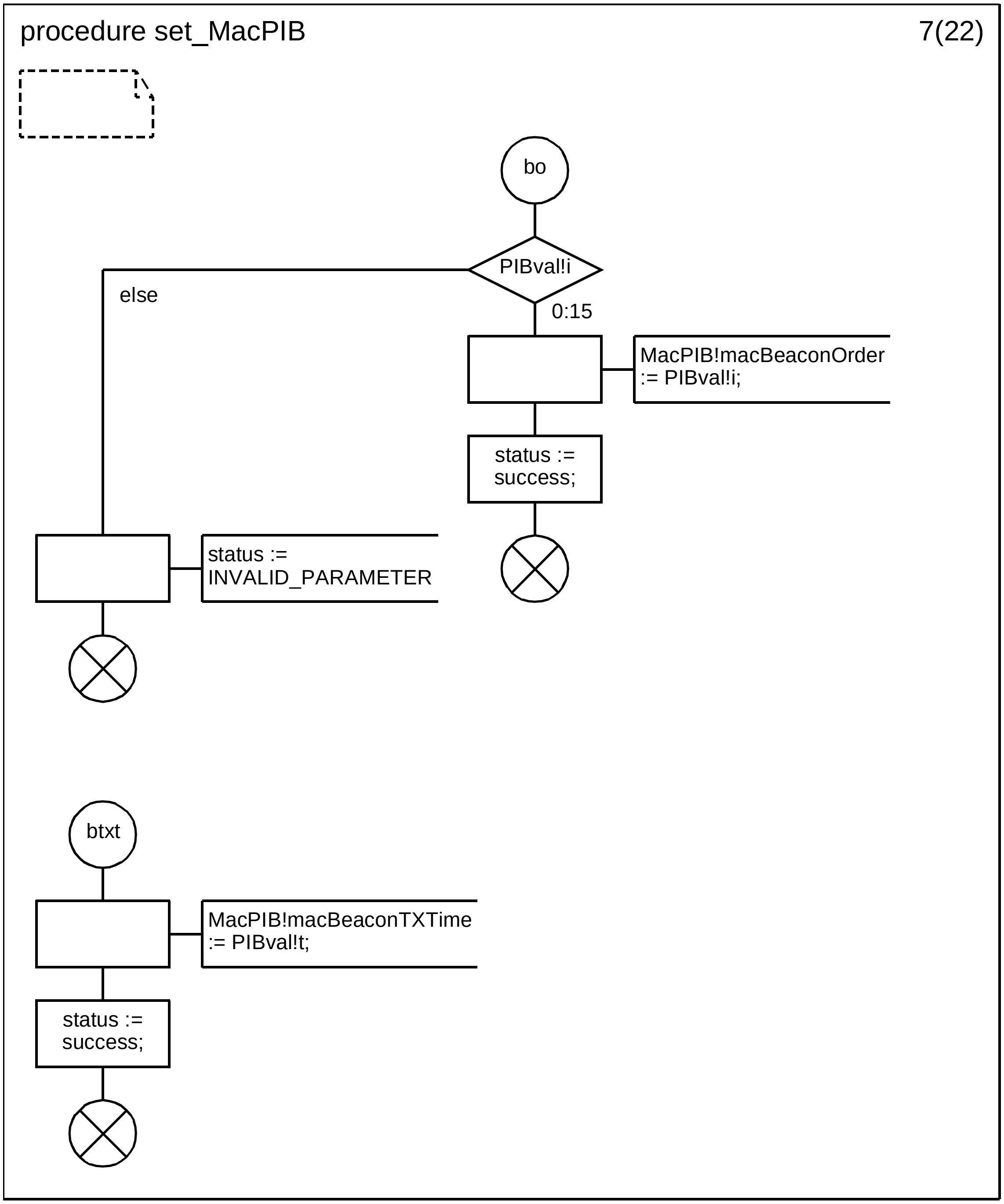
D.3.1.154.103 Procedure set_MacPIB (6)

.



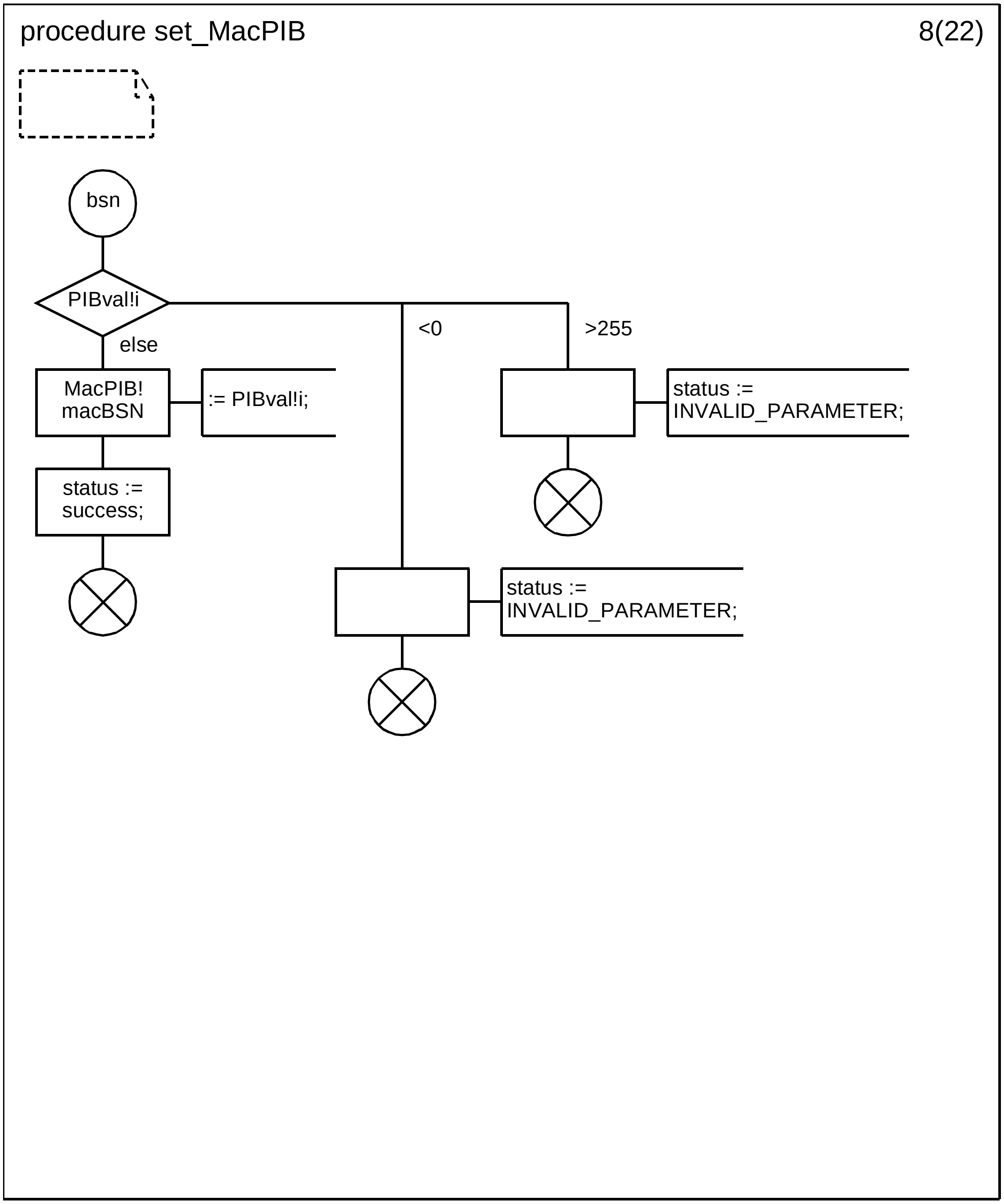
D.3.1.154.104 Procedure set_MacPIB (7)

.



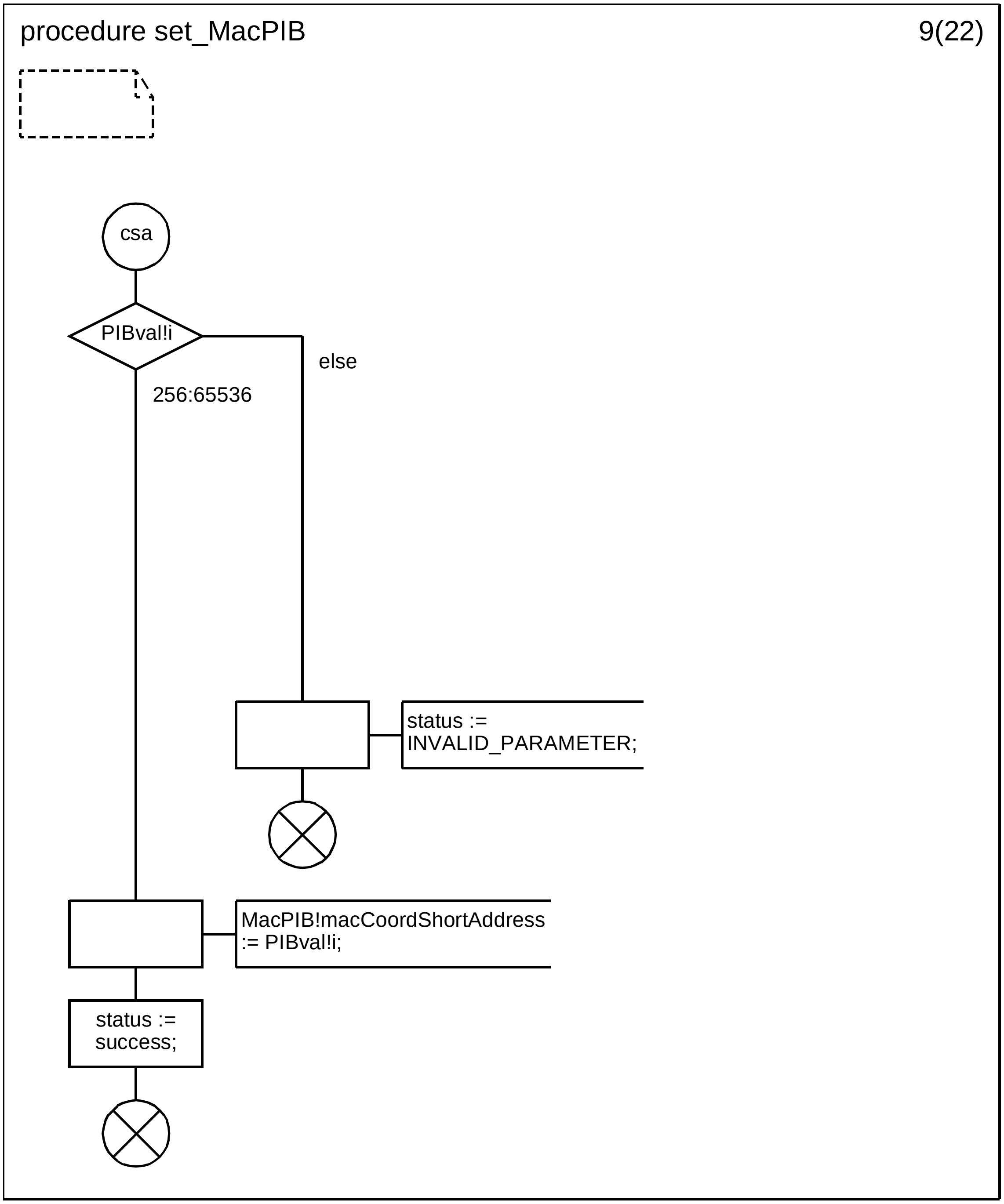
D.3.1.154.105 Procedure set_MacPIB (8)

.



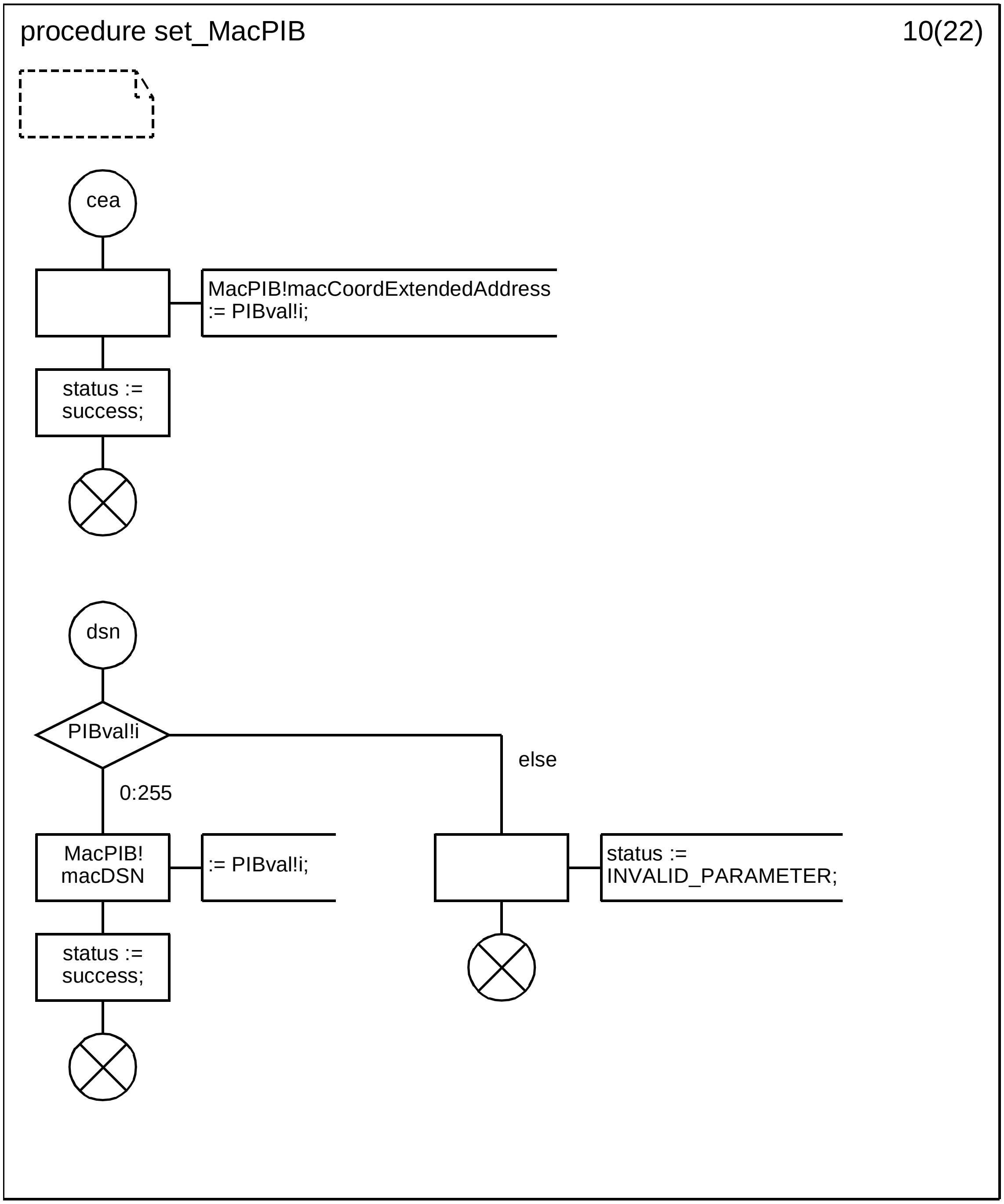
D.3.1.154.106 Procedure set_MacPIB (9)

.



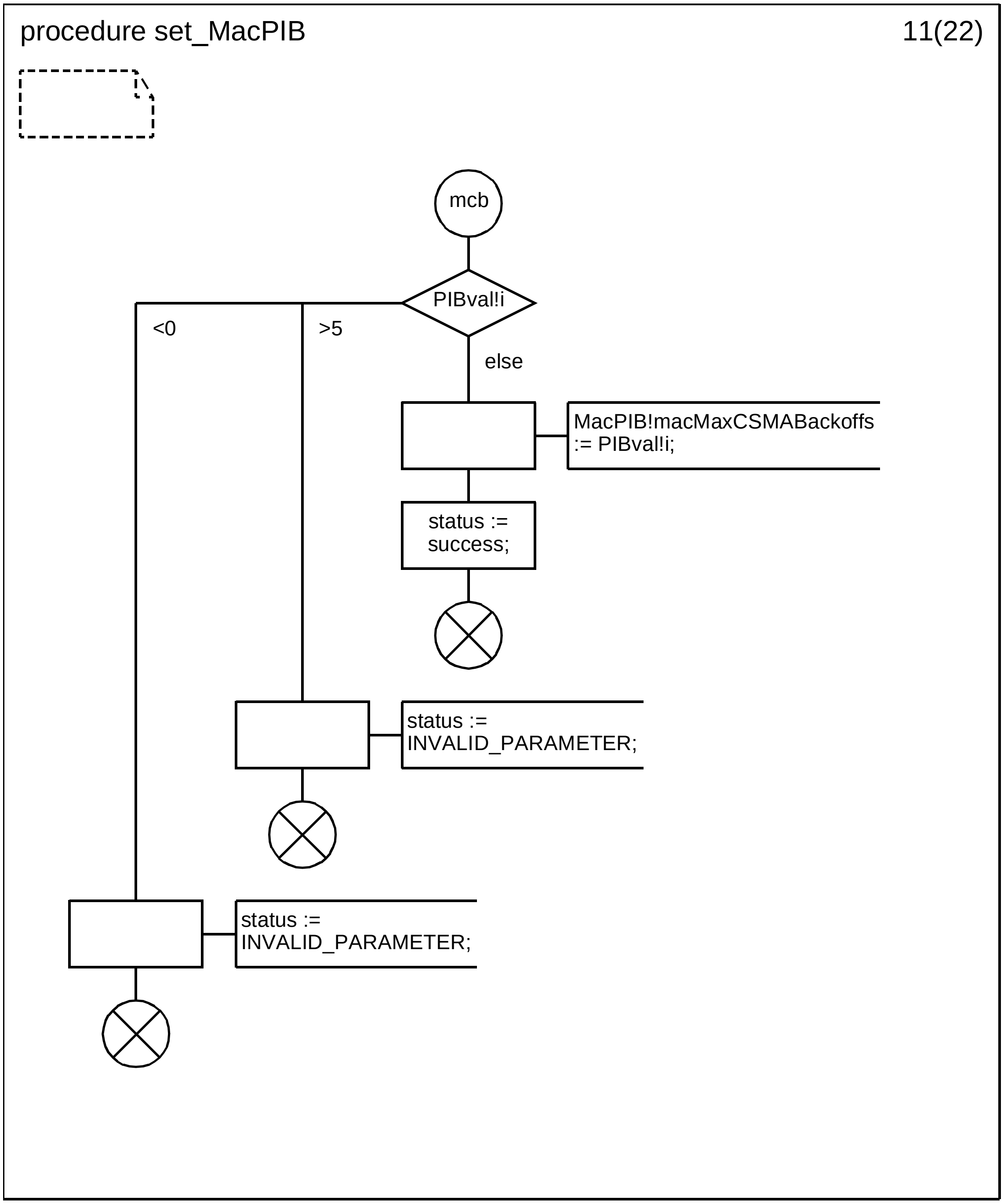
D.3.1.154.107 Procedure set_MacPIB (10)

.



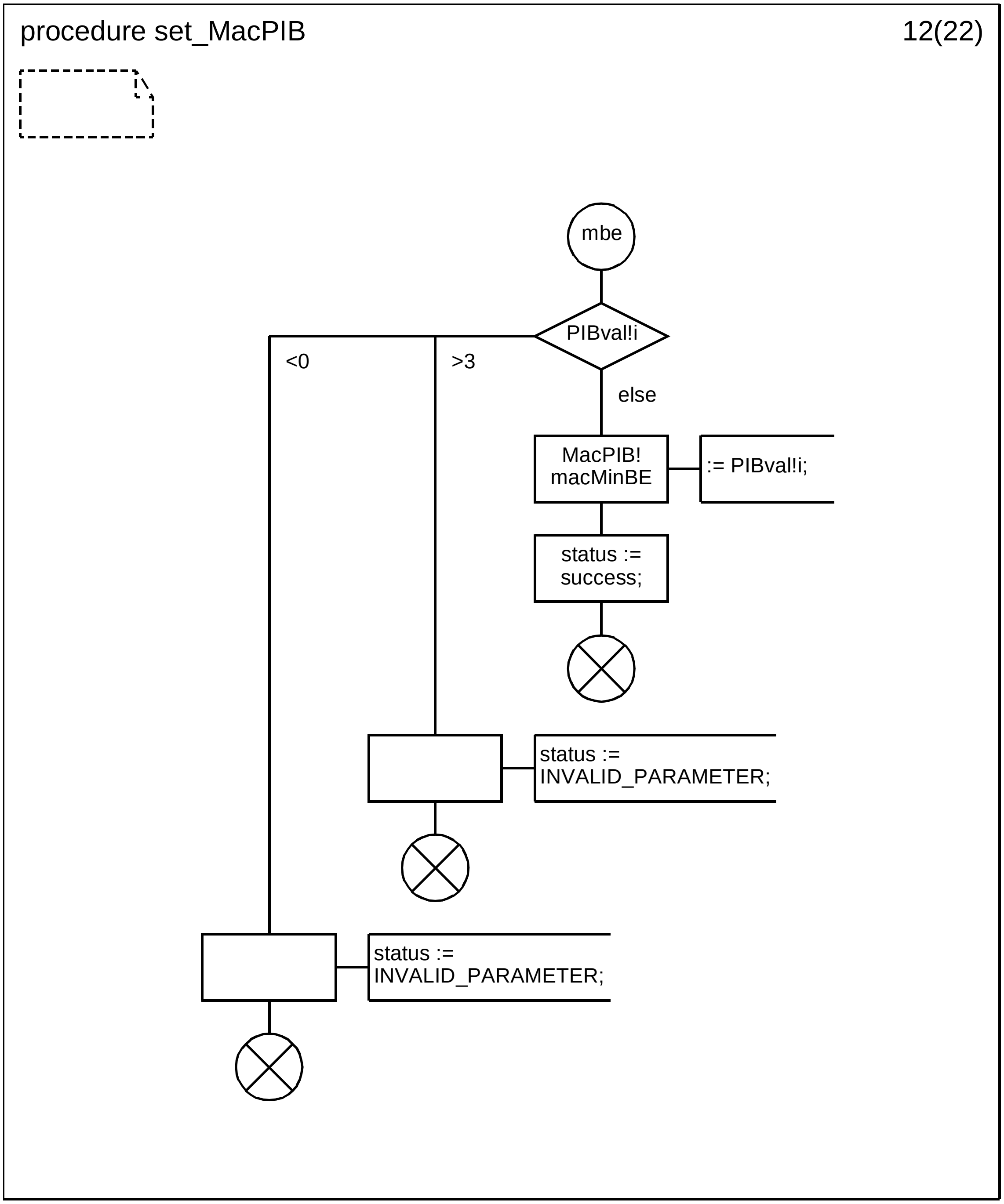
D.3.1.154.108 Procedure set_MacPIB (11)

.



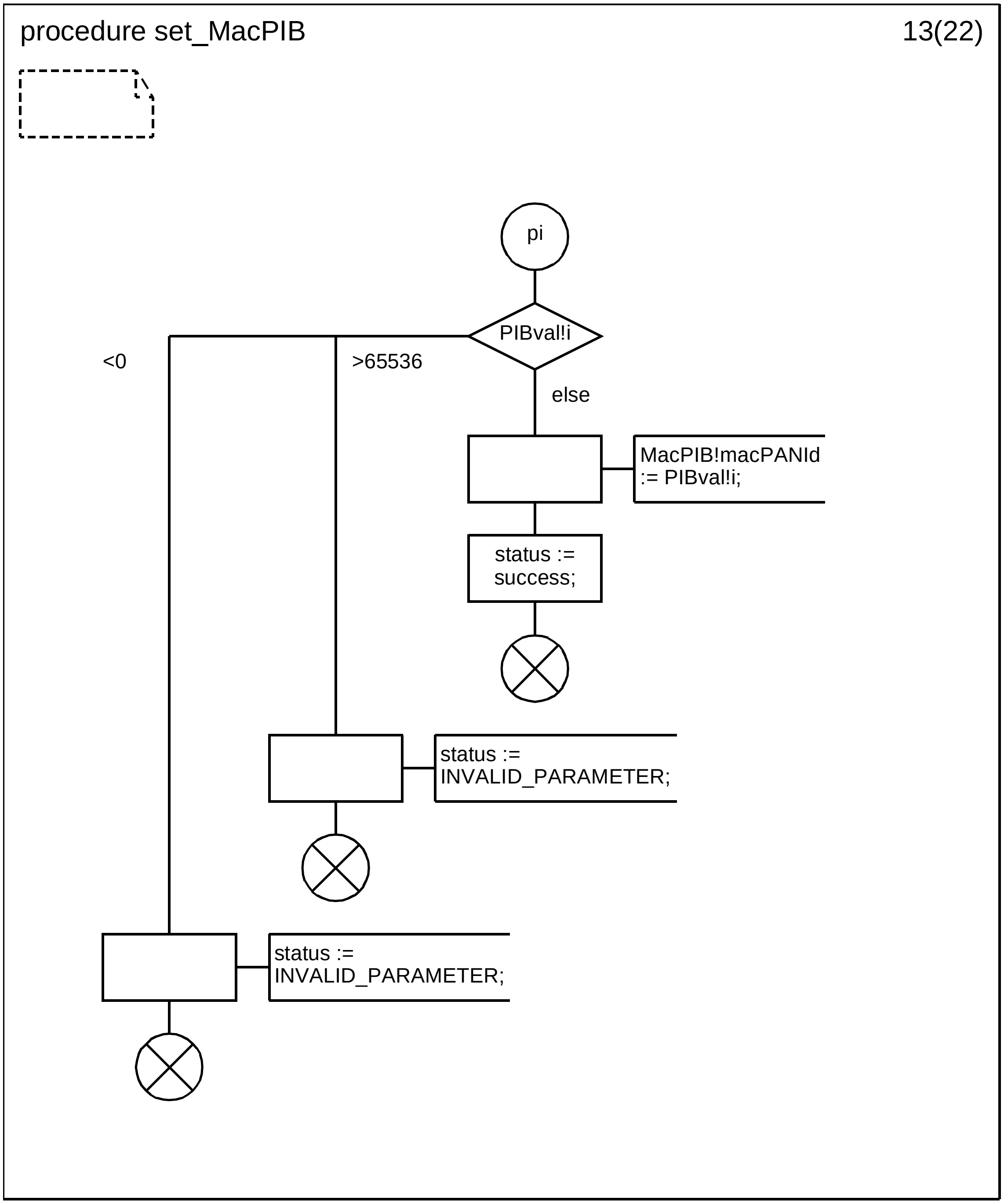
D.3.1.154.109 Procedure set_MacPIB (12)

.



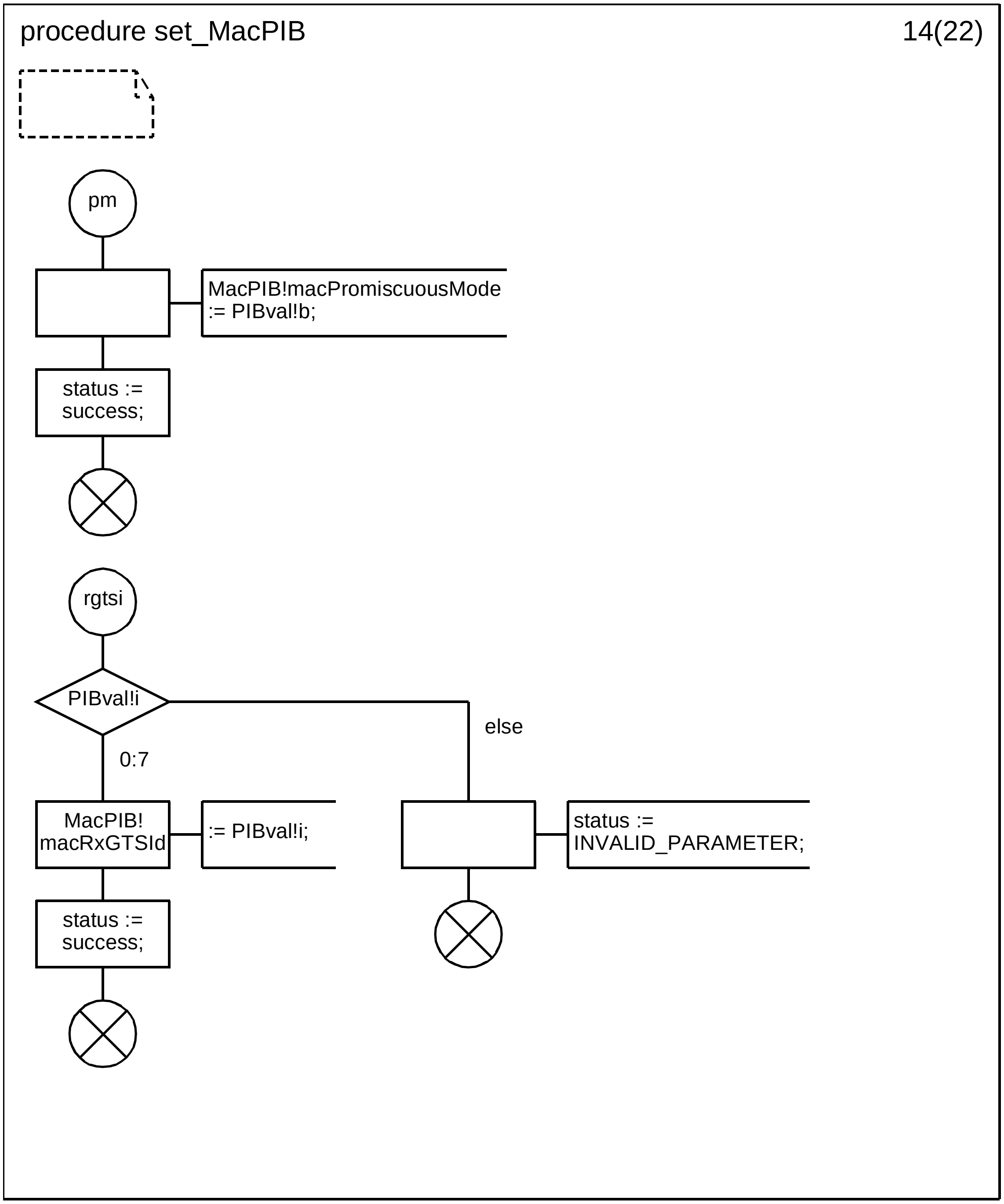
D.3.1.154.110 Procedure set_MacPIB (13)

.



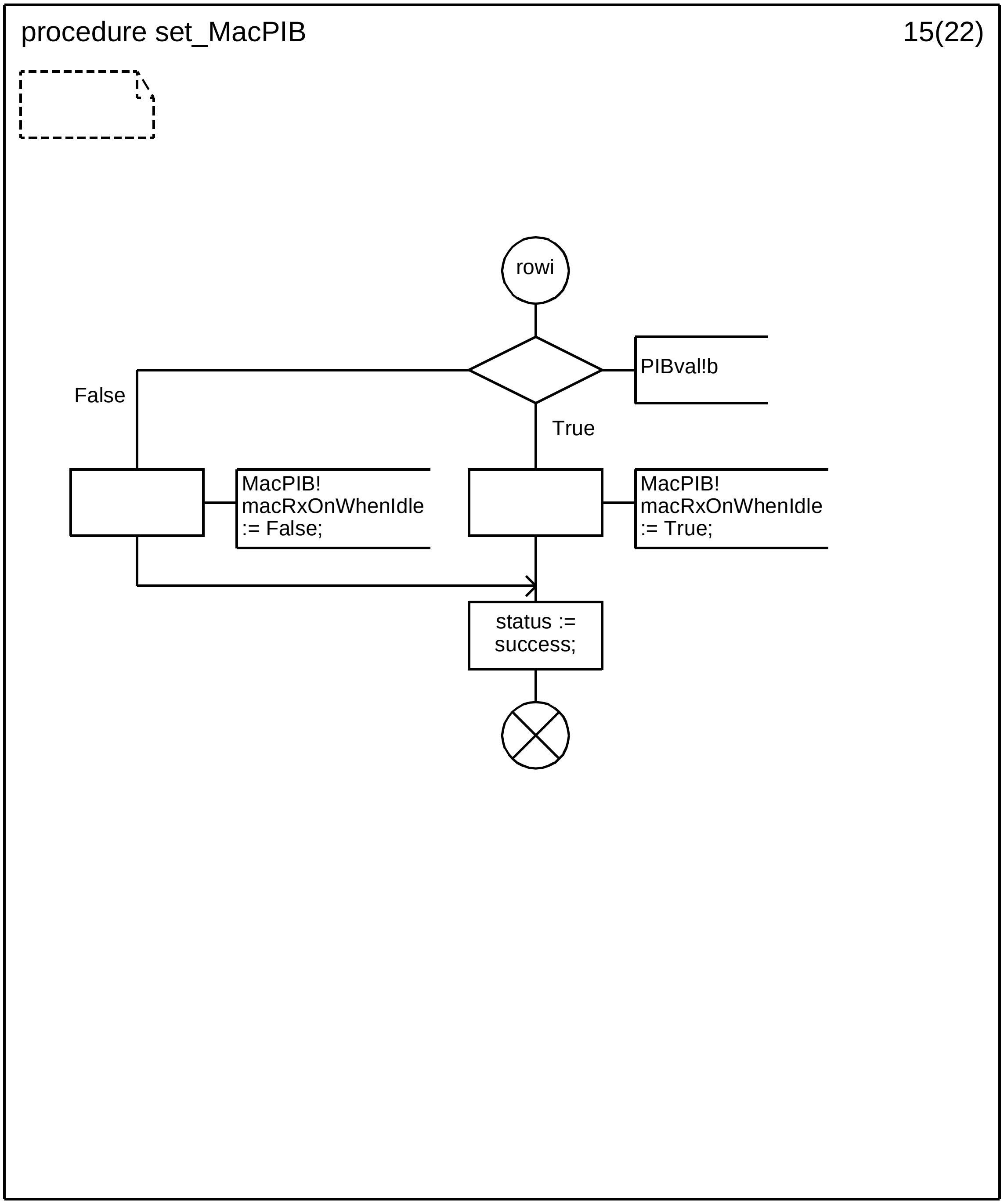
D.3.1.154.111 Procedure set_MacPIB (14)

.



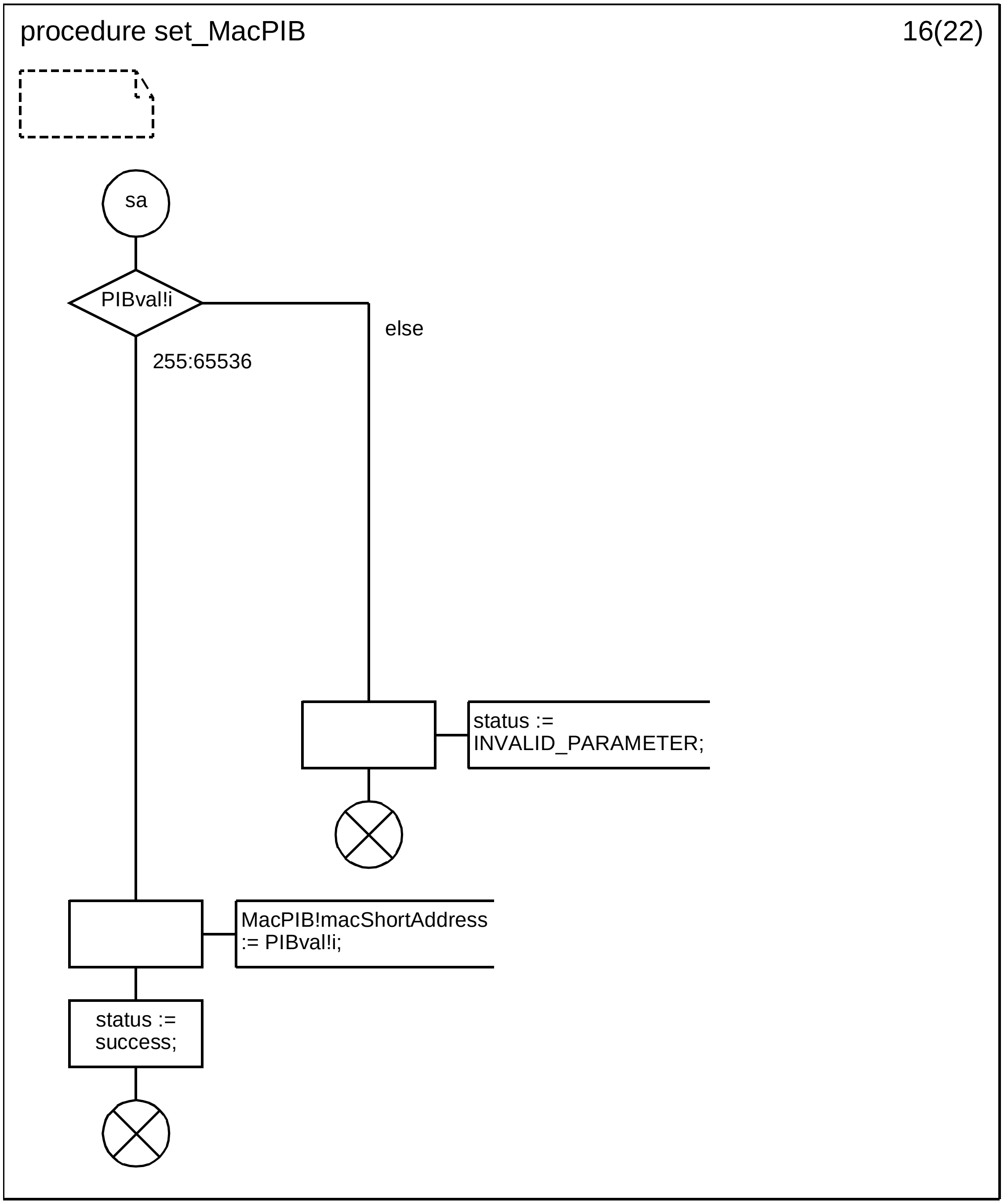
D.3.1.154.112 Procedure set_MacPIB (15)

.



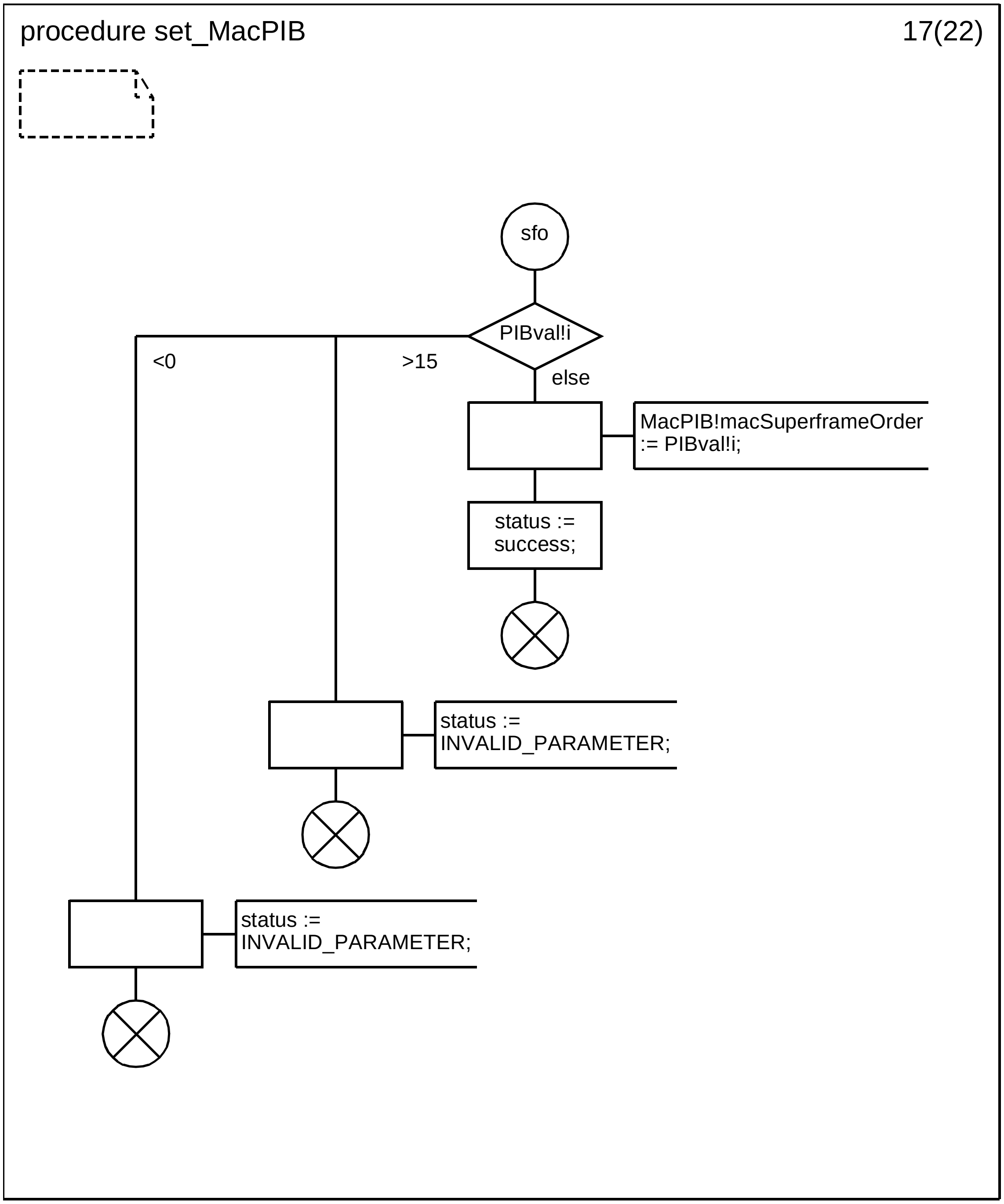
D.3.1.154.113 Procedure set_MacPIB (16)

.



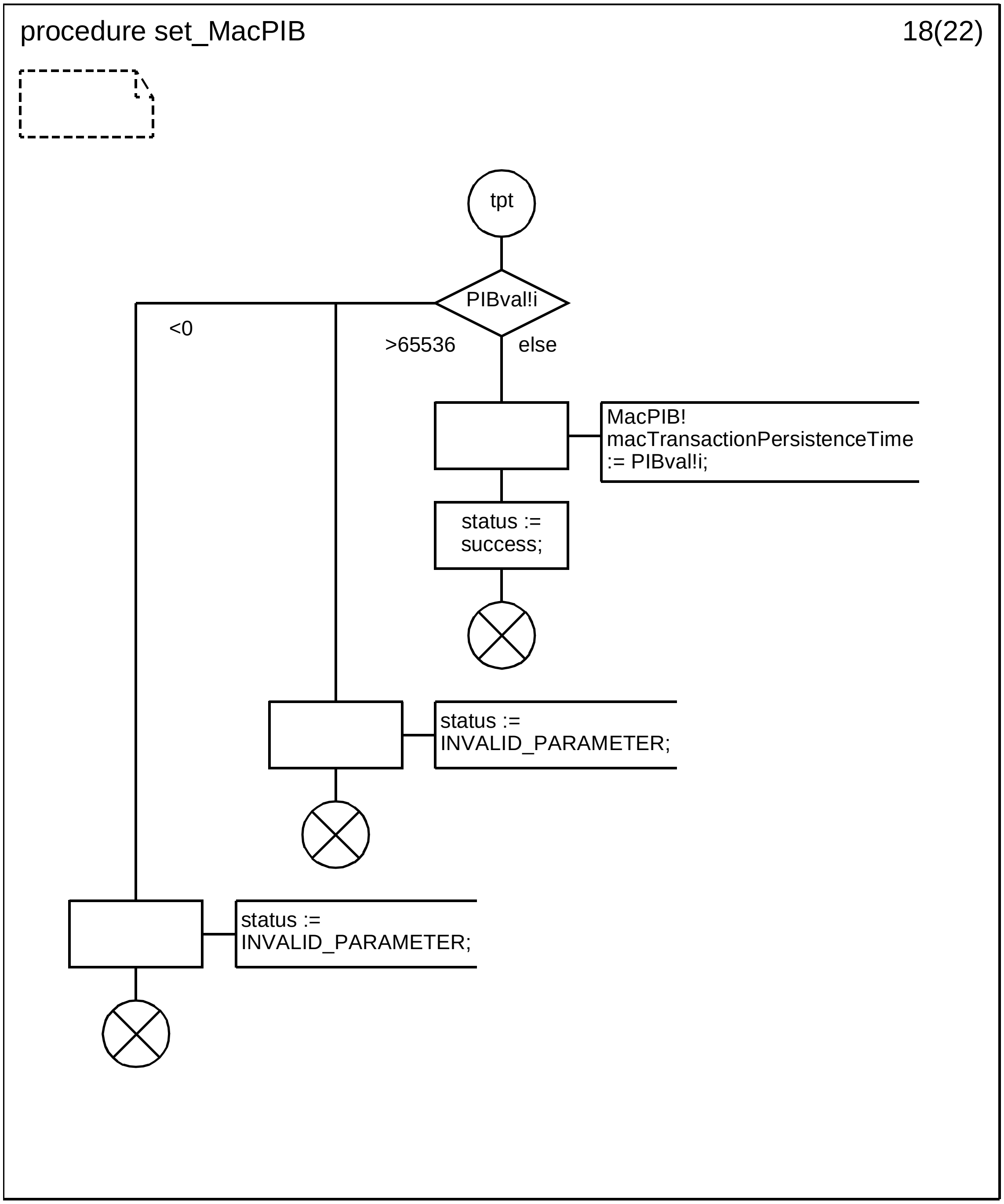
D.3.1.154.114 Procedure set_MacPIB (17)

.



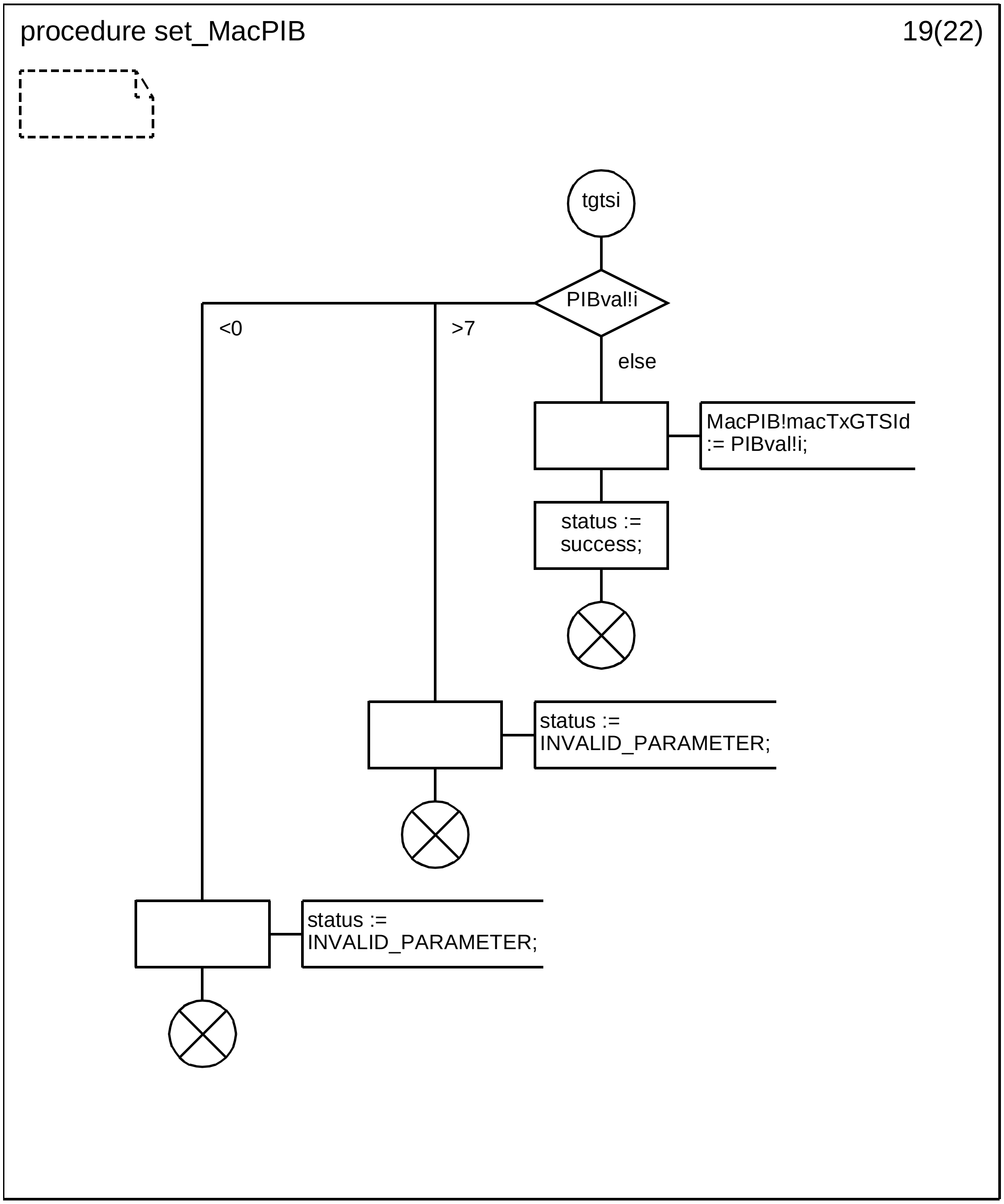
D.3.1.154.115 Procedure set_MacPIB (18)

.



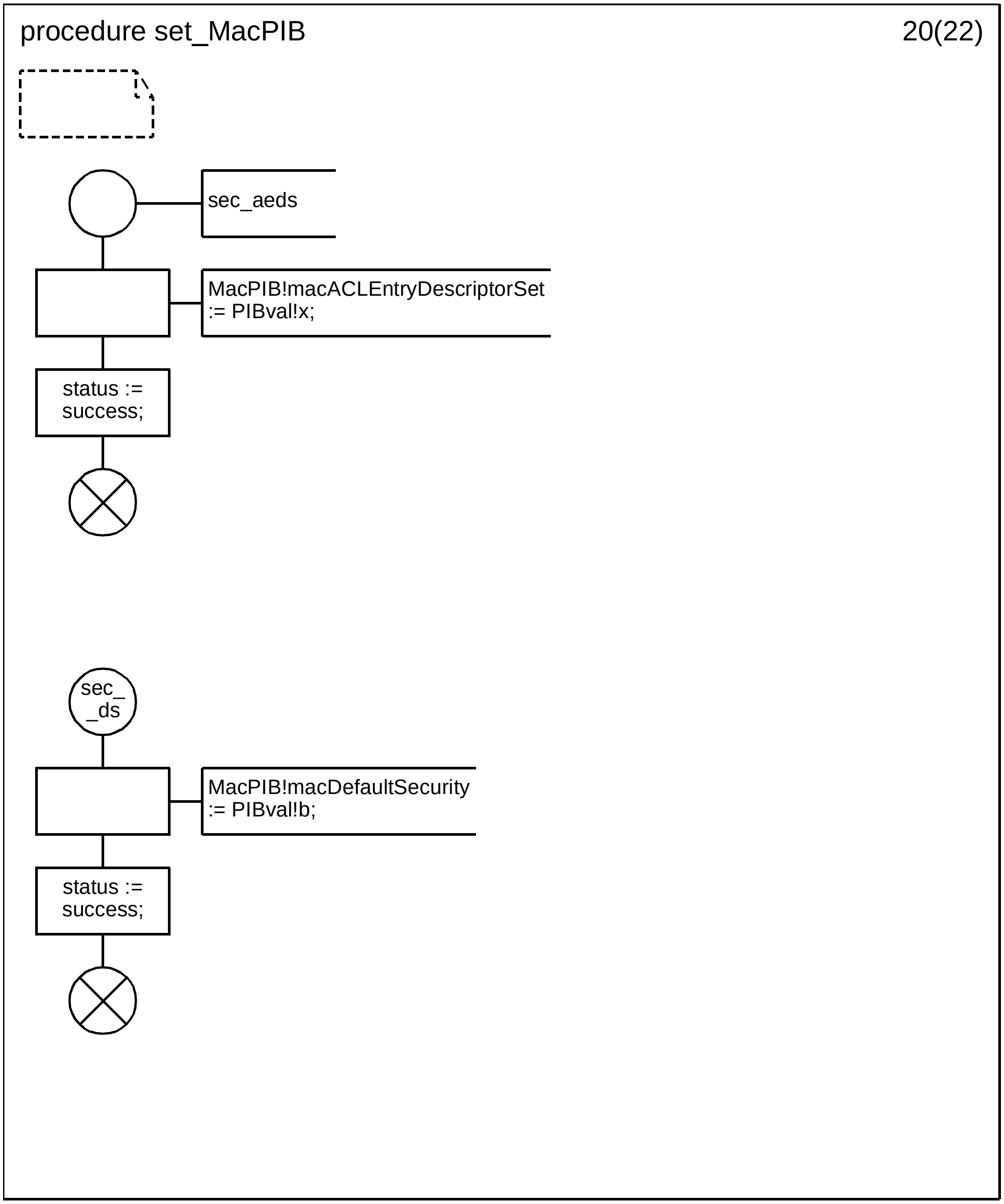
D.3.1.154.116 Procedure set_MacPIB (19)

.



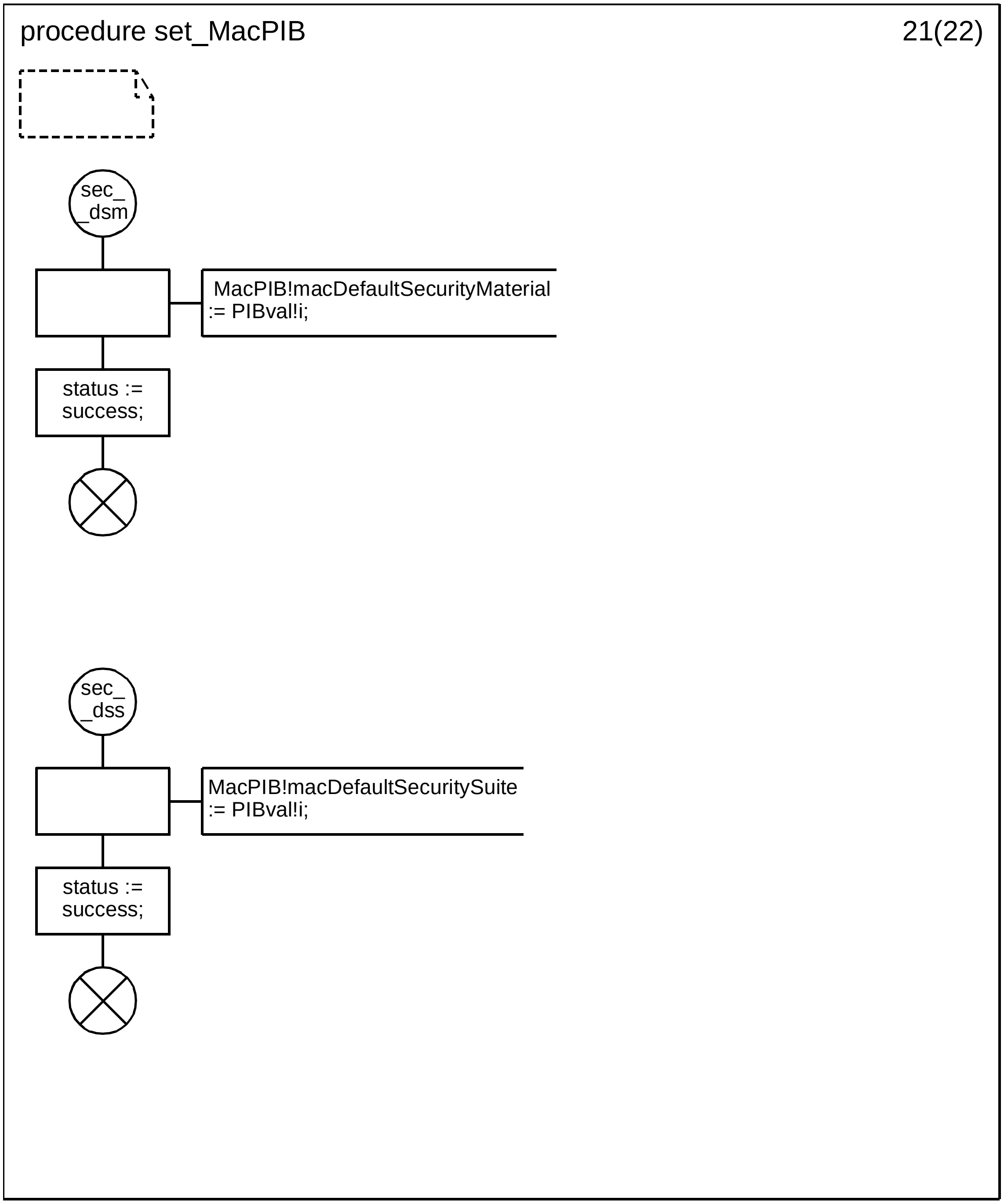
D.3.1.154.117 Procedure set_MacPIB (20)

.



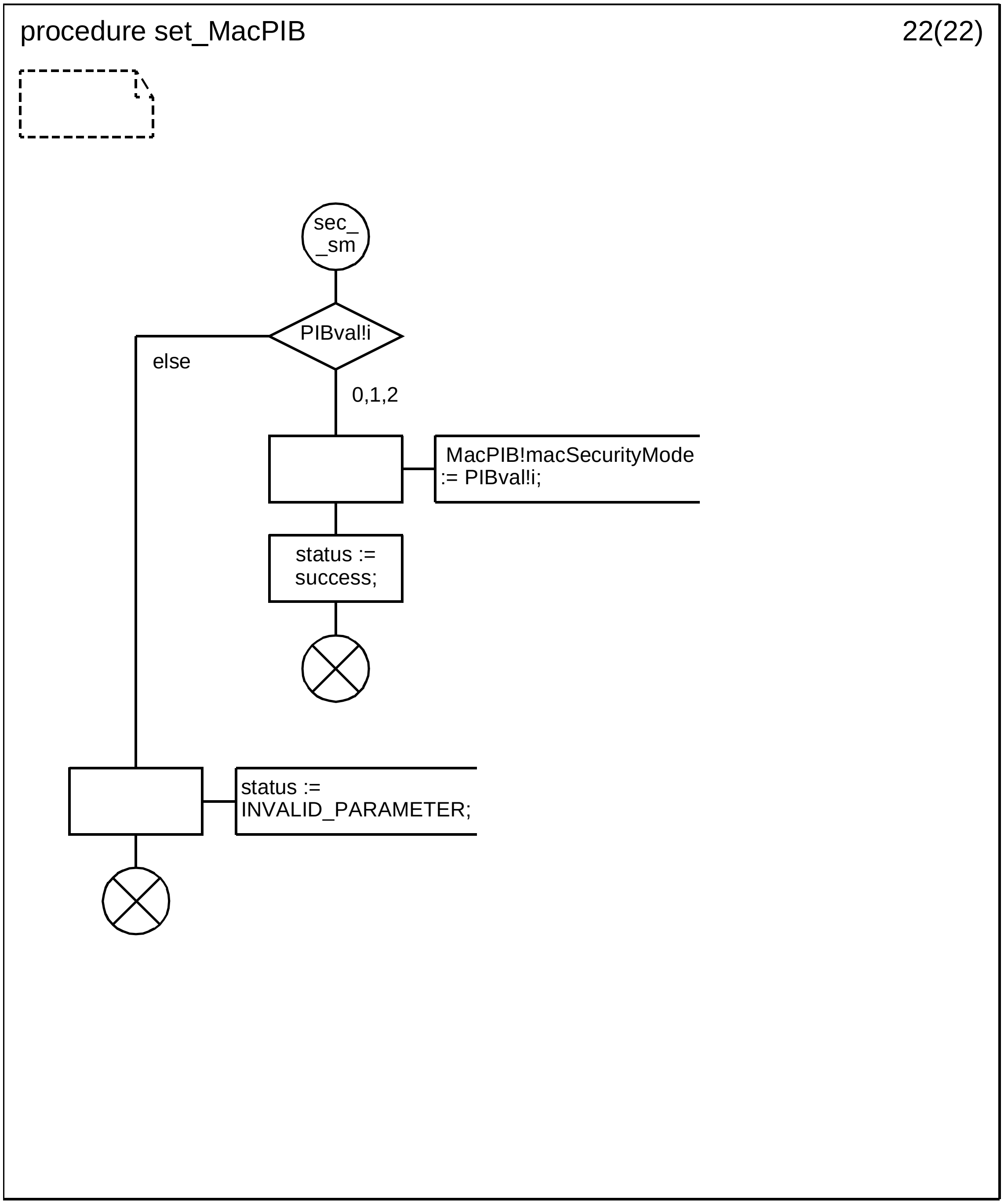
D.3.1.154.118 Procedure set_MacPIB (21)

.



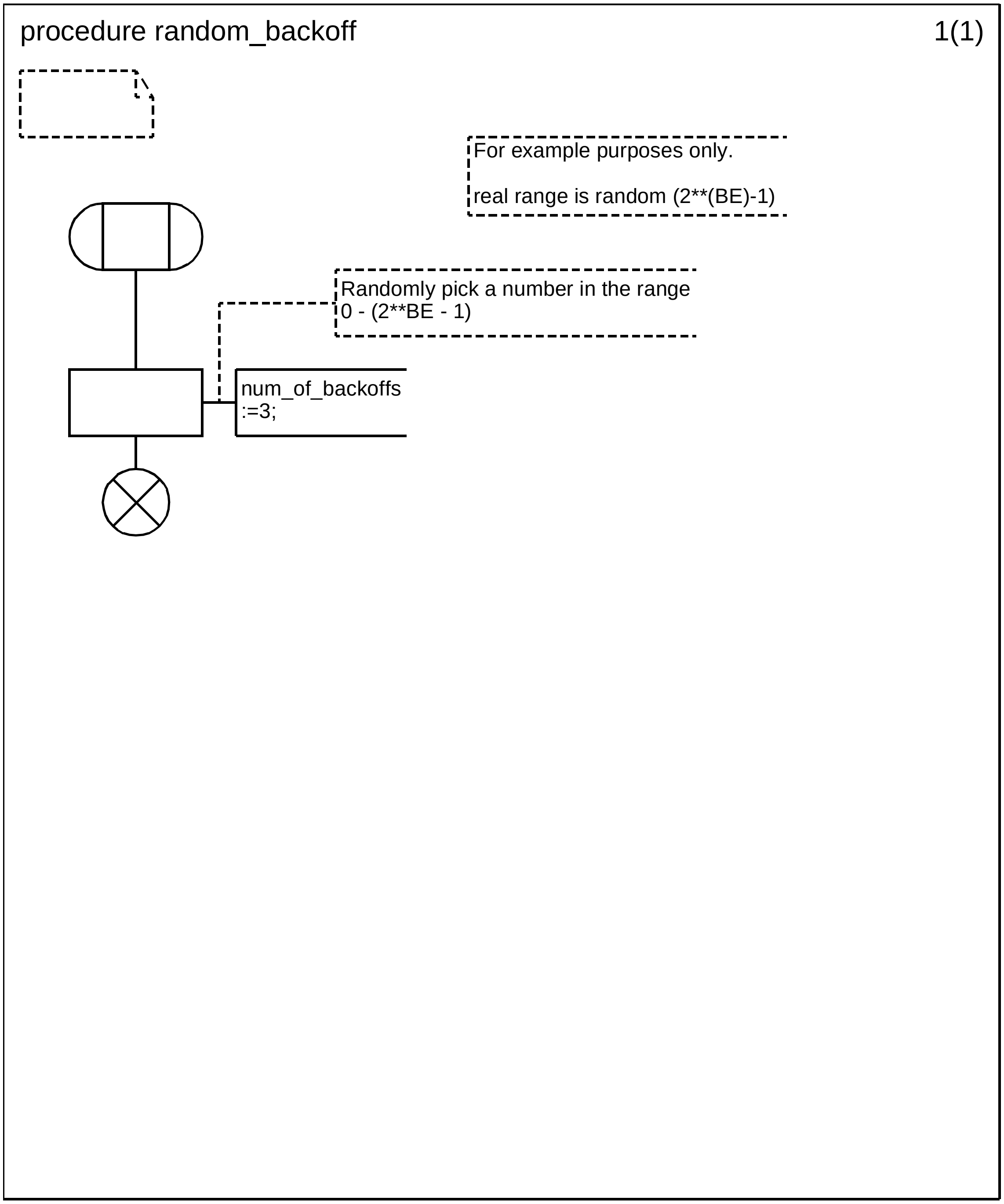
D.3.1.154.119 Procedure set_MacPIB (22)

.



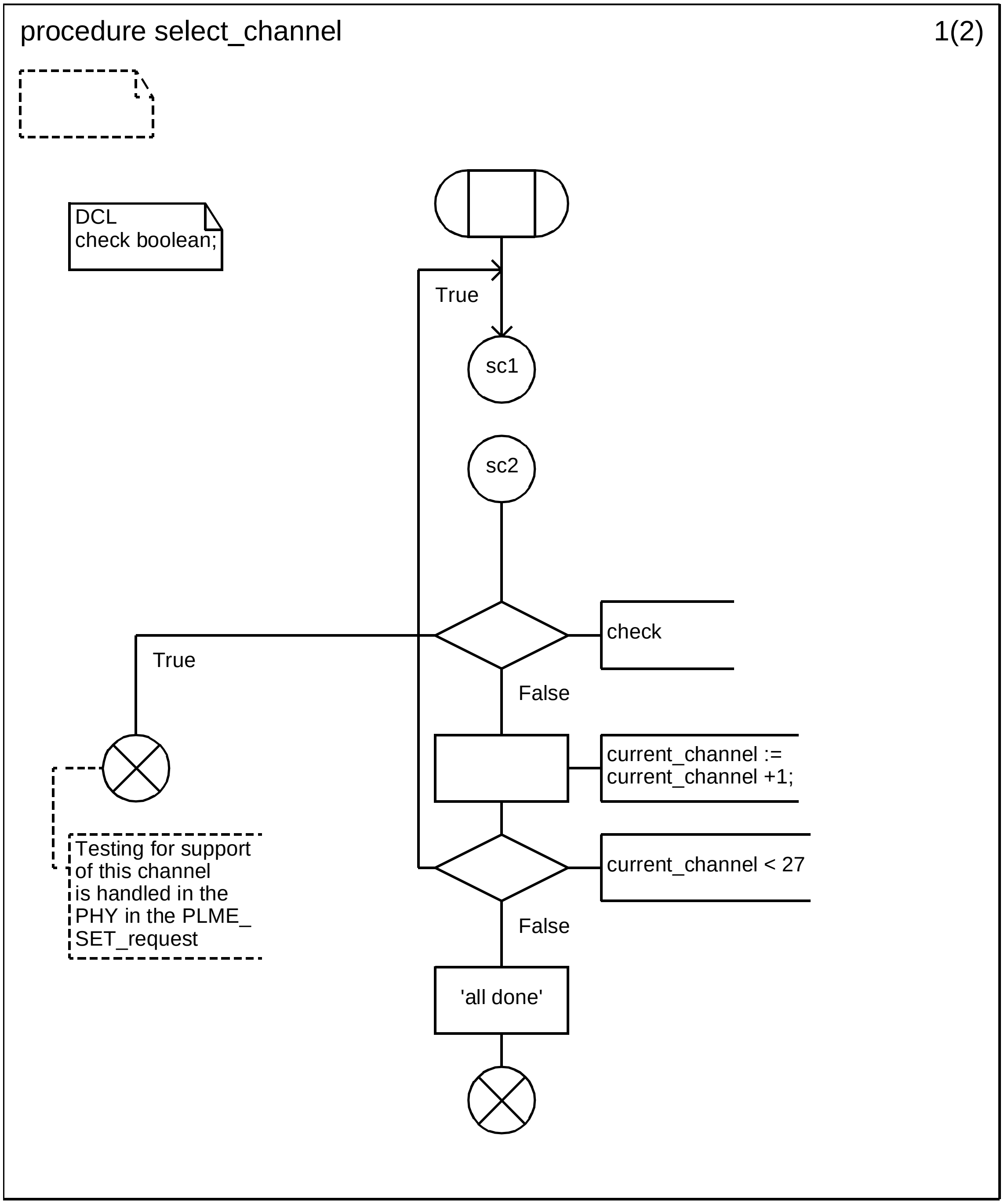
D.3.1.154.120 Procedure random_backoff

.


procedure random_backoff
1(1)
For example purposes only.
real range is random (2**(BE)-1)
Randomly pick a number in the range
0 - (2**BE - 1)
num_of_backoffs
:=3;

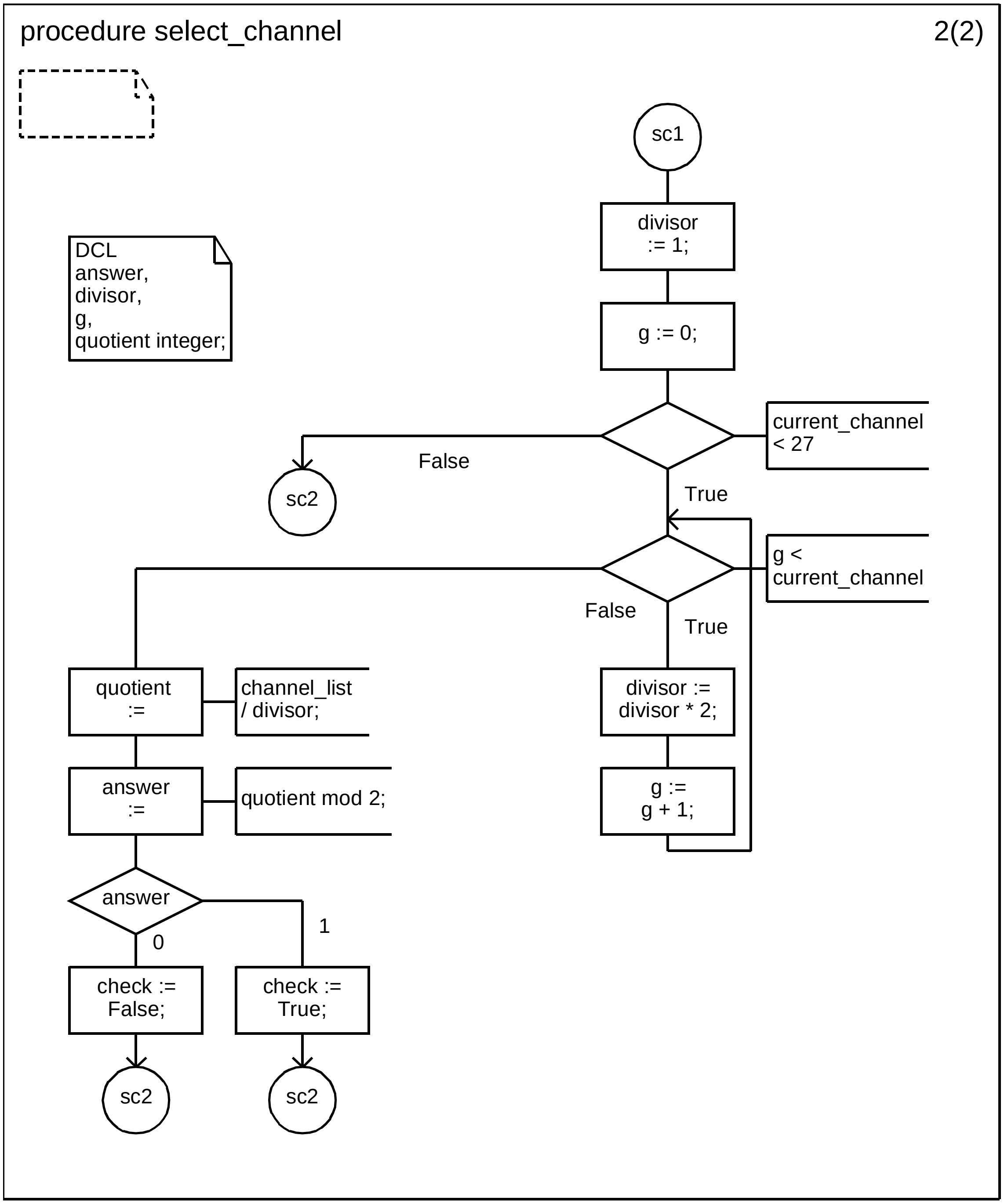
D.3.1.154.121 Procedure select_channel (1)

.



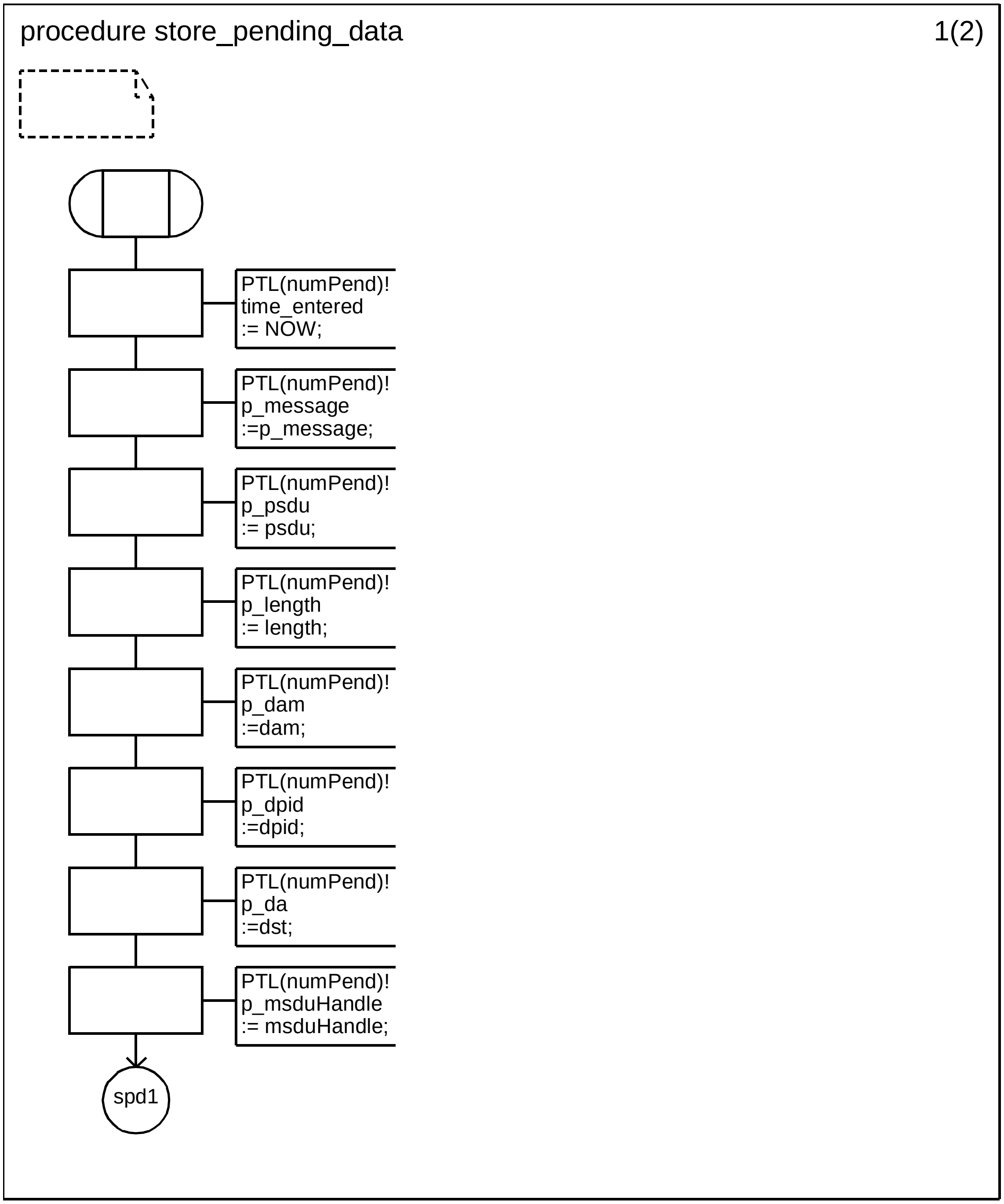
D.3.1.154.122 Procedure select_channel (2)

.



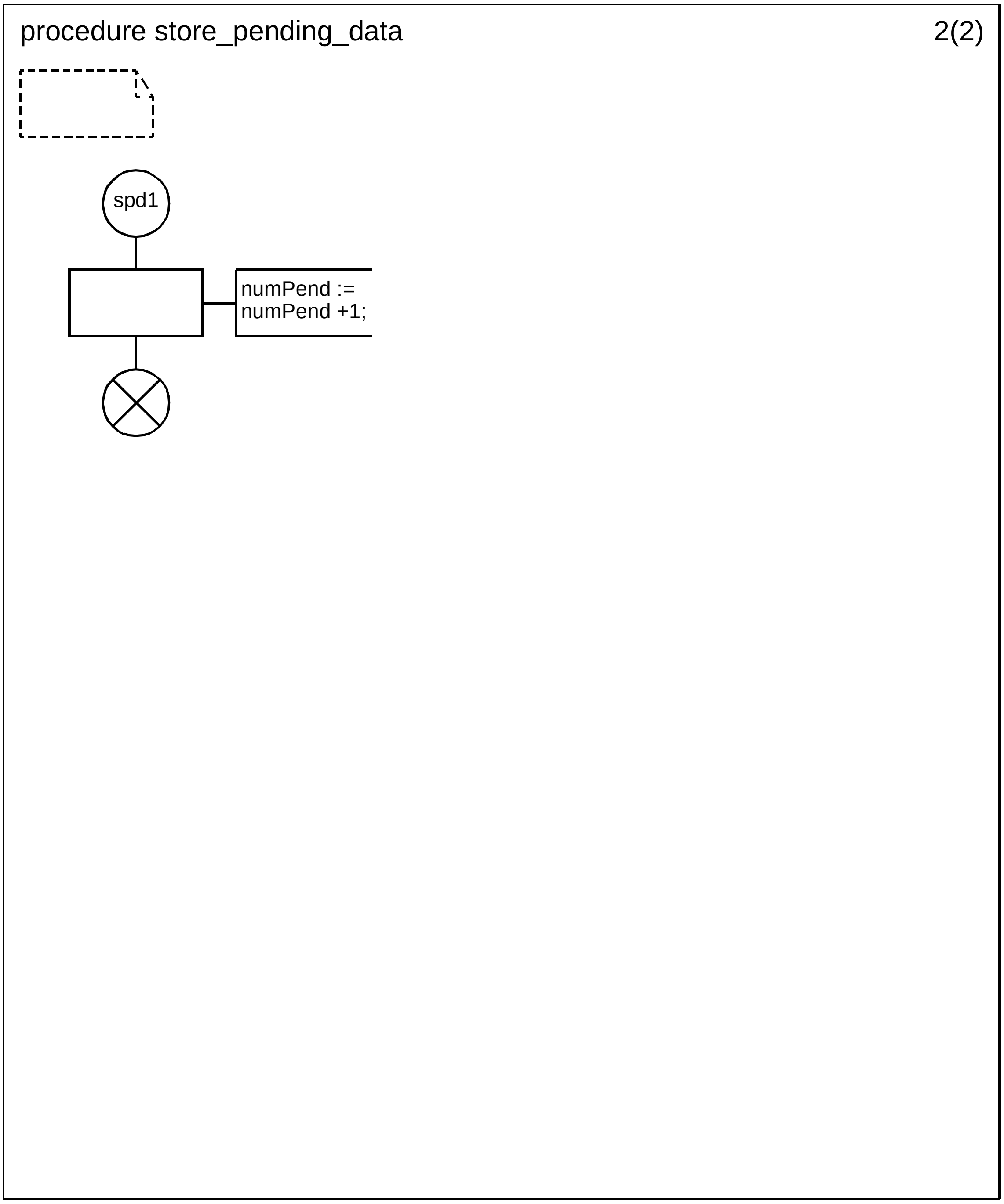
D.3.1.154.122.1 Procedure store_pending_data (1)

.



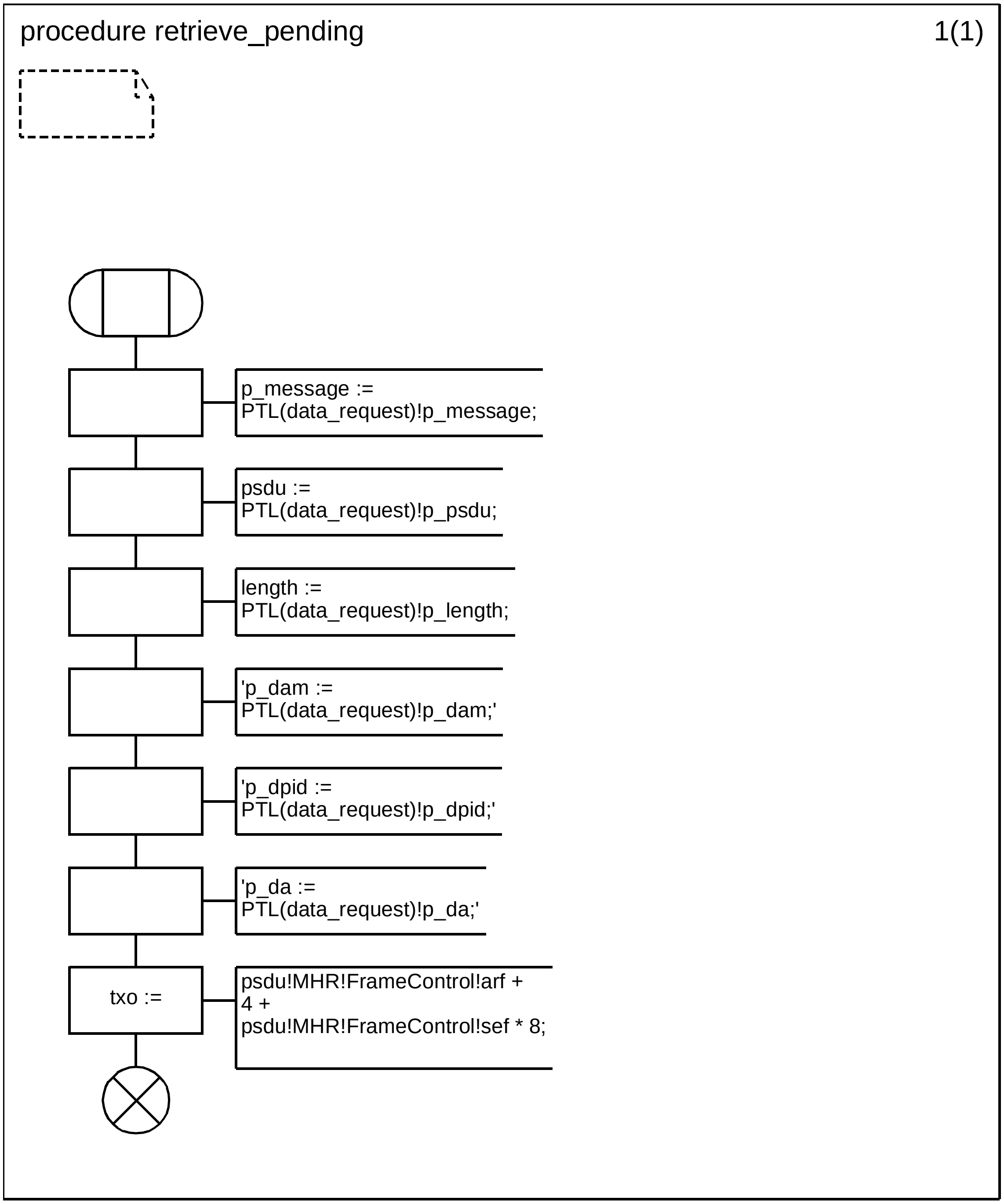
D.3.1.154.122.2 Procedure store_pending_data (2)

.



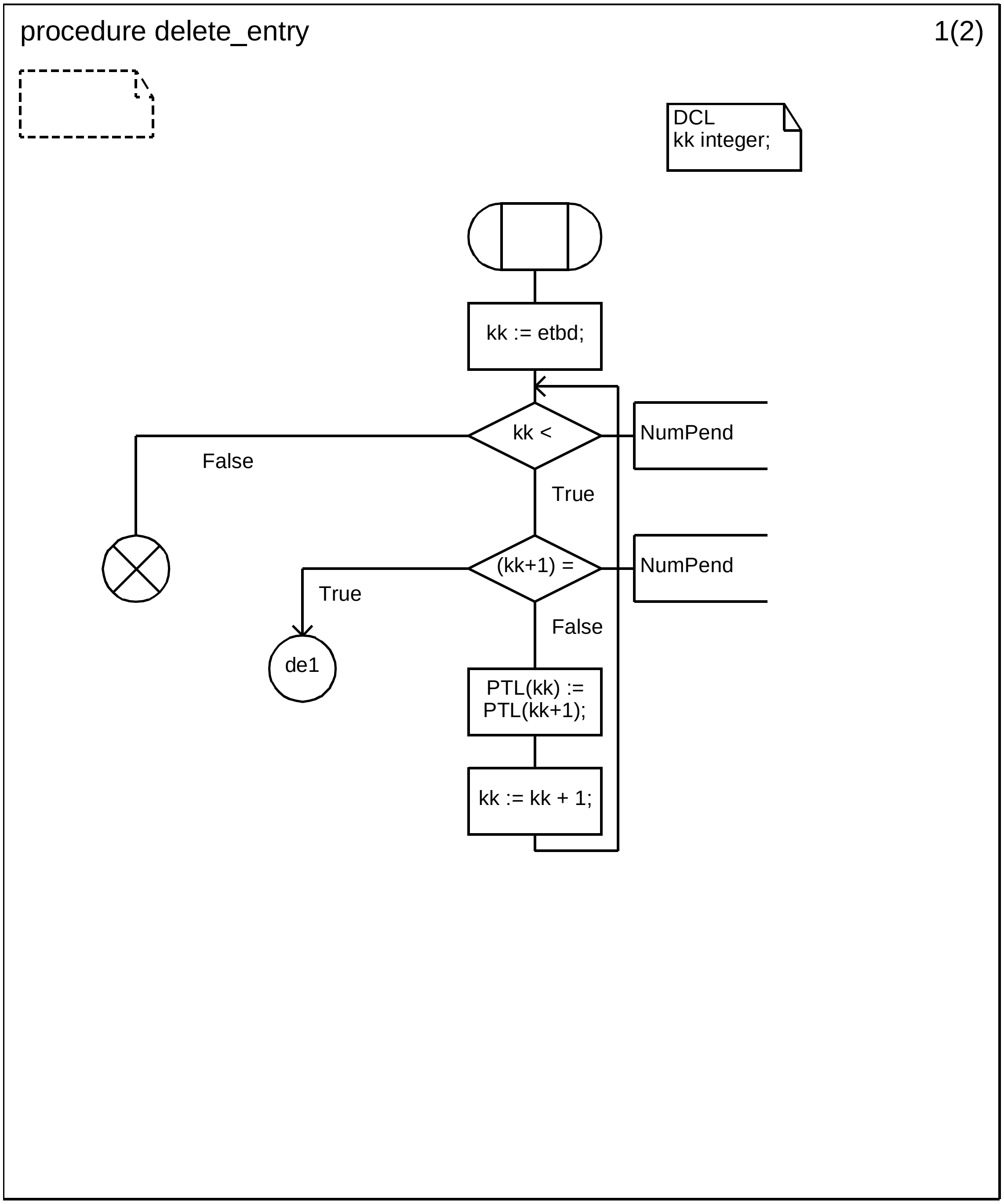
D.3.1.154.122.3 Procedure retrieve_pending

.


procedure retrieve_pending
1(1)
p_message :=
PTL(data_request)!p_message;
psdu :=
PTL(data_request)!p_psdu;
length :=
PTL(data_request)!p_length;
'p_dam :=
PTL(data_request)!p_dam;'
'p_dpid :=
PTL(data_request)!p_dpid;'
'p_da :=
PTL(data_request)!p_da;'
txo :=
psdu!MHR!FrameControl!arf +
4 +
psdu!MHR!FrameControl!sef * 8;

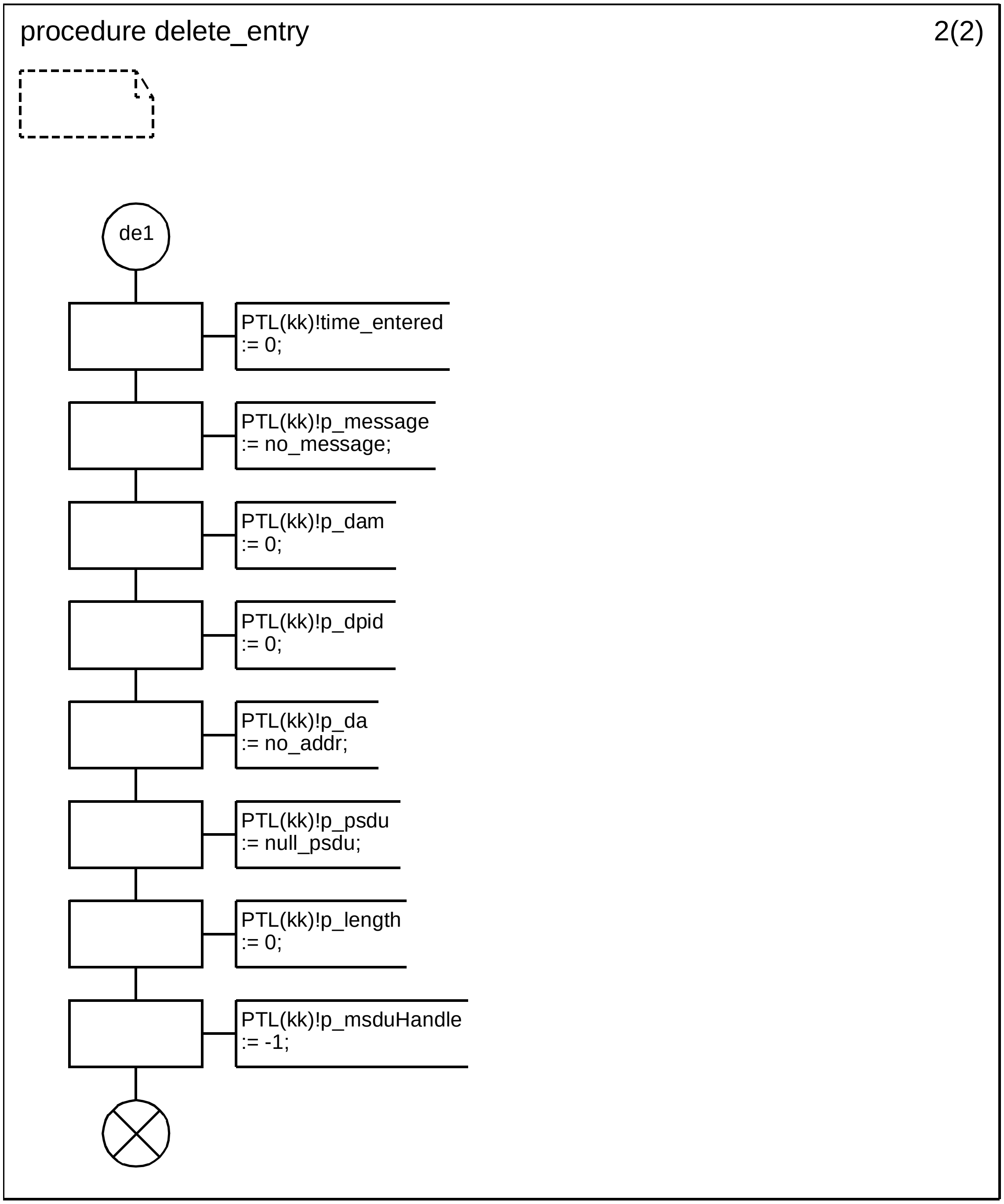
D.3.1.154.122.4 Procedure delete_entry (1)

.



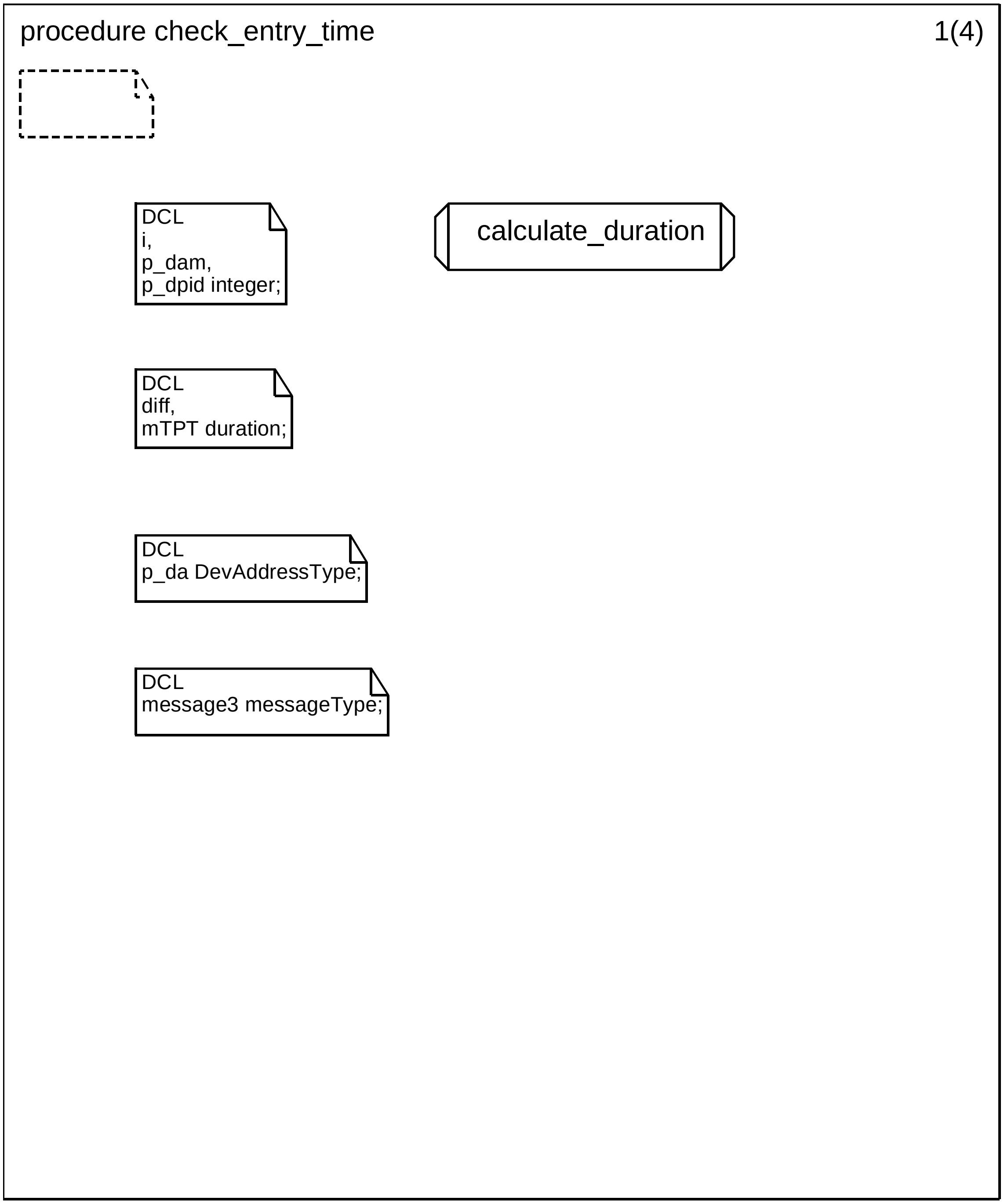
D.3.1.154.122.5 Procedure delete_entry (2)

.



D.3.1.154.122.6 Procedure check_entry_time (1)

.



```
DCL
i,
p_dam,
p_dpid integer;
```

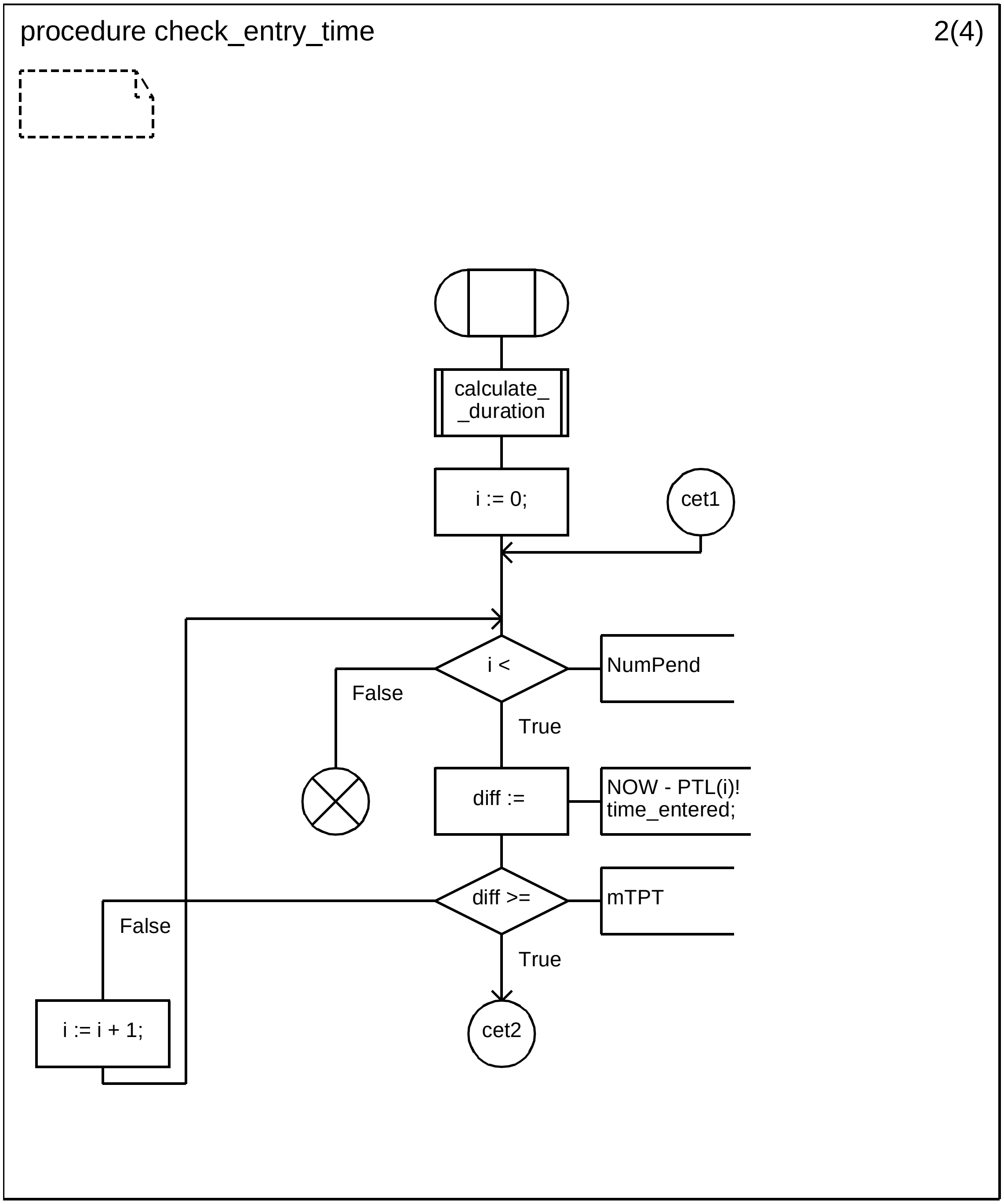
```
DCL
diff,
mTPT duration;
```

```
DCL
p_da DevAddressType;
```

```
DCL
message3 messageType;
```

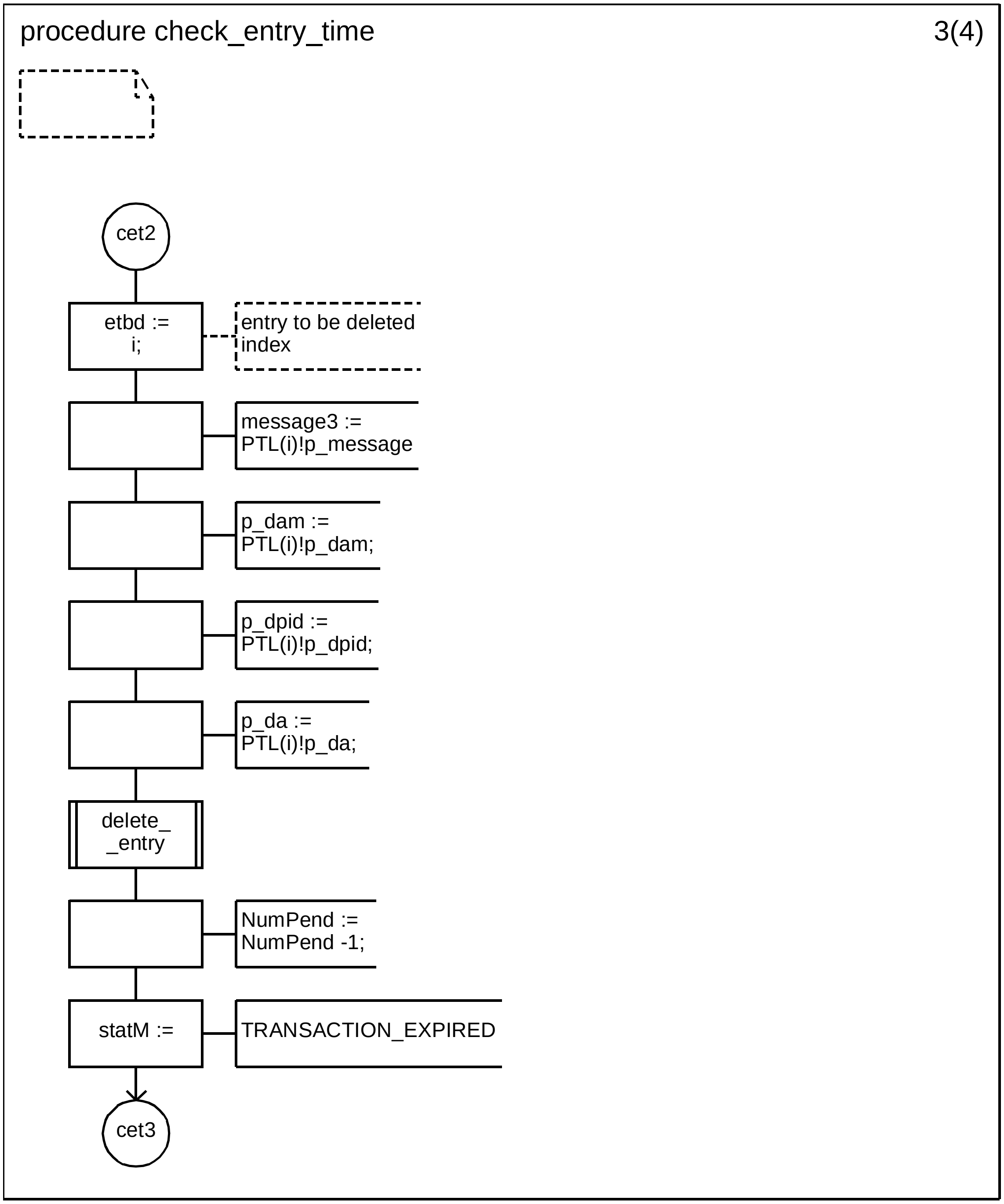
D.3.1.154.122.7 Procedure check_entry_time (2)

.



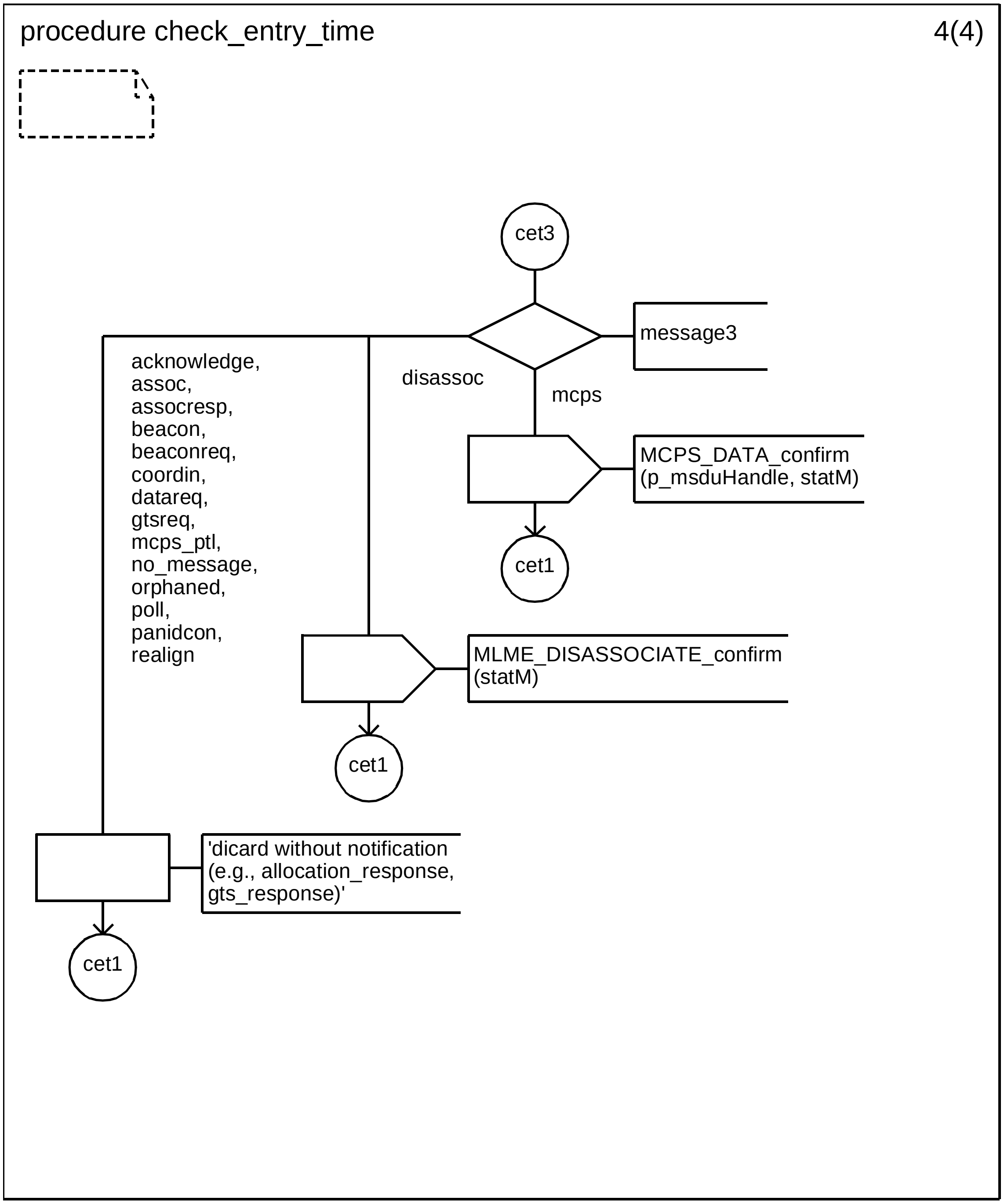
D.3.1.154.122.8 Procedure check_entry_time (3)

.



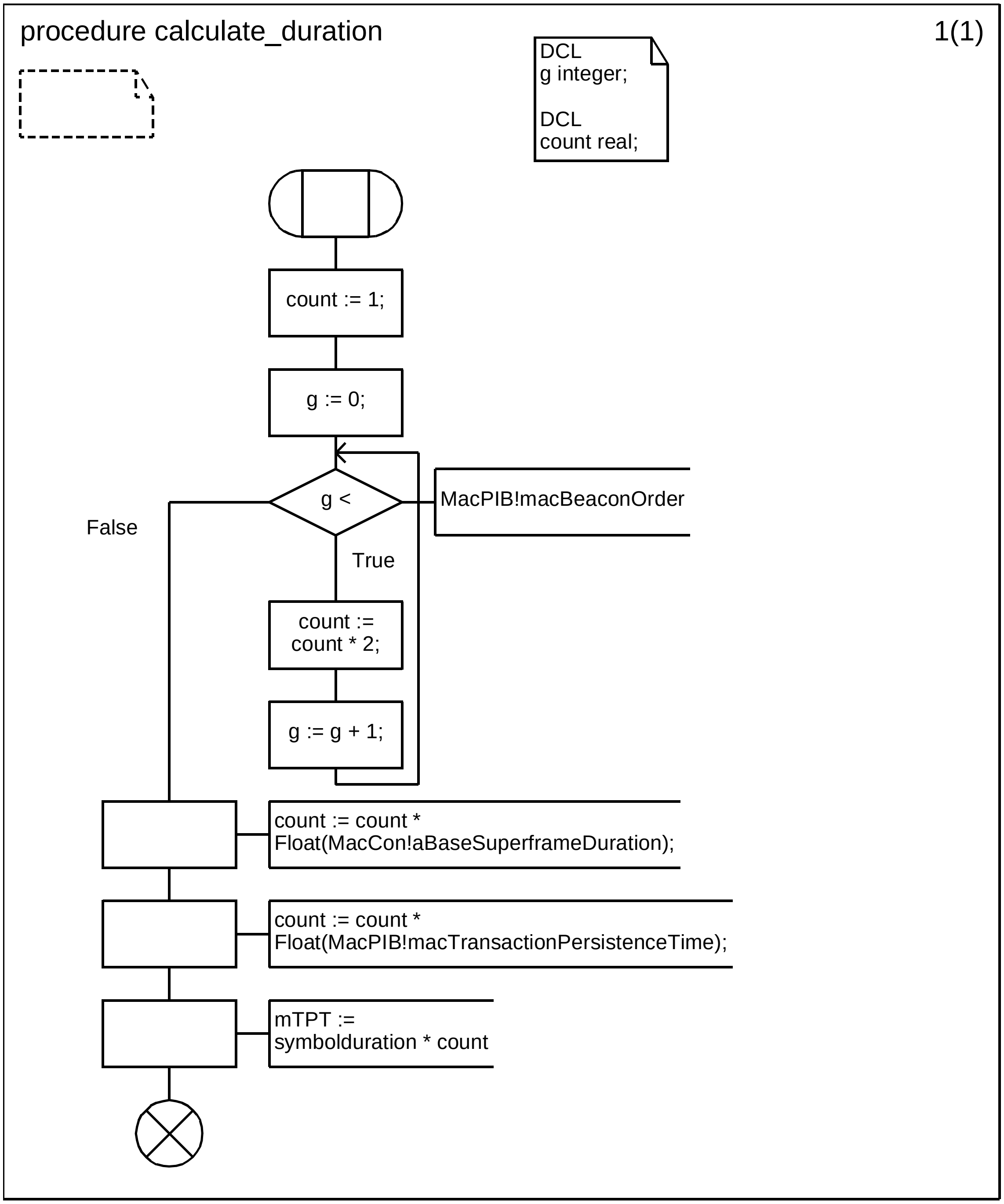
D.3.1.154.122.9 Procedure check_entry_time (4)

.



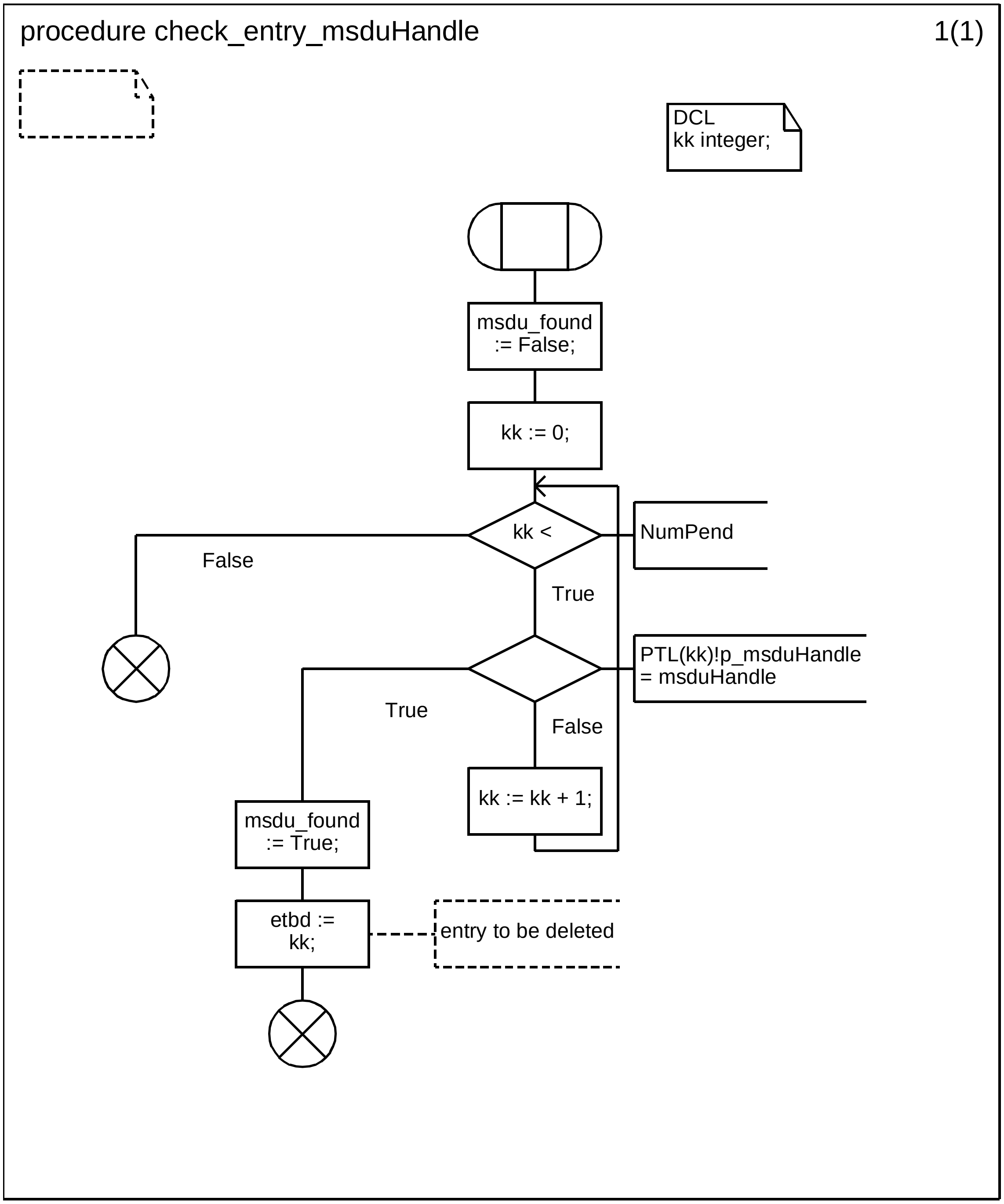
D.3.1.154.122.10 Procedure calculate_duration

.



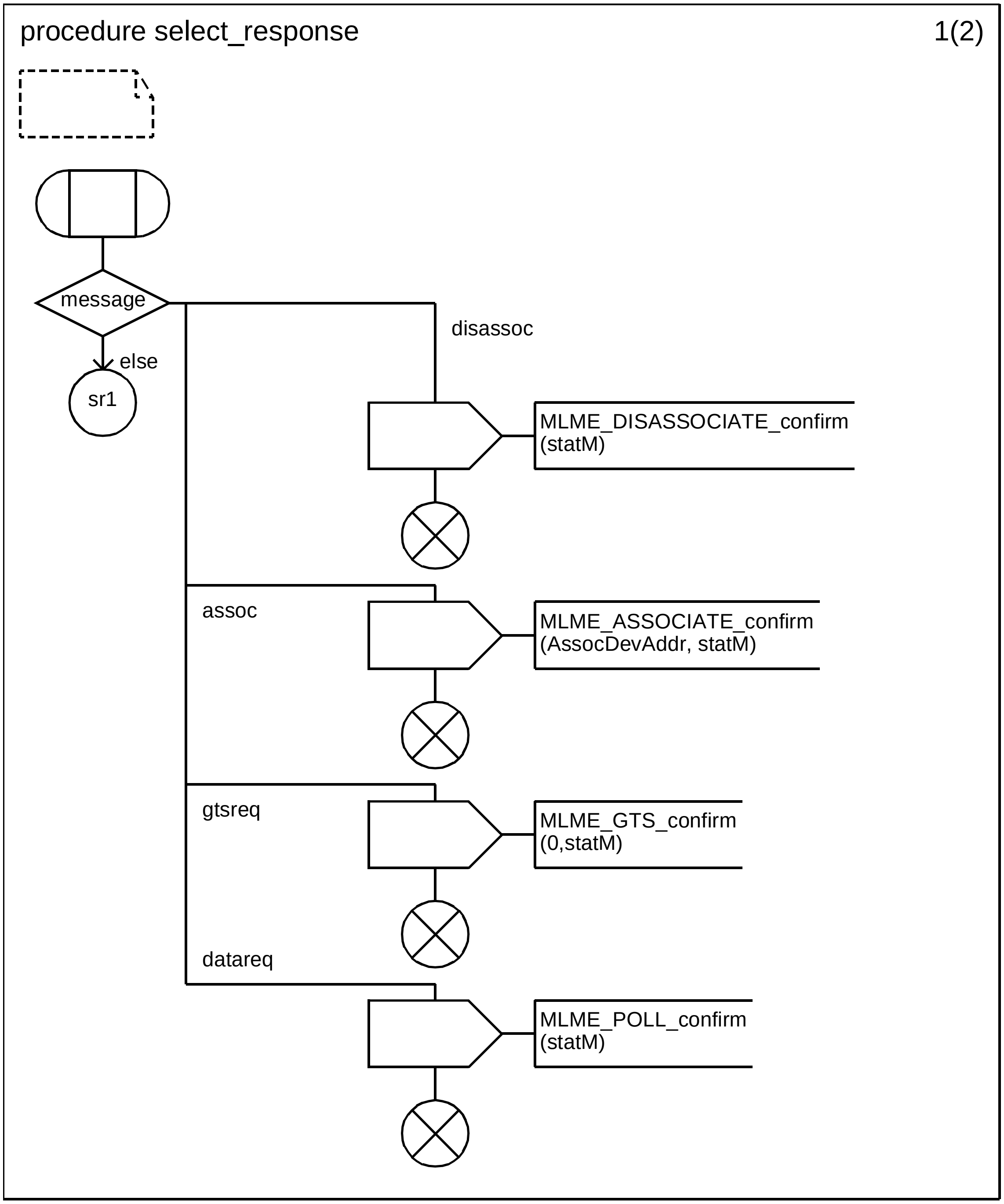
D.3.1.154.123 Procedure check_entry_msduHandle

.

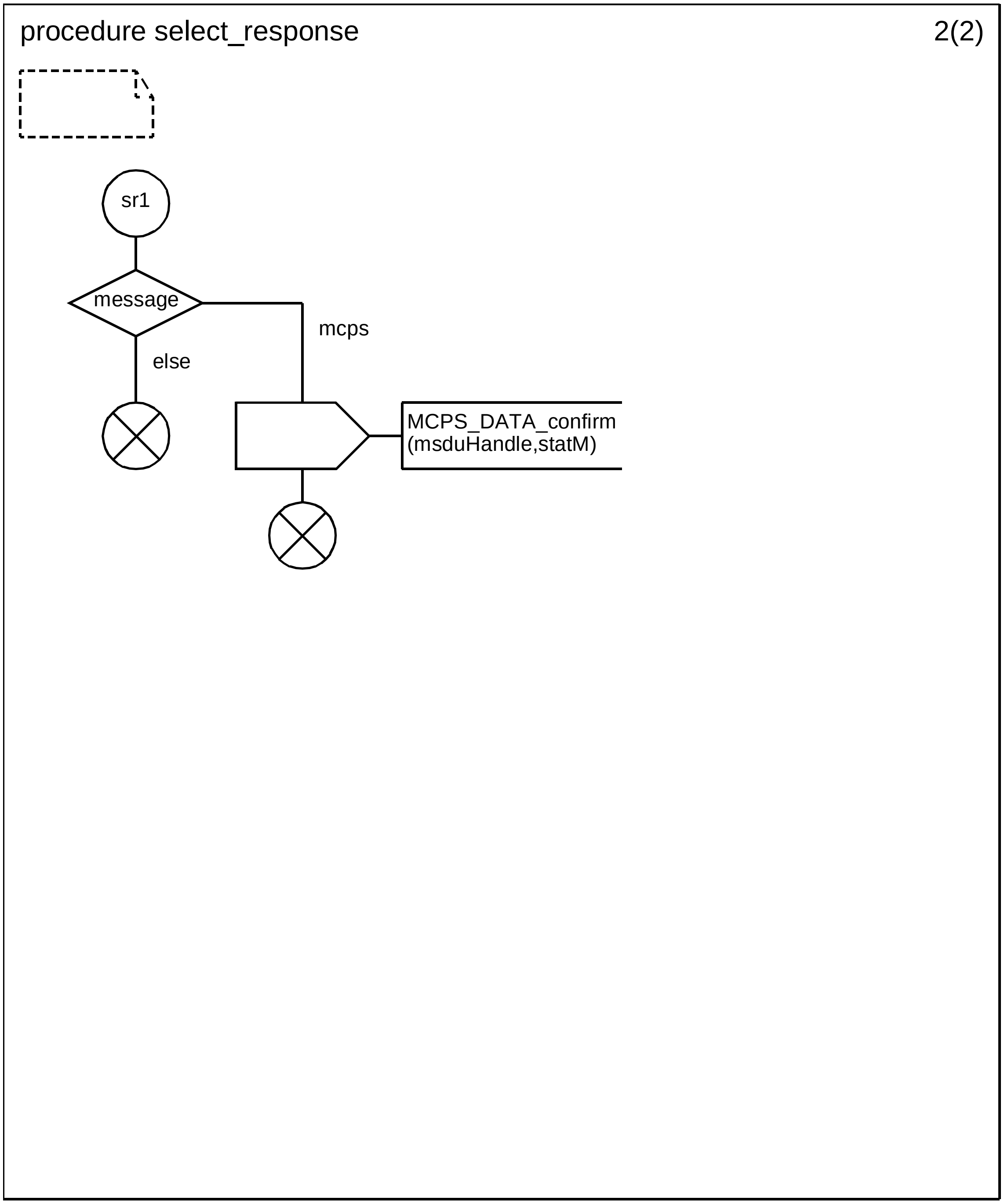


D.3.1.154.124 Procedure select_response (1)

.

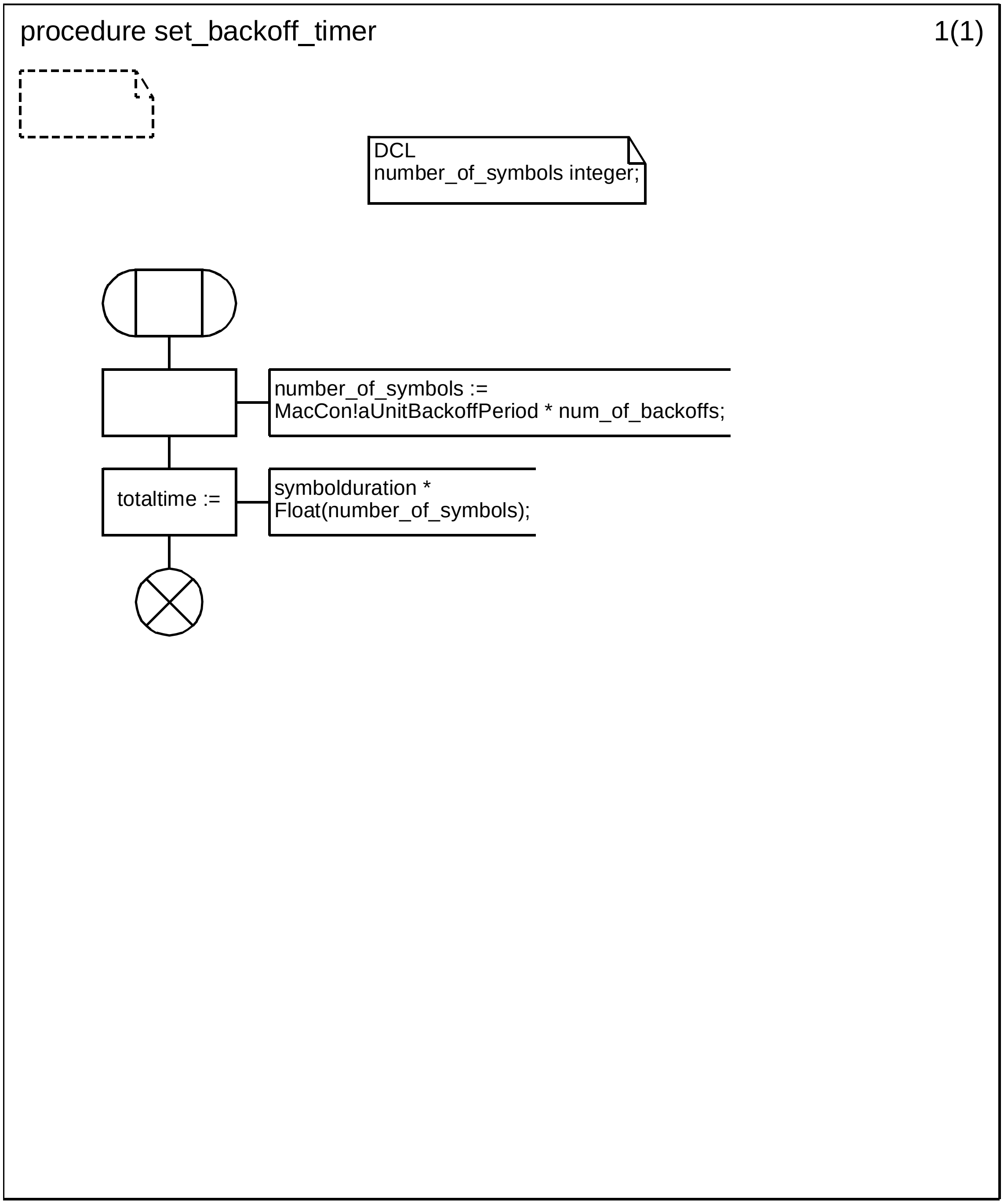


D.3.1.154.125 Procedure select_response (2)



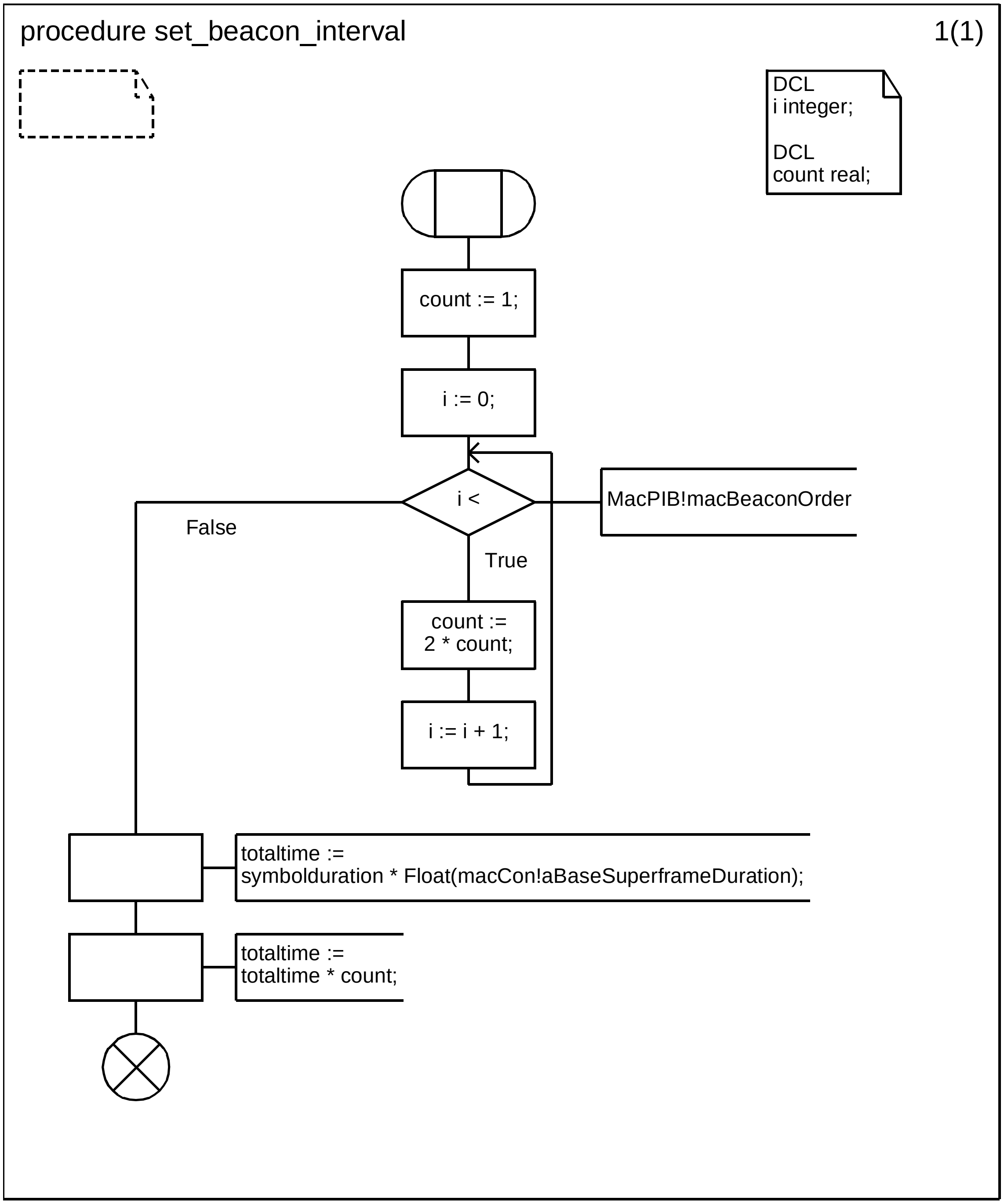
D.3.1.154.126 Procedure set_backoff_timer

.


procedure set_backoff_timer
1(1)
DCL
number_of_symbols integer;
number_of_symbols :=
MacCon!aUnitBackoffPeriod * num_of_backoffs;
totaltime :=
symbolduration *
Float(number_of_symbols);

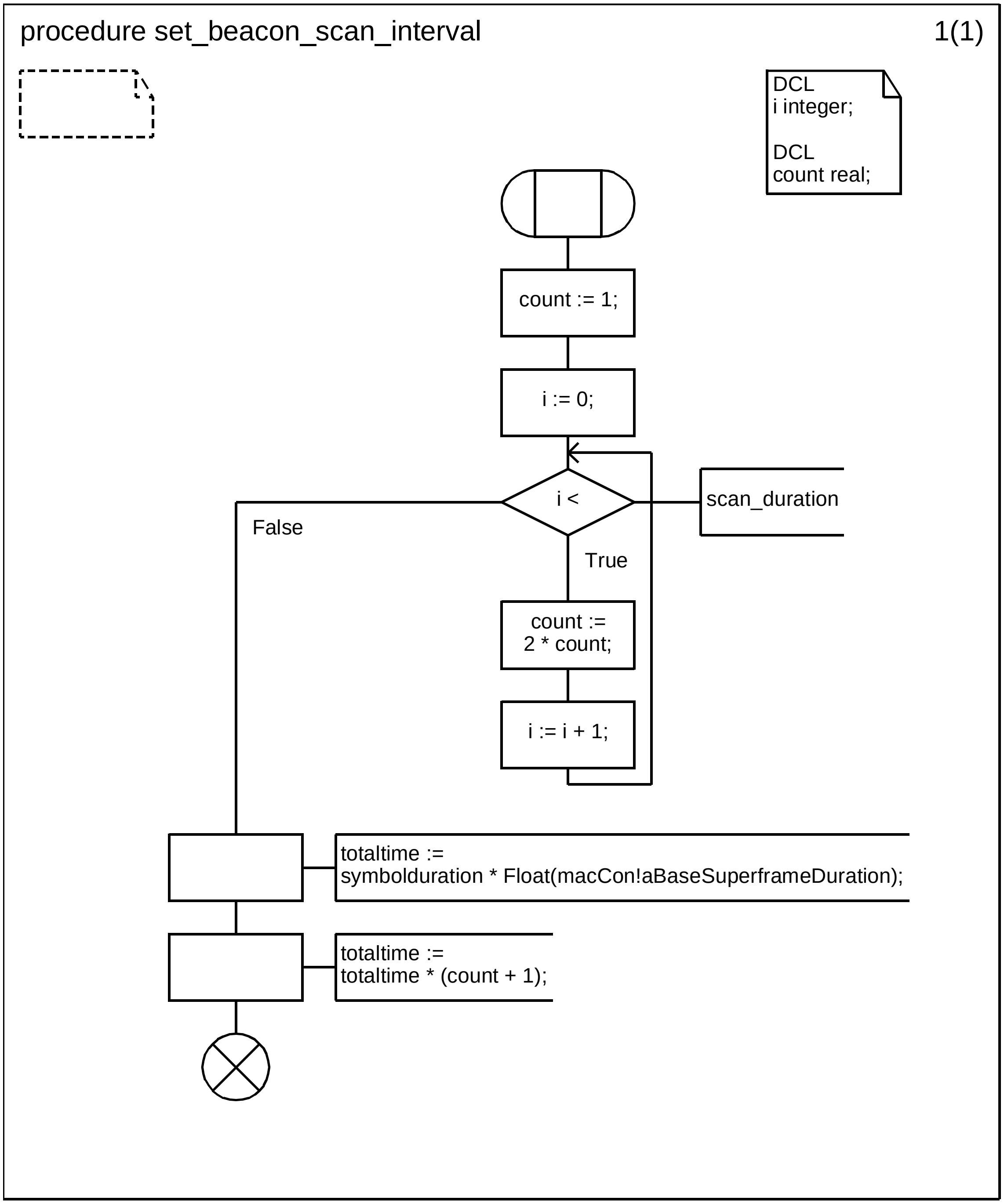
D.3.1.154.127 Procedure set_beacon_interval

.



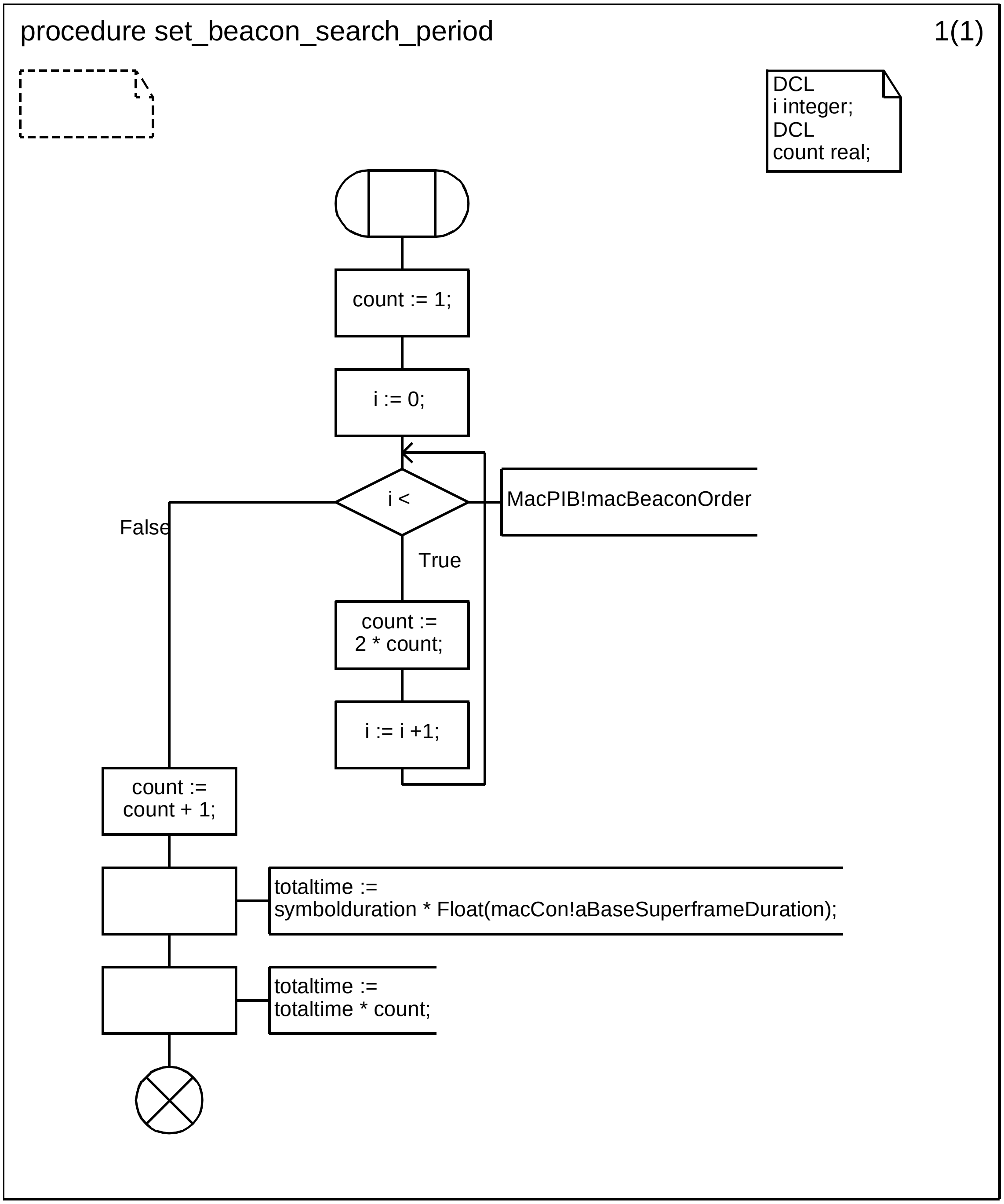
D.3.1.154.128 Procedure set_beacon_scan_interval

.



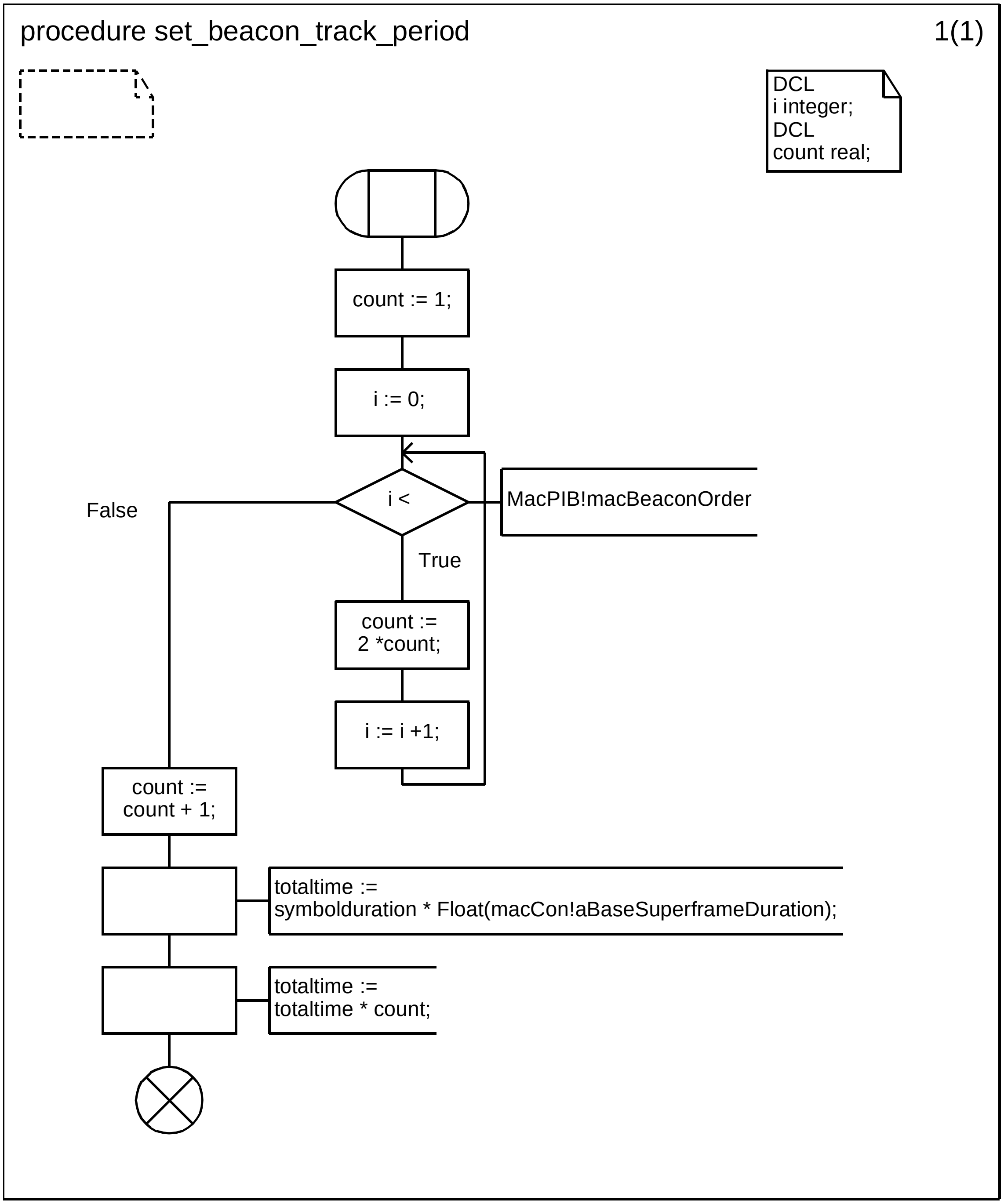
D.3.1.154.129 Procedure set_beacon_search_period

.



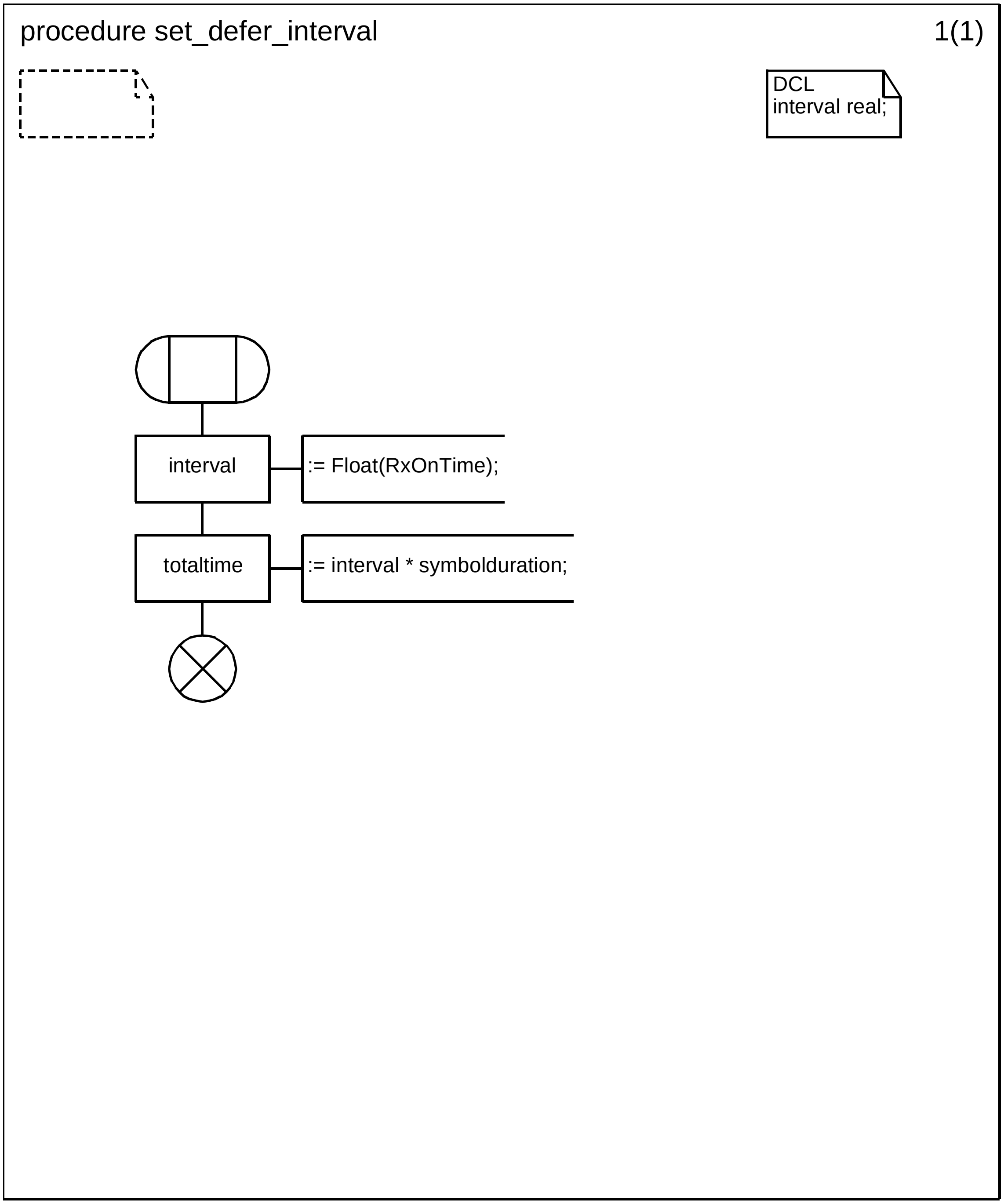
D.3.1.154.130 Procedure set_beacon_track_period

.



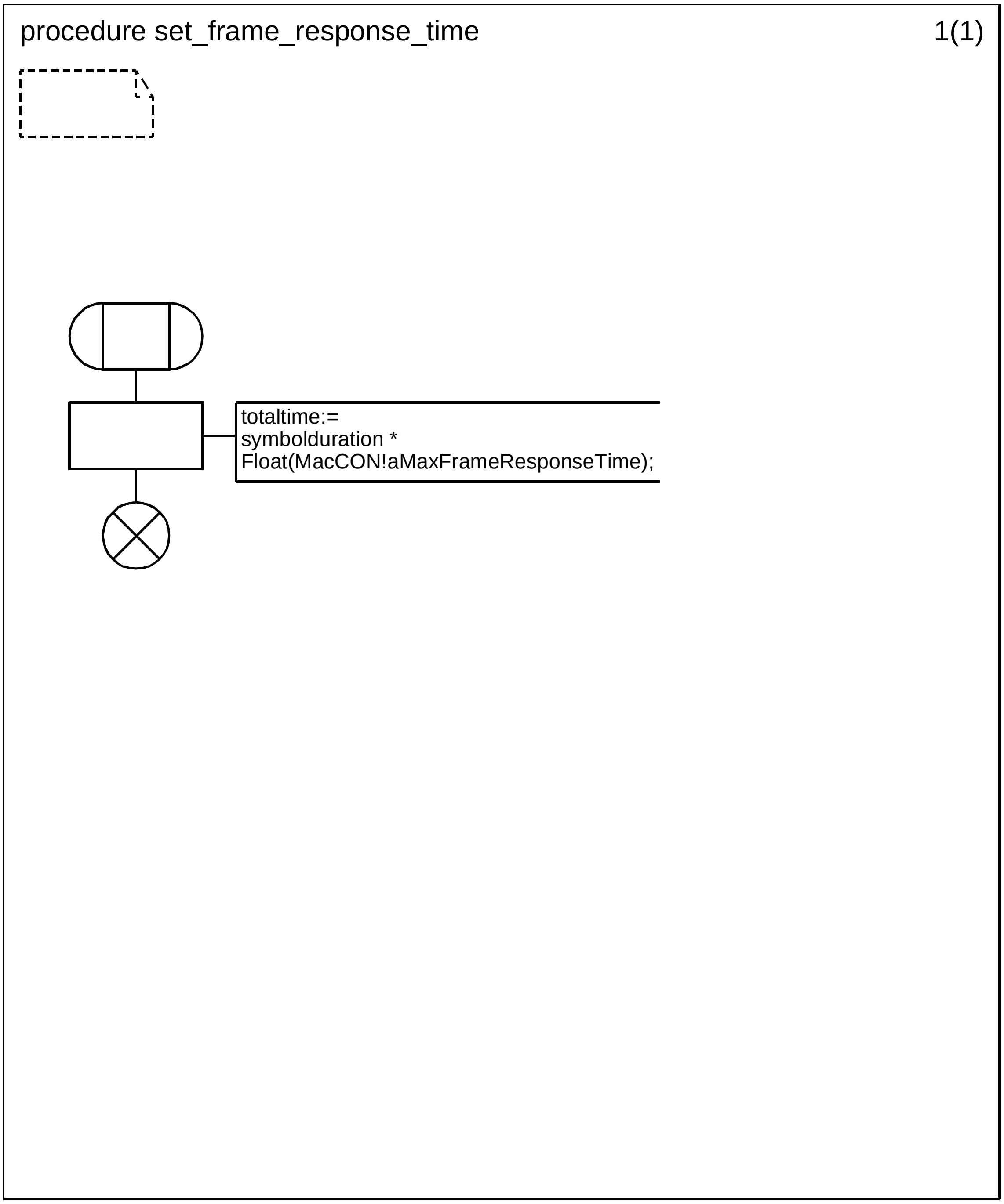
D.3.1.154.131 Procedure set_defer_interval

.



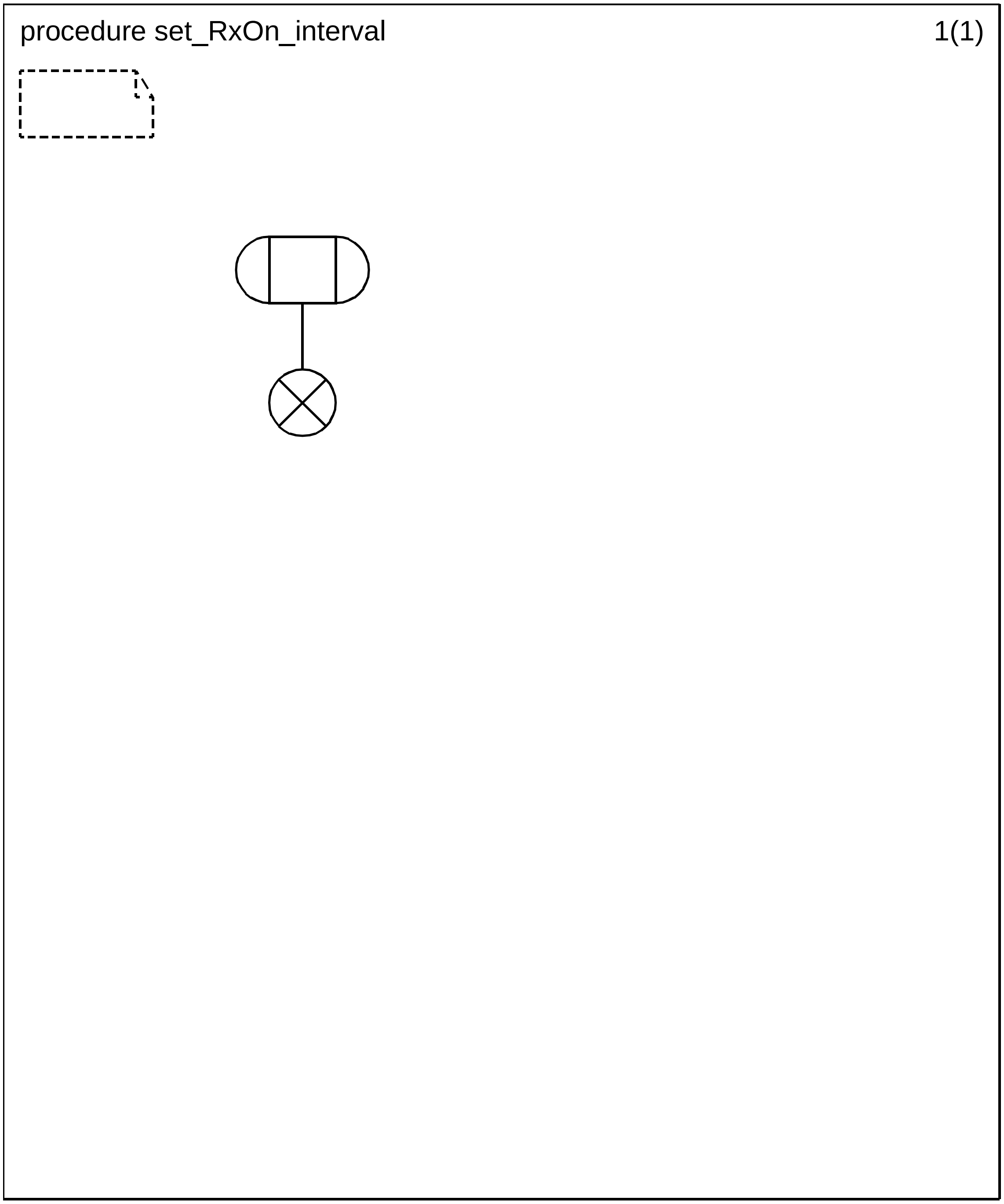
D.3.1.154.132 Procedure set_frame_response_time

.


procedure set_frame_response_time
1(1)
totaltime:=
symbolduration *
Float(MacCON!aMaxFrameResponseTime);

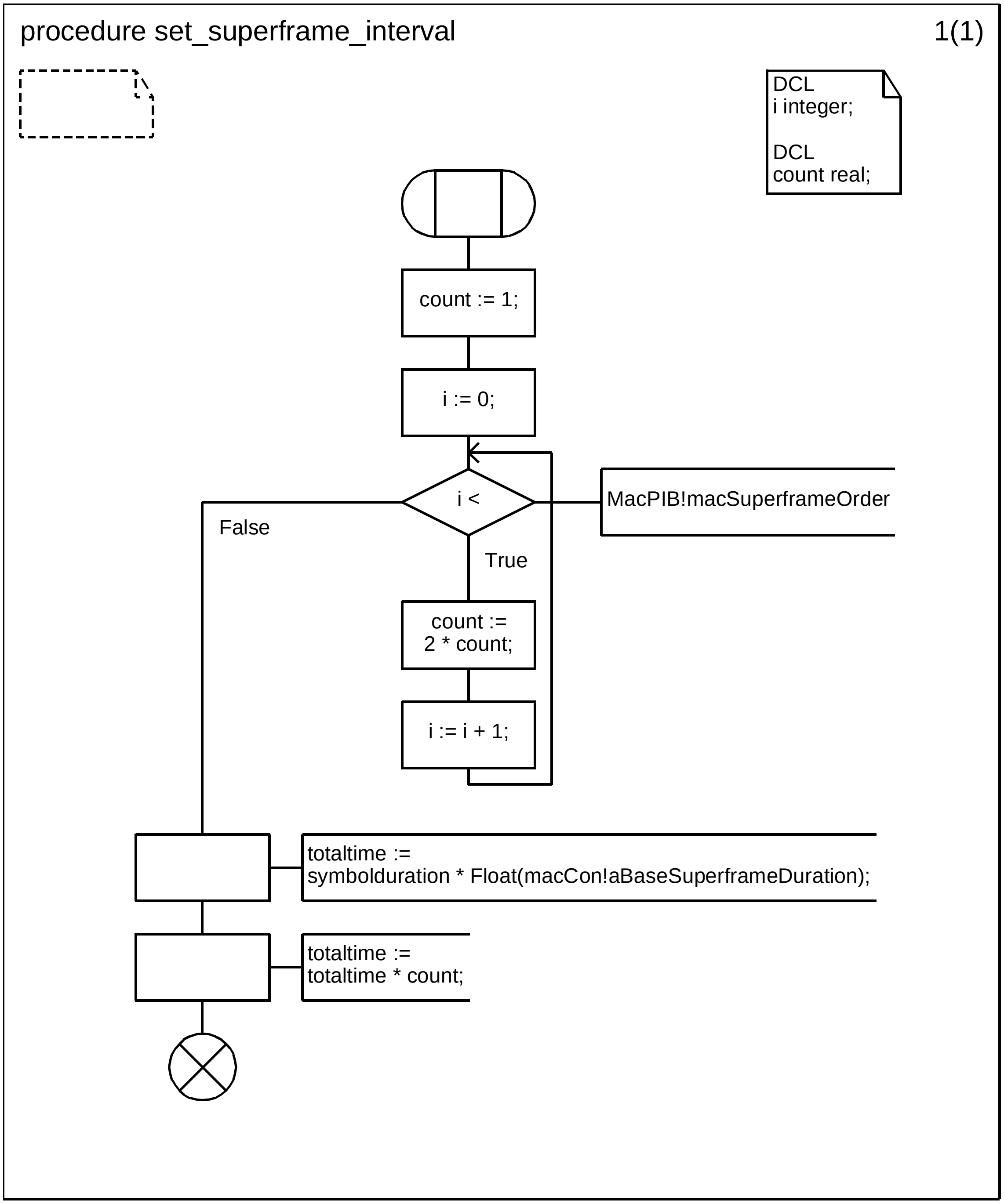
D.3.1.154.133 Procedure set_RxOn_interval

.



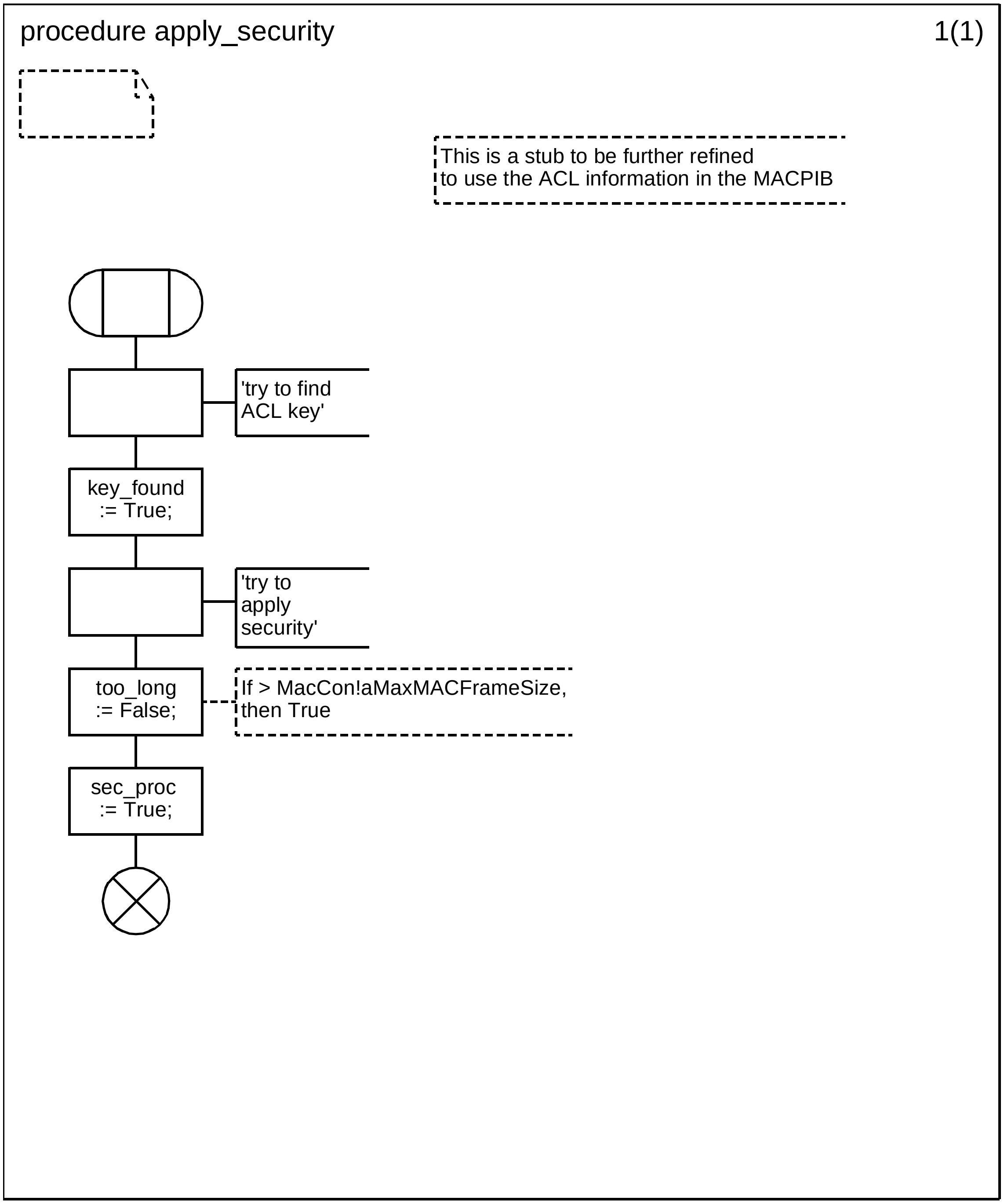
D.3.1.154.134 Procedure set_superframe_interval

.



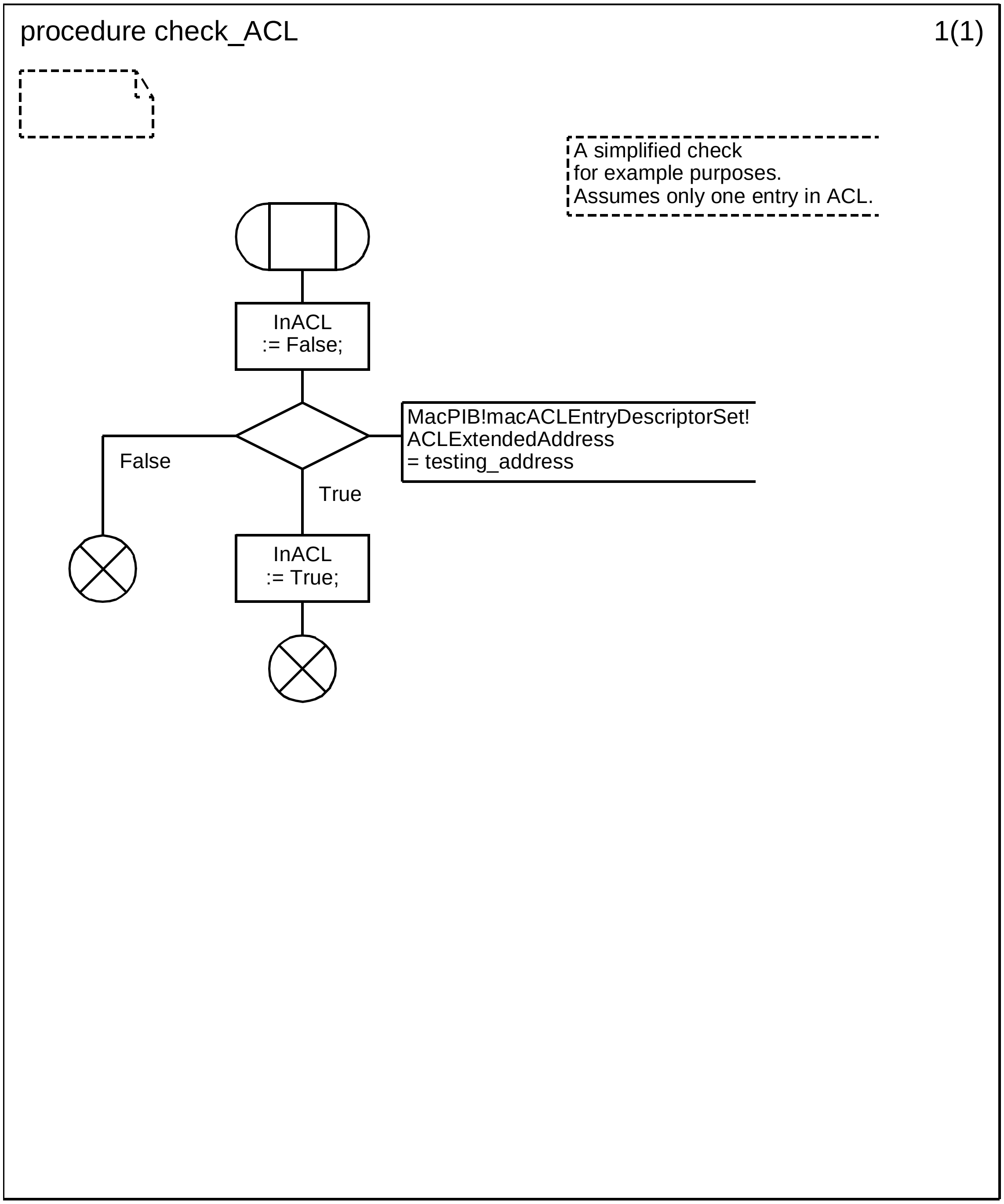
D.3.1.154.134.1 Procedure apply_security (1)

.

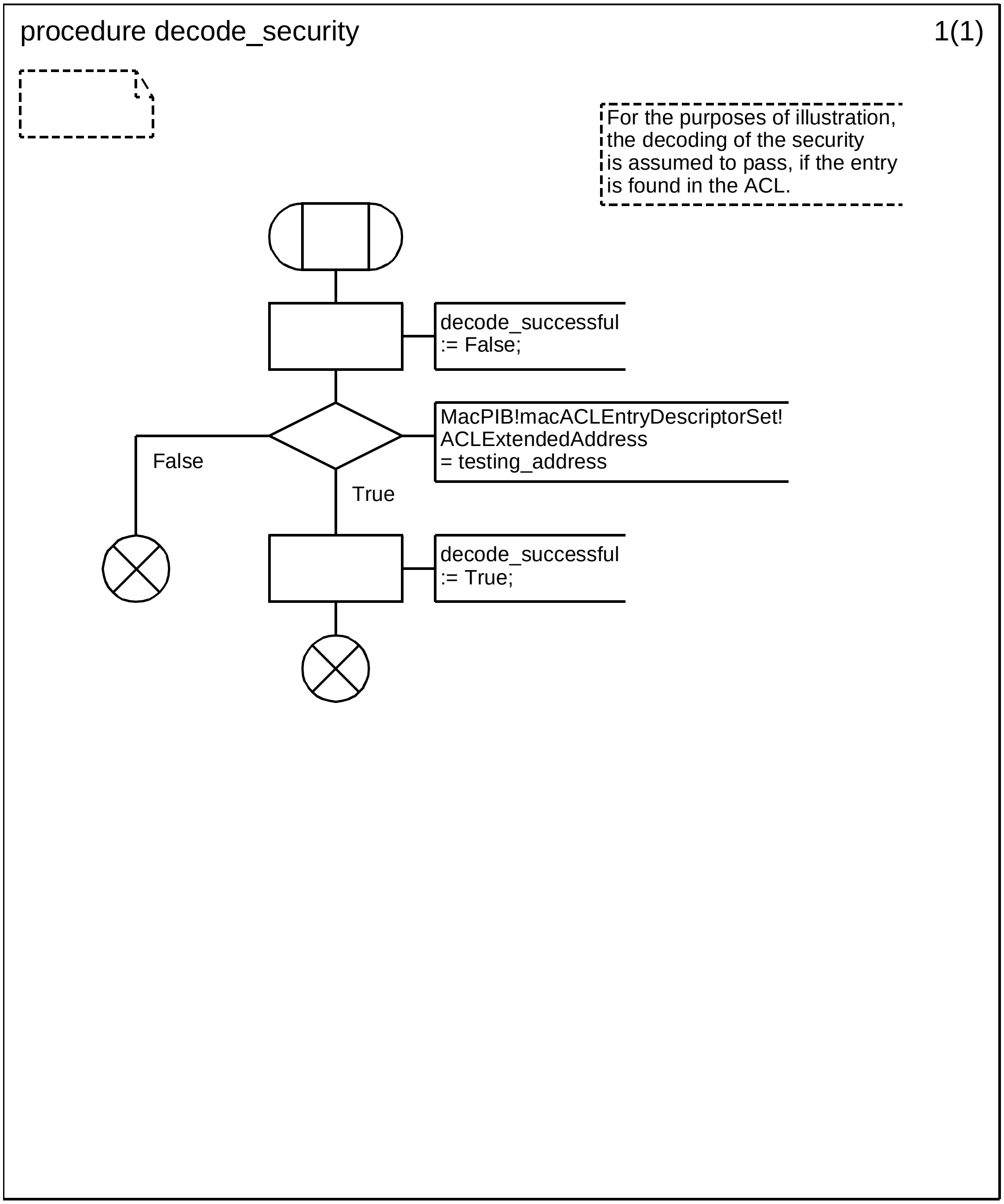


D.3.1.154.134.2 Procedure check_ACL

.

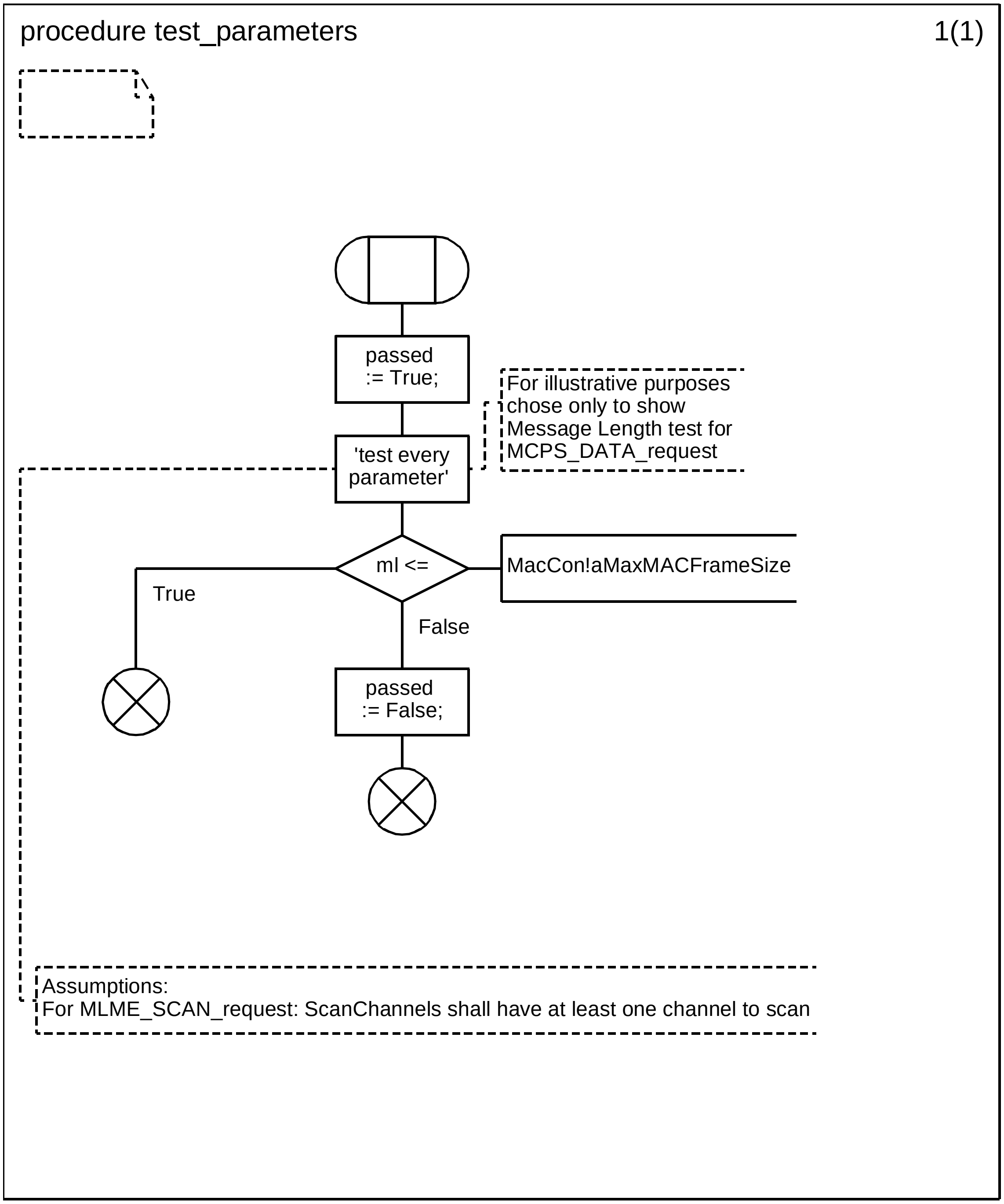


D.3.1.154.134.3 Procedure decode_security


procedure decode_security
1(1)
For the purposes of illustration, the decoding of the security is assumed to pass, if the entry is found in the ACL.
decode_successful := False;
MacPIB!macACLEntryDescriptorSet! ACLExtendedAddress = testing_address
False
True
decode_successful := True;

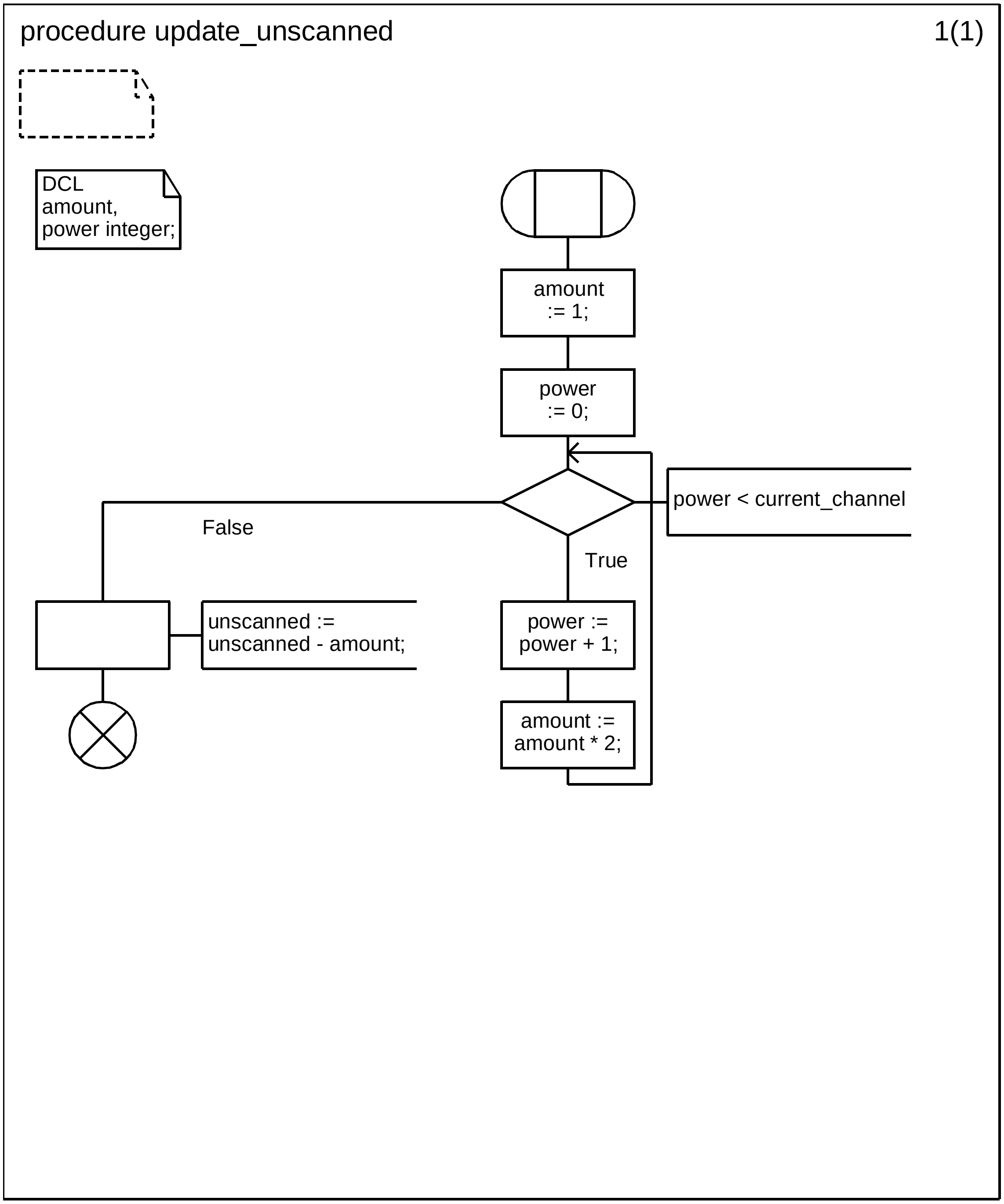
D.3.1.154.134.4 Procedure test_parameters (1)

.



D.3.1.154.134.5 Procedure update_unscanned

.

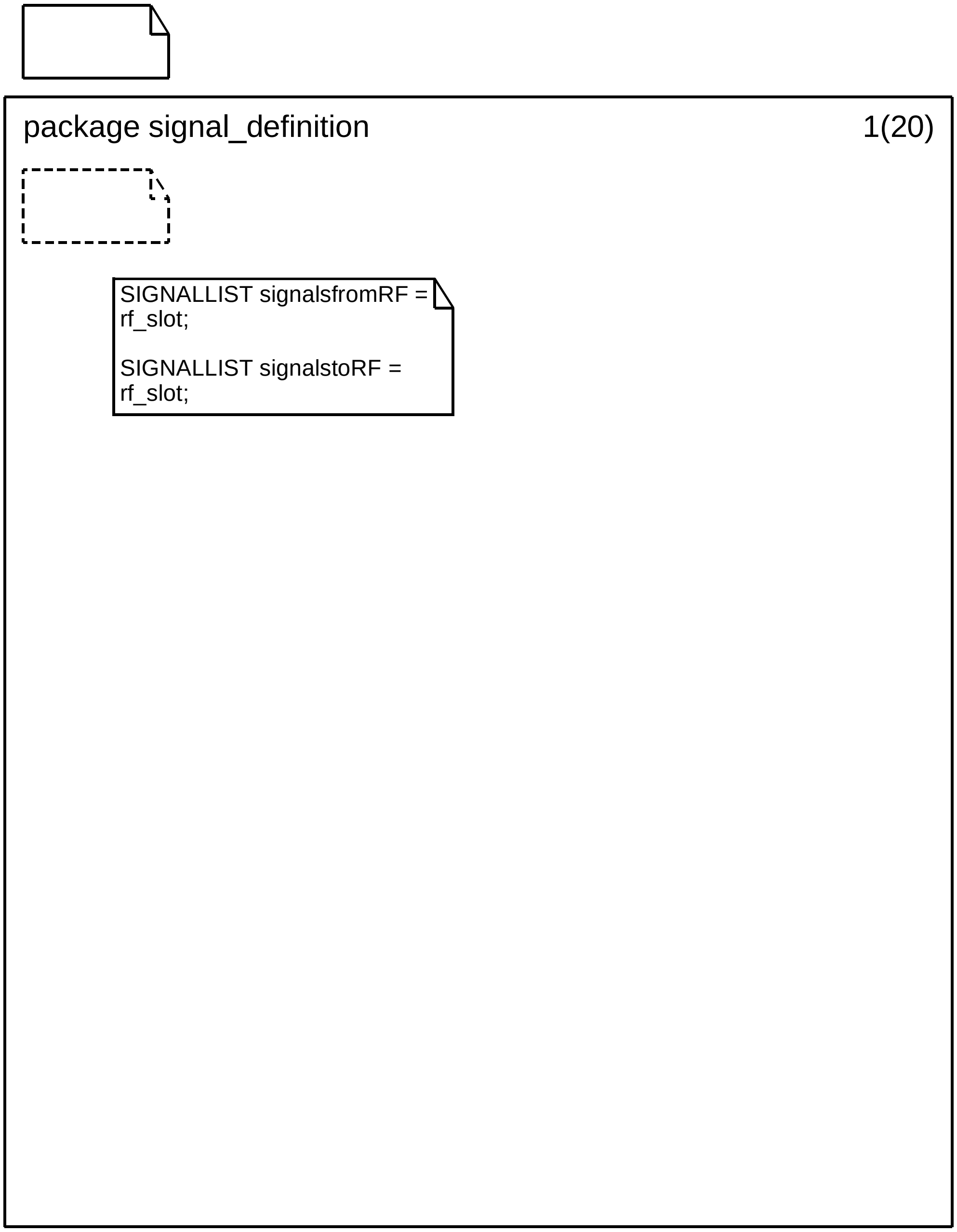


D.4 Signal definition package

This package defines the various signals, signallists, and frame types (i.e., in newtypes and syntypes in SDL) that are used by the other packages and system model

D.4.1 Signal definition package (1)

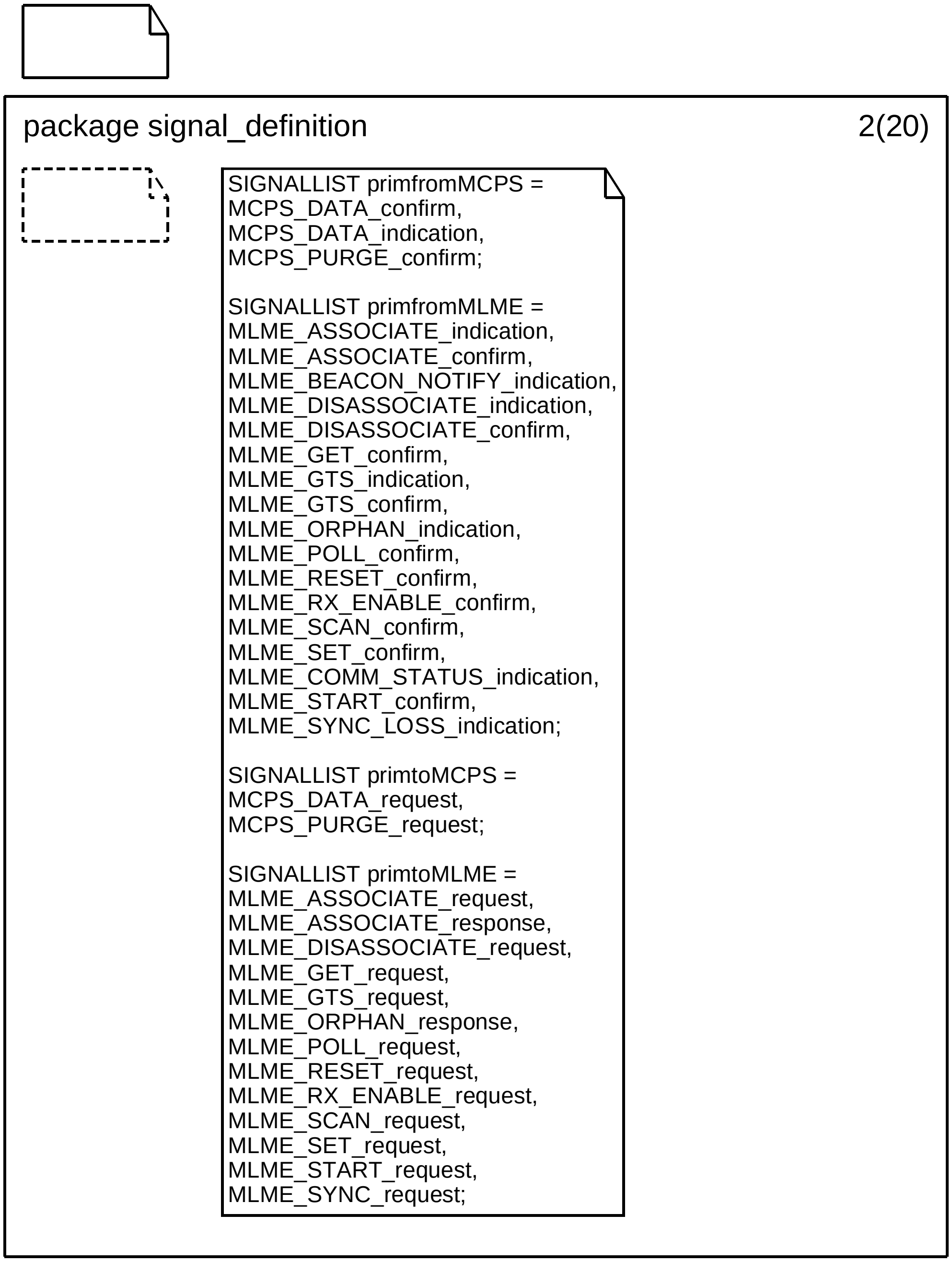
.



```
SIGNALLIST signalsfromRF =
rf_slot;

SIGNALLIST signalstoRF =
rf_slot;
```

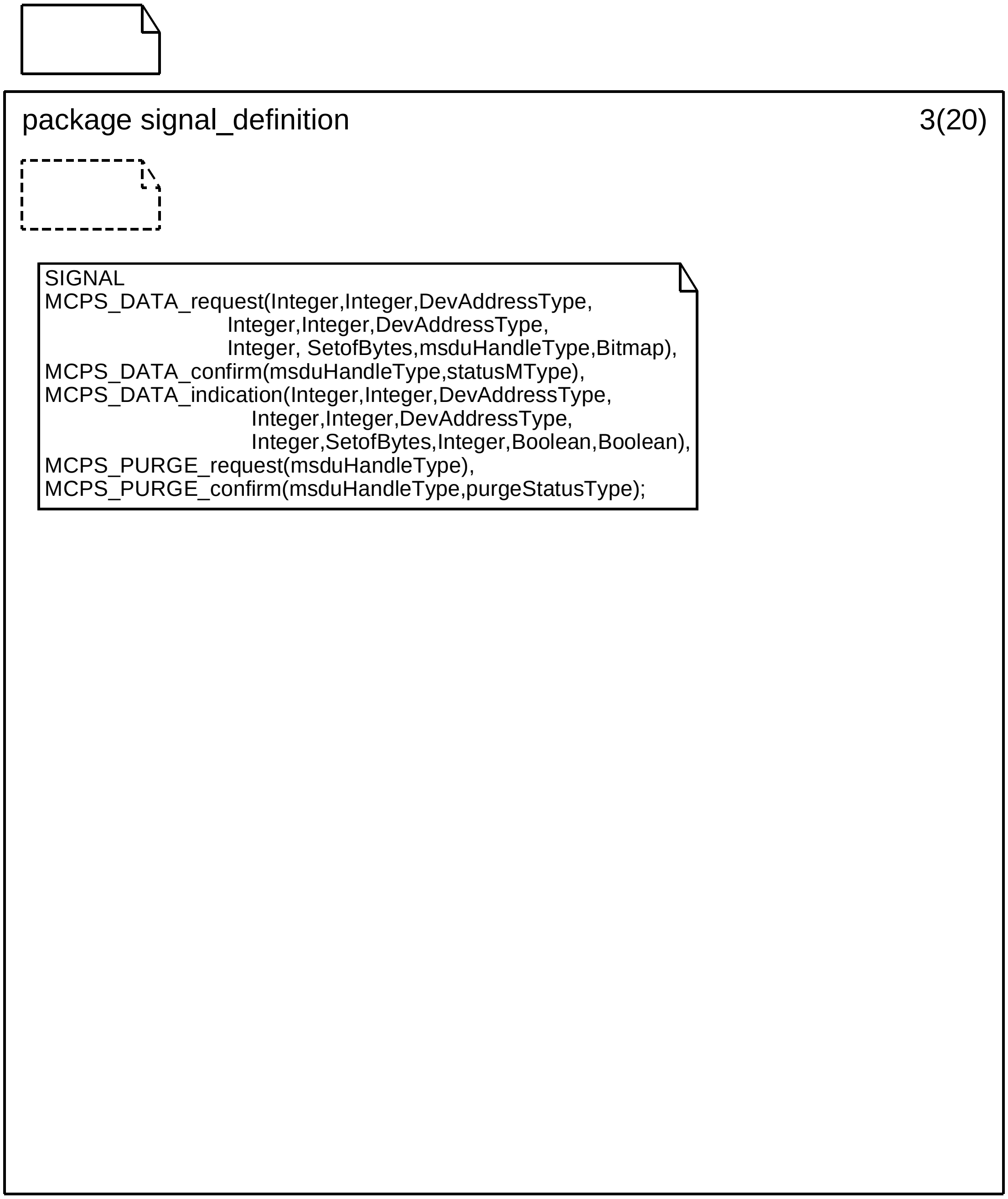
D.4.2 Signal definition package (2)

.



```
SIGNALLIST primfromMCPS =
MCPS_DATA_confirm,
MCPS_DATA_indication,
MCPS_PURGE_confirm;

SIGNALLIST primfromMLME =
MLME_ASSOCIATE_indication,
MLME_ASSOCIATE_confirm,
MLME_BEACON_NOTIFY_indication,
MLME_DISASSOCIATE_indication,
MLME_DISASSOCIATE_confirm,
MLME_GET_confirm,
MLME_GTS_indication,
MLME_GTS_confirm,
MLME_ORPHAN_indication,
MLME_POLL_confirm,
MLME_RESET_confirm,
MLME_RX_ENABLE_confirm,
MLME_SCAN_confirm,
MLME_SET_confirm,
MLME_COMM_STATUS_indication,
MLME_START_confirm,
MLME_SYNC_LOSS_indication;

SIGNALLIST primtoMCPS =
MCPS_DATA_request,
MCPS_PURGE_request;

SIGNALLIST primtoMLME =
MLME_ASSOCIATE_request,
MLME_ASSOCIATE_response,
MLME_DISASSOCIATE_request,
MLME_GET_request,
MLME_GTS_request,
MLME_ORPHAN_response,
MLME_POLL_request,
MLME_RESET_request,
MLME_RX_ENABLE_request,
MLME_SCAN_request,
MLME_SET_request,
MLME_START_request,
MLME_SYNC_request;
```

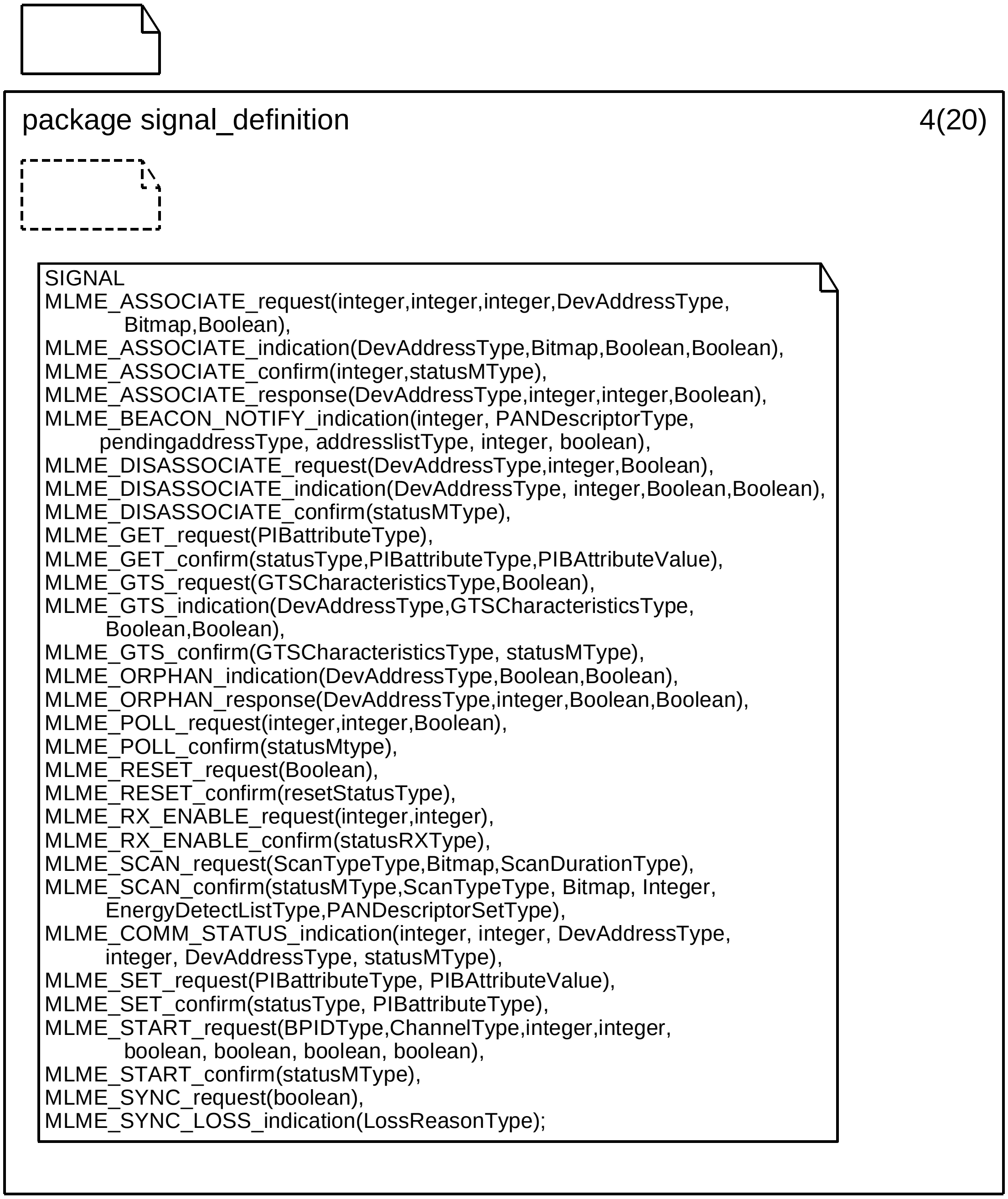
D.4.3 Signal definition package (3)



```
SIGNAL
MCPS_DATA_request(Integer,Integer,DevAddressType,
                        Integer,Integer,DevAddressType,
                        Integer, SetofBytes,msduHandleType,Bitmap),
MCPS_DATA_confirm(msduHandleType,statusMType),
MCPS_DATA_indication(Integer,Integer,DevAddressType,
                           Integer,Integer,DevAddressType,
                           Integer,SetofBytes,Integer,Boolean,Boolean),
MCPS_PURGE_request(msduHandleType),
MCPS_PURGE_confirm(msduHandleType,purgeStatusType);
```

D.4.4 Signal definition package (4)

.

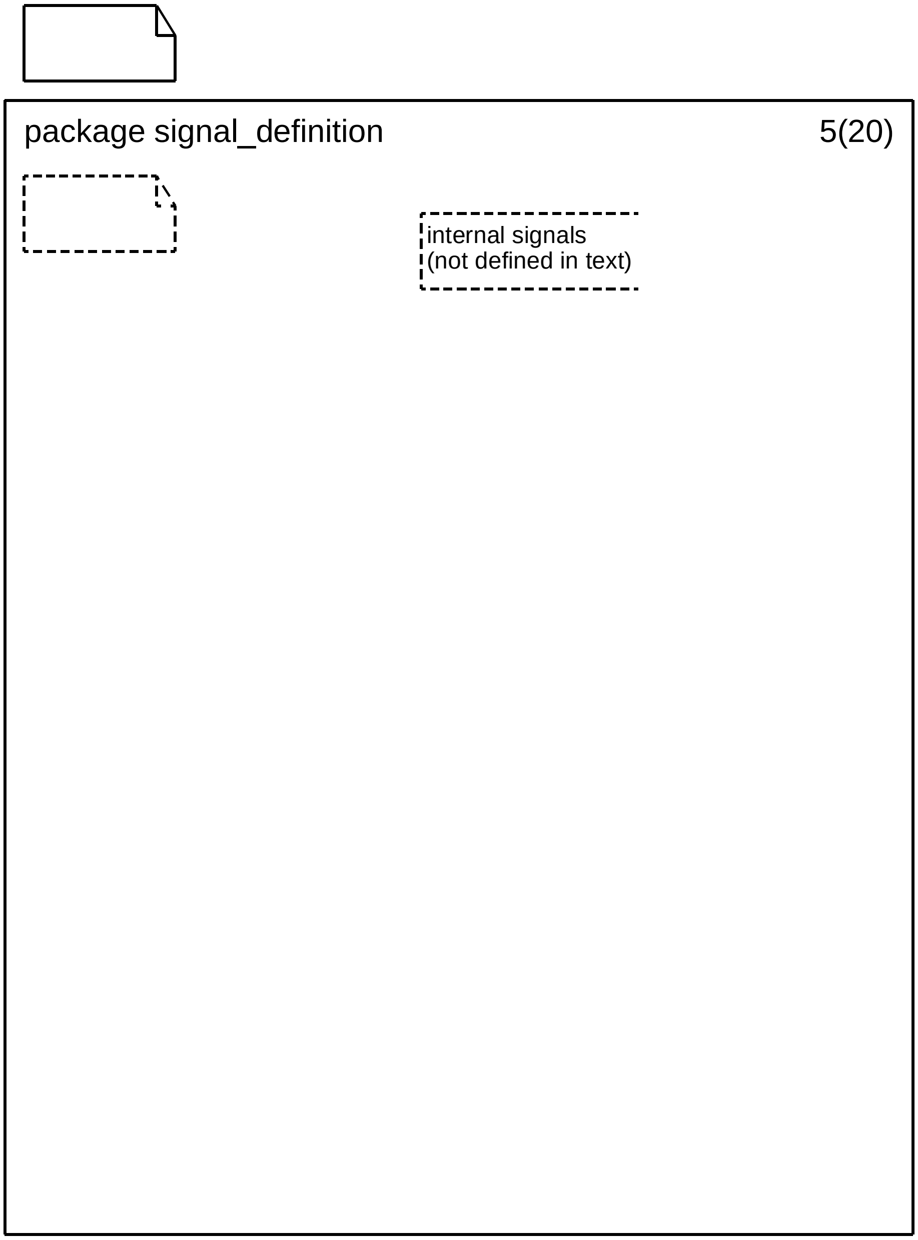


package signal_definition 4(20)

```
SIGNAL
MLME_ASSOCIATE_request(integer,integer,integer,DevAddressType,
          Bitmap,Boolean),
MLME_ASSOCIATE_indication(DevAddressType,Bitmap,Boolean,Boolean),
MLME_ASSOCIATE_confirm(integer,statusMType),
MLME_ASSOCIATE_response(DevAddressType,integer,integer,Boolean),
MLME_BEACON_NOTIFY_indication(integer, PANDescriptorType,
       pendingaddressType, addresslistType, integer, boolean),
MLME_DISASSOCIATE_request(DevAddressType,integer,Boolean),
MLME_DISASSOCIATE_indication(DevAddressType, integer,Boolean,Boolean),
MLME_DISASSOCIATE_confirm(statusMType),
MLME_GET_request(PIBattributeType),
MLME_GET_confirm(statusType,PIBattributeType,PIBAttributeValue),
MLME_GTS_request(GTSCharacteristicsType,Boolean),
MLME_GTS_indication(DevAddressType,GTSCharacteristicsType,
        Boolean,Boolean),
MLME_GTS_confirm(GTSCharacteristicsType, statusMType),
MLME_ORPHAN_indication(DevAddressType,Boolean,Boolean),
MLME_ORPHAN_response(DevAddressType,integer,Boolean,Boolean),
MLME_POLL_request(integer,integer,Boolean),
MLME_POLL_confirm(statusMtype),
MLME_RESET_request(Boolean),
MLME_RESET_confirm(resetStatusType),
MLME_RX_ENABLE_request(integer,integer),
MLME_RX_ENABLE_confirm(statusRXType),
MLME_SCAN_request(ScanTypeType,Bitmap,ScanDurationType),
MLME_SCAN_confirm(statusMType,ScanTypeType, Bitmap, Integer,
        EnergyDetectListType,PANDescriptorSetType),
MLME_COMM_STATUS_indication(integer, integer, DevAddressType,
        integer, DevAddressType, statusMType),
MLME_SET_request(PIBattributeType, PIBAttributeValue),
MLME_SET_confirm(statusType, PIBattributeType),
MLME_START_request(BPIDType,ChannelType,integer,integer,
          boolean, boolean, boolean, boolean),
MLME_START_confirm(statusMType),
MLME_SYNC_request(boolean),
MLME_SYNC_LOSS_indication(LossReasonType);
```

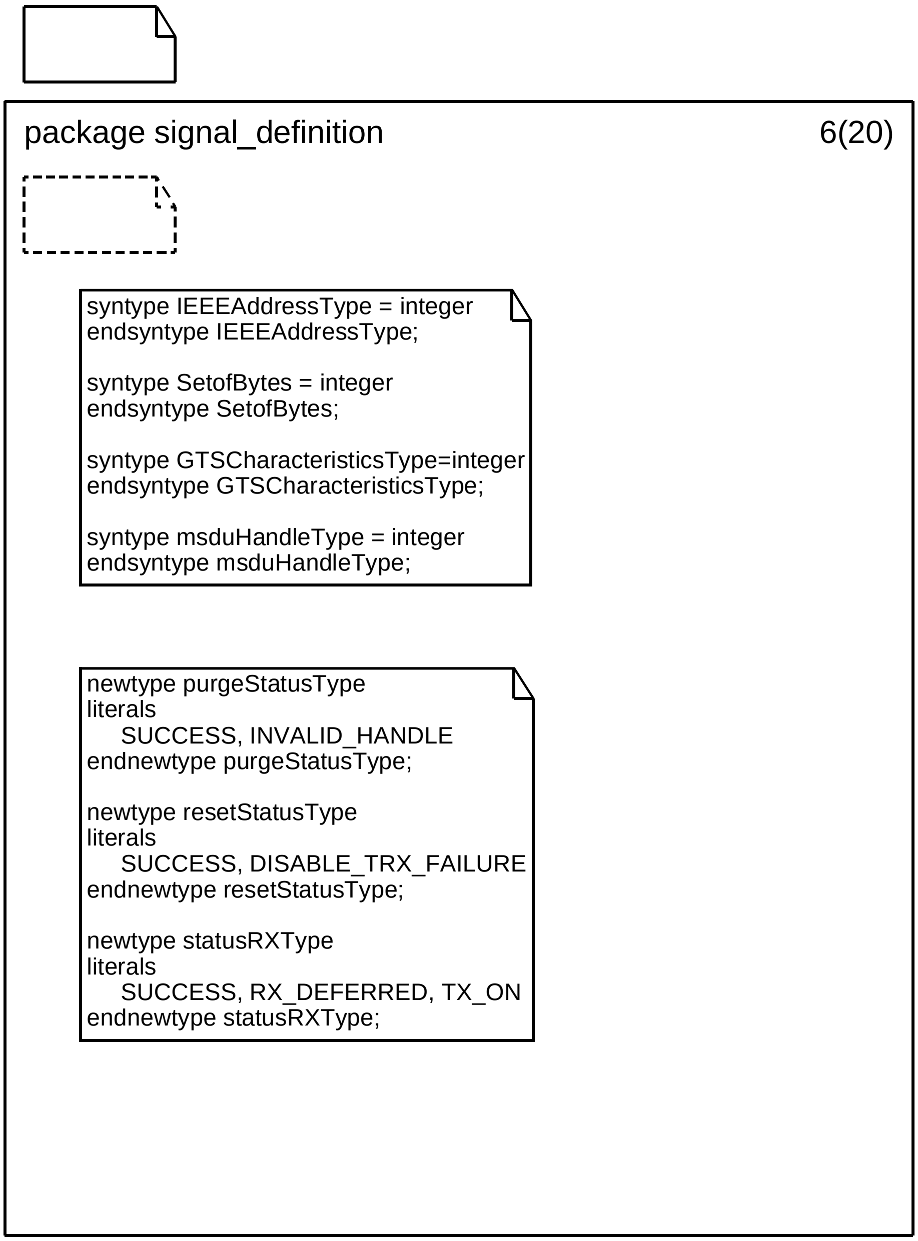
D.4.5 Signal definition package (5)

.



D.4.6 Signal definition package (6)

.

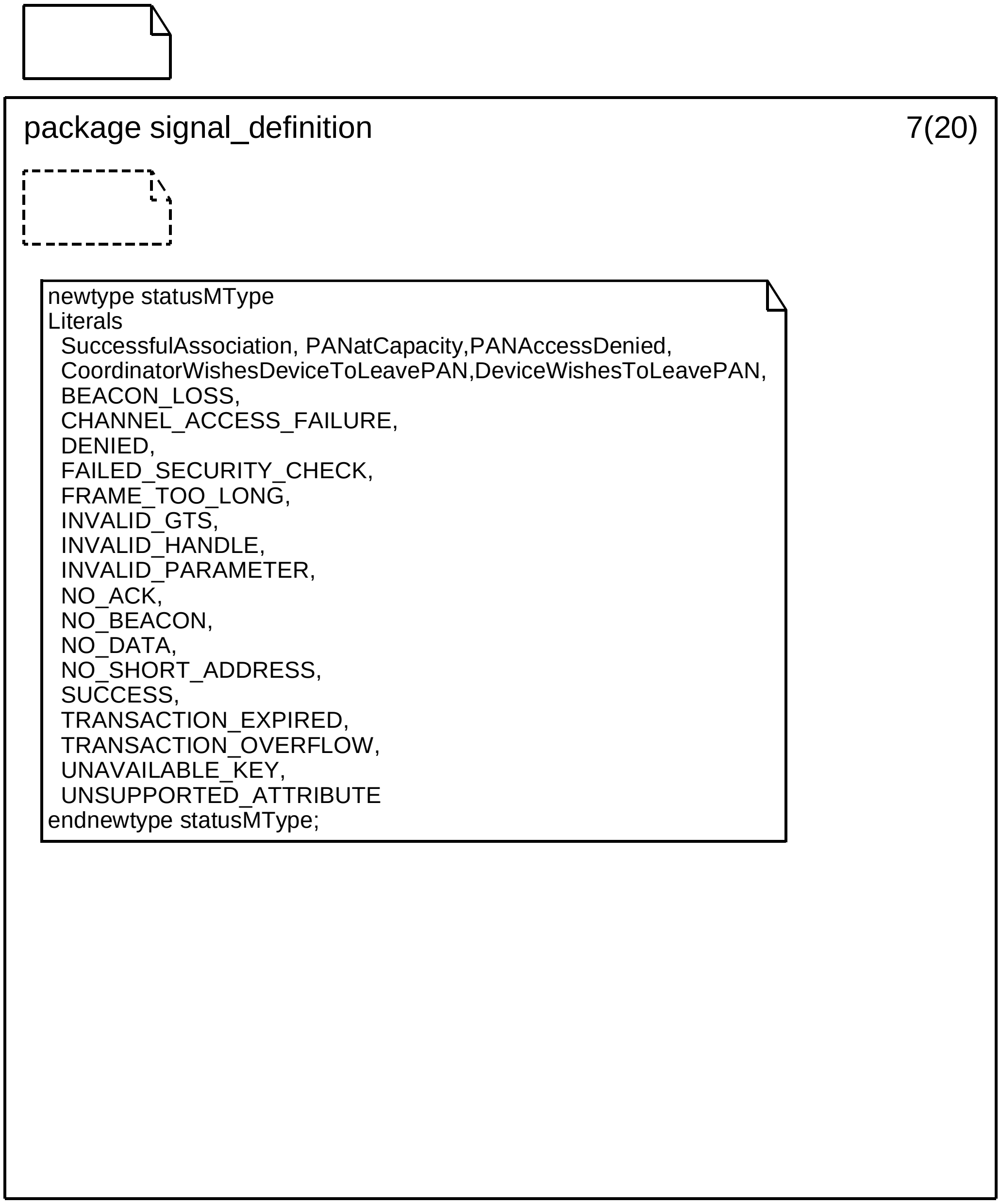


```
syntype IEEEAddressType = integer
endsyntype IEEEAddressType;

syntype SetofBytes = integer
endsyntype SetofBytes;

syntype GTSCharacteristicsType=integer
endsyntype GTSCharacteristicsType;

syntype msduHandleType = integer
endsyntype msduHandleType;
```

```
newtype purgeStatusType
literals
    SUCCESS, INVALID_HANDLE
endnewtype purgeStatusType;

newtype resetStatusType
literals
    SUCCESS, DISABLE_TRX_FAILURE
endnewtype resetStatusType;

newtype statusRXType
literals
    SUCCESS, RX_DEFERRED, TX_ON
endnewtype statusRXType;
```

D.4.7 Signal definition package (7)

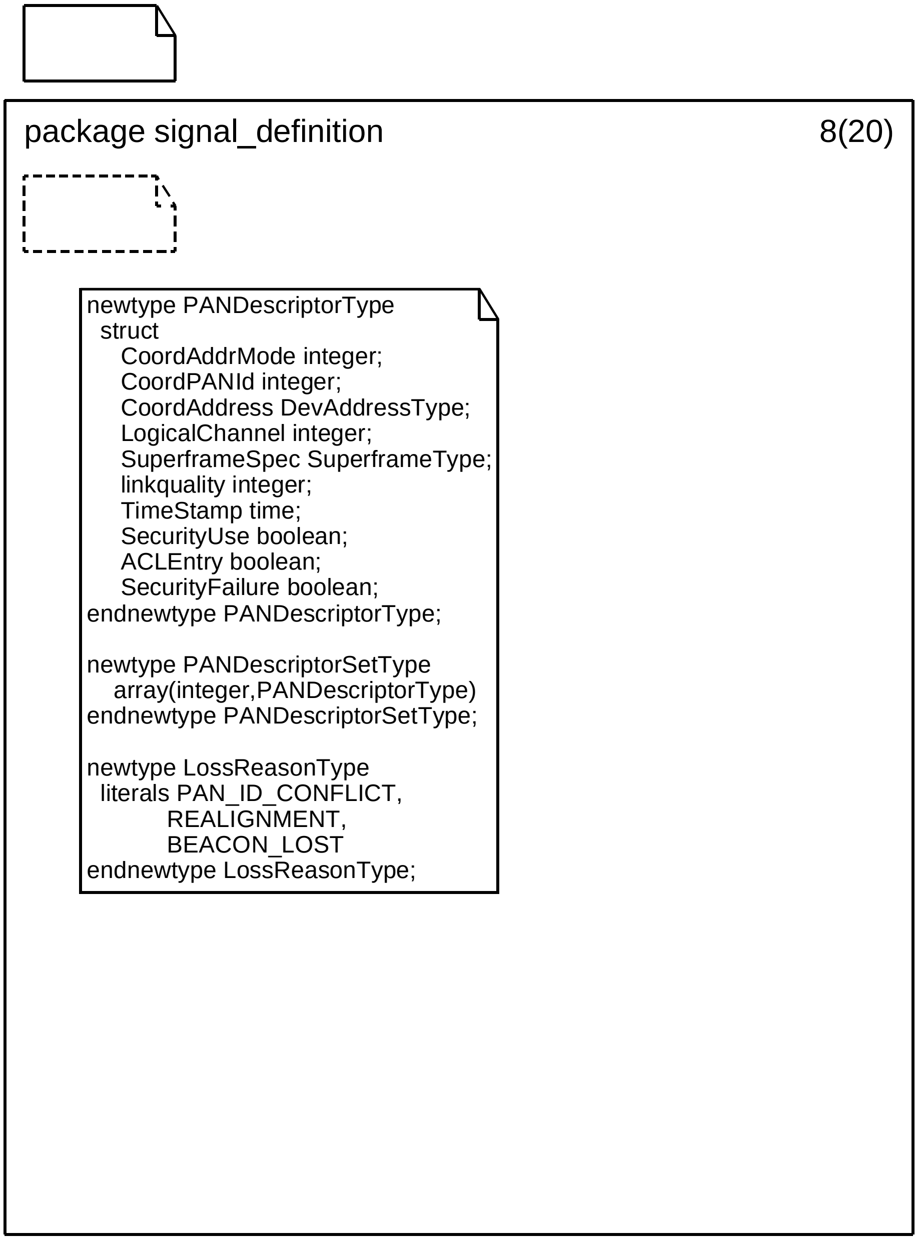
.



```
newtype statusMType
Literals
  SuccessfulAssociation, PANatCapacity,PANAccessDenied,
  CoordinatorWishesDeviceToLeavePAN,DeviceWishesToLeavePAN,
  BEACON_LOSS,
  CHANNEL_ACCESS_FAILURE,
  DENIED,
  FAILED_SECURITY_CHECK,
  FRAME_TOO_LONG,
  INVALID_GTS,
  INVALID_HANDLE,
  INVALID_PARAMETER,
  NO_ACK,
  NO_BEACON,
  NO_DATA,
  NO_SHORT_ADDRESS,
  SUCCESS,
  TRANSACTION_EXPIRED,
  TRANSACTION_OVERFLOW,
  UNAVAILABLE_KEY,
  UNSUPPORTED_ATTRIBUTE
endnewtype statusMType;
```

D.4.8 Signal definition package (8)

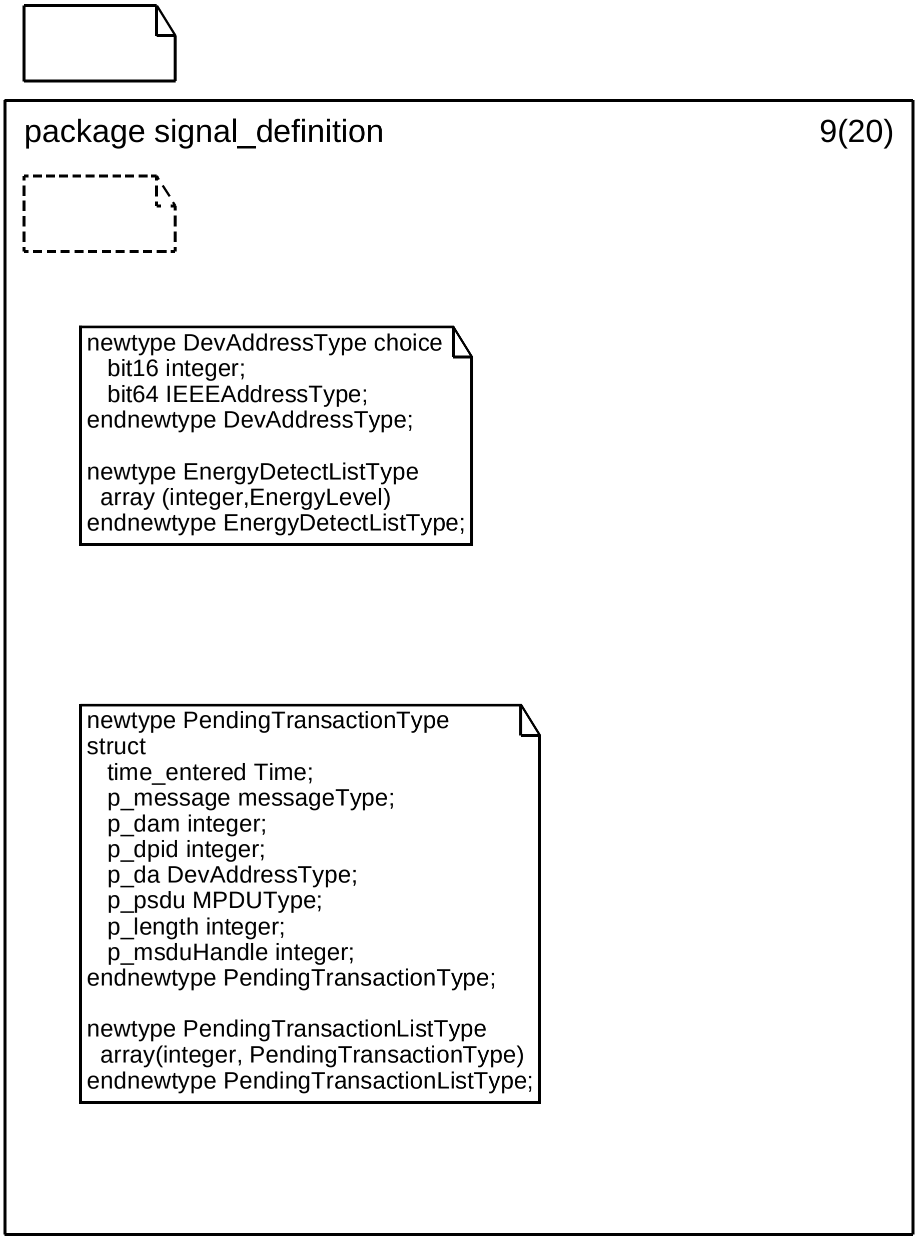
.



```
newtype PANDescriptorType
  struct
     CoordAddrMode integer;
     CoordPANId integer;
     CoordAddress DevAddressType;
     LogicalChannel integer;
     SuperframeSpec SuperframeType;
     linkquality integer;
     TimeStamp time;
     SecurityUse boolean;
     ACLEntry boolean;
     SecurityFailure boolean;
endnewtype PANDescriptorType;

newtype PANDescriptorSetType
    array(integer,PANDescriptorType)
endnewtype PANDescriptorSetType;

newtype LossReasonType
  literals PAN_ID_CONFLICT,
            REALIGNMENT,
            BEACON_LOST
endnewtype LossReasonType;
```

D.4.9 Signal definition package (9)

.

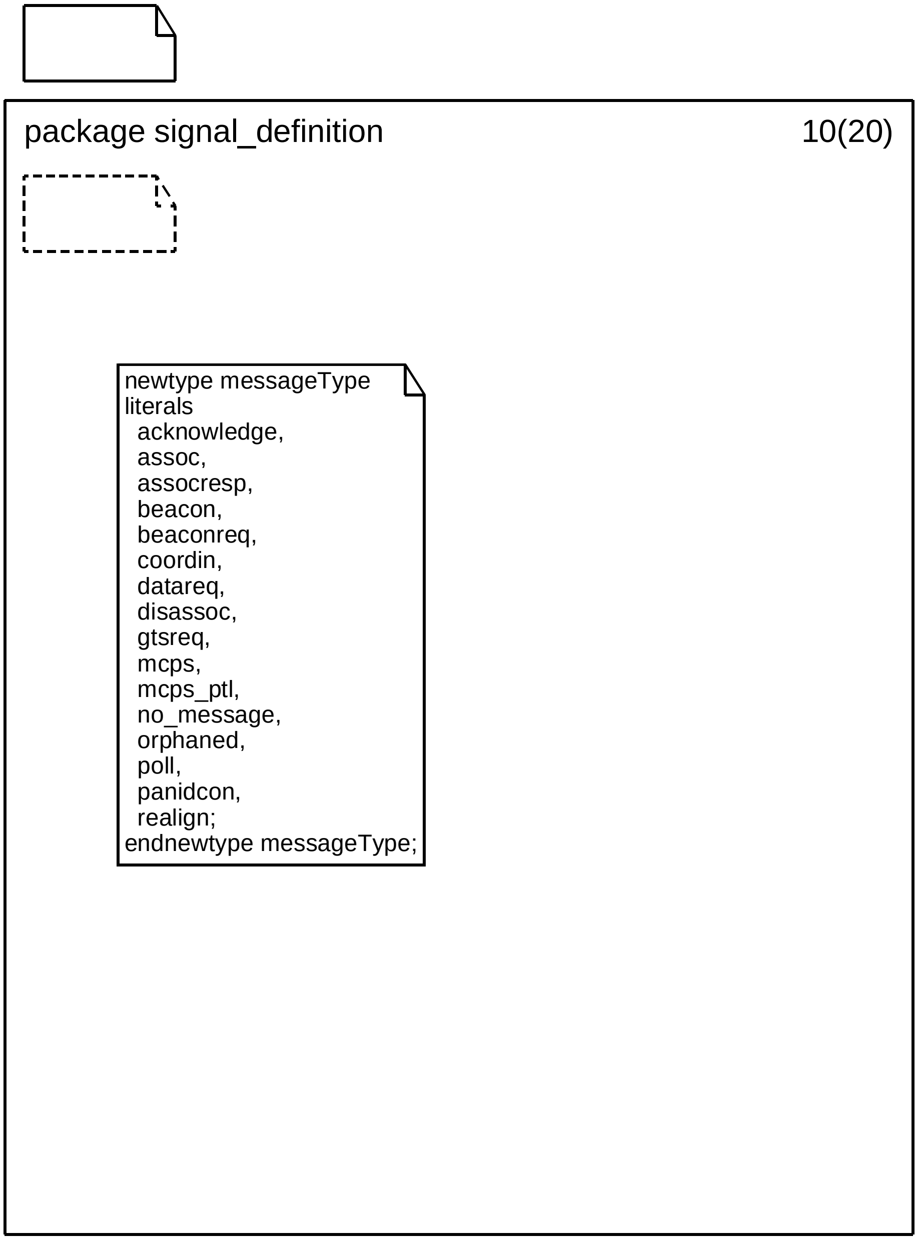


```
newtype DevAddressType choice
   bit16 integer;
   bit64 IEEEAddressType;
endnewtype DevAddressType;

newtype EnergyDetectListType
  array (integer,EnergyLevel)
endnewtype EnergyDetectListType;
```

```
newtype PendingTransactionType
struct
   time_entered Time;
   p_message messageType;
   p_dam integer;
   p_dpid integer;
   p_da DevAddressType;
   p_psdu MPDUType;
   p_length integer;
   p_msduHandle integer;
endnewtype PendingTransactionType;

newtype PendingTransactionListType
  array(integer, PendingTransactionType)
endnewtype PendingTransactionListType;
```

D.4.10 Signal definition package (10)

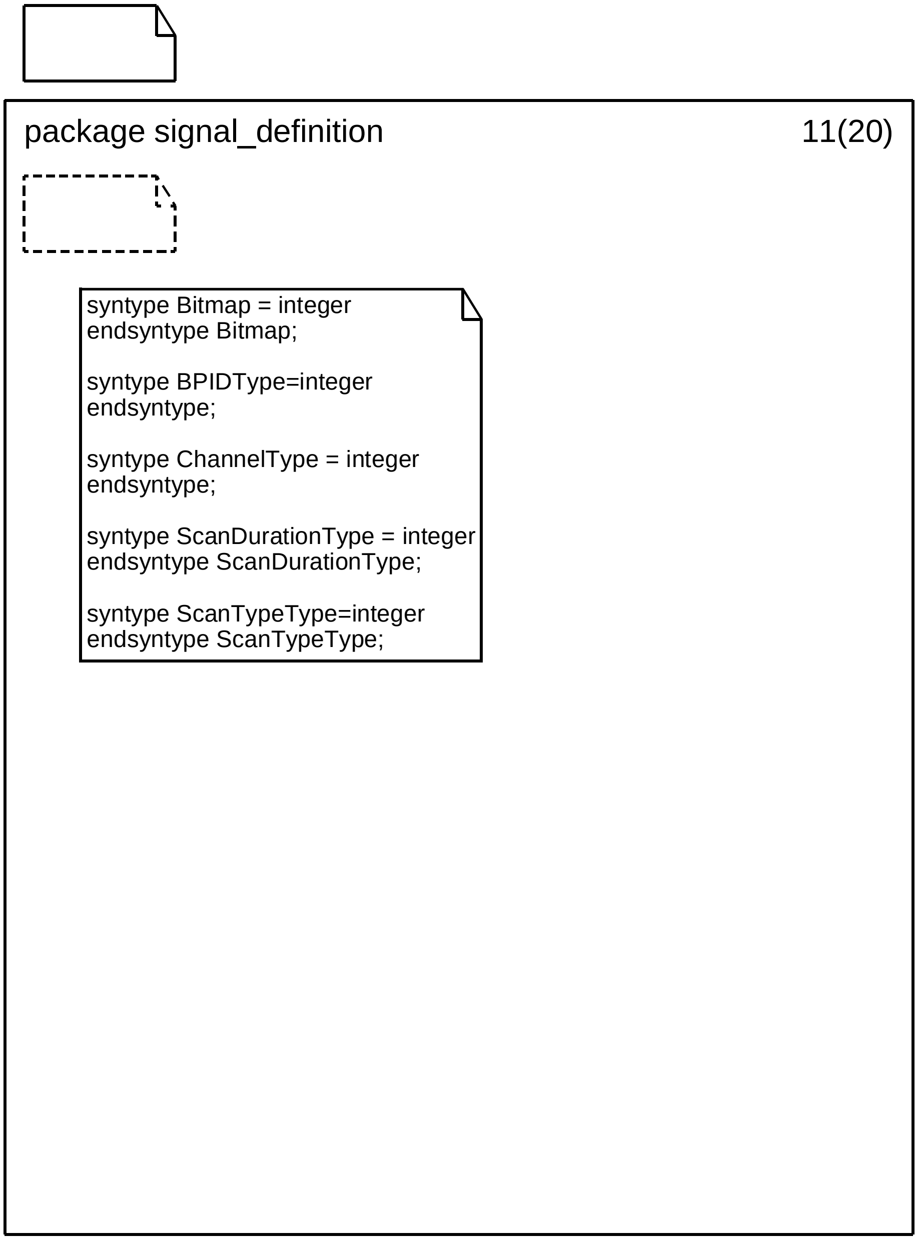
.



```
newtype messageType
literals
  acknowledge,
  assoc,
  assocresp,
  beacon,
  beaconreq,
  coordin,
  datareq,
  disassoc,
  gtsreq,
  mcps,
  mcps_ptl,
  no_message,
  orphaned,
  poll,
  panidcon,
  realign;
endnewtype messageType;
```

D.4.11 Signal definition package (11)

.

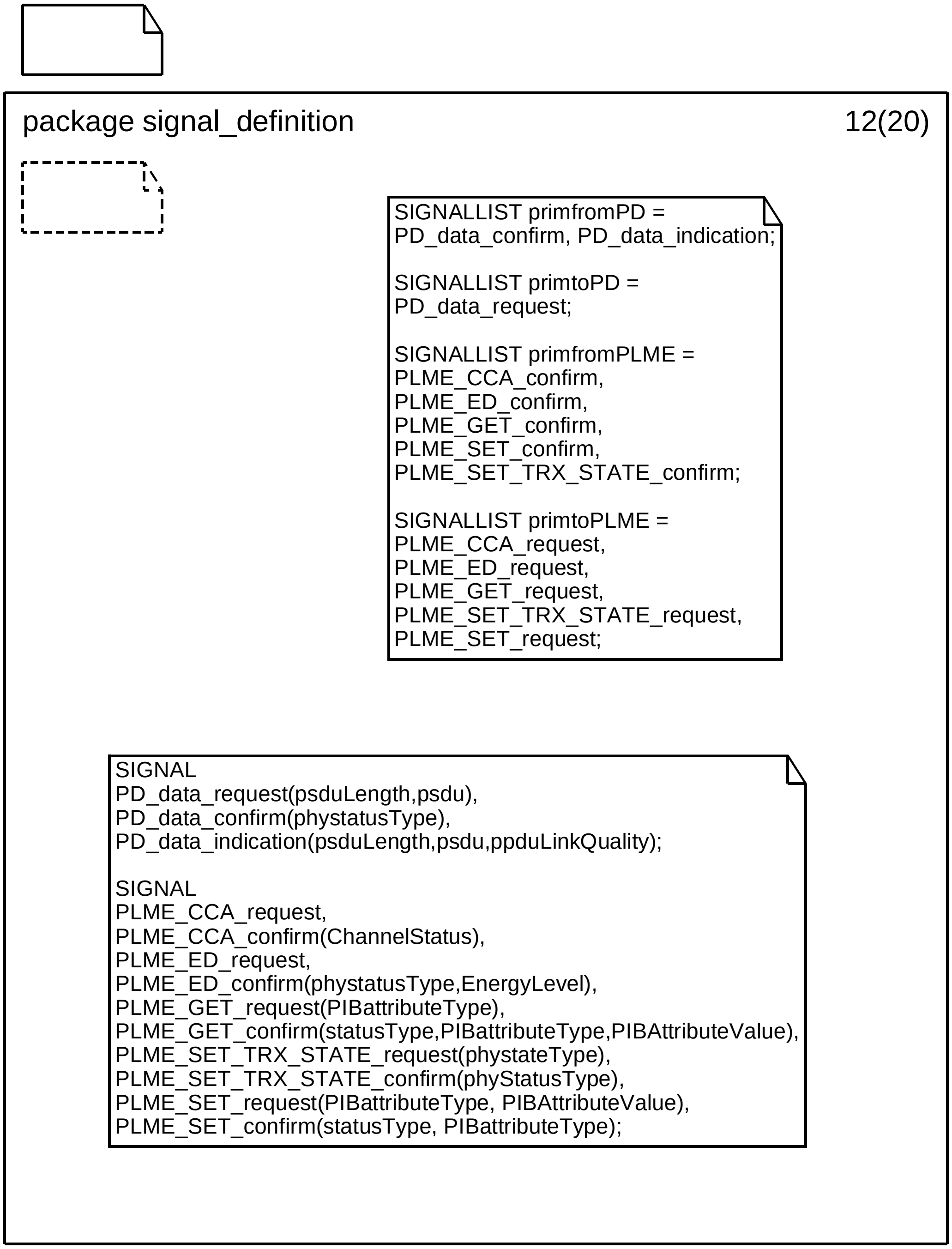


package signal_definition 11(20)

```
syntype Bitmap = integer
endsyntype Bitmap;

syntype BPIDType=integer
endsyntype;

syntype ChannelType = integer
endsyntype;

syntype ScanDurationType = integer
endsyntype ScanDurationType;

syntype ScanTypeType=integer
endsyntype ScanTypeType;
```

D.4.12 Signal definition package (12)

.

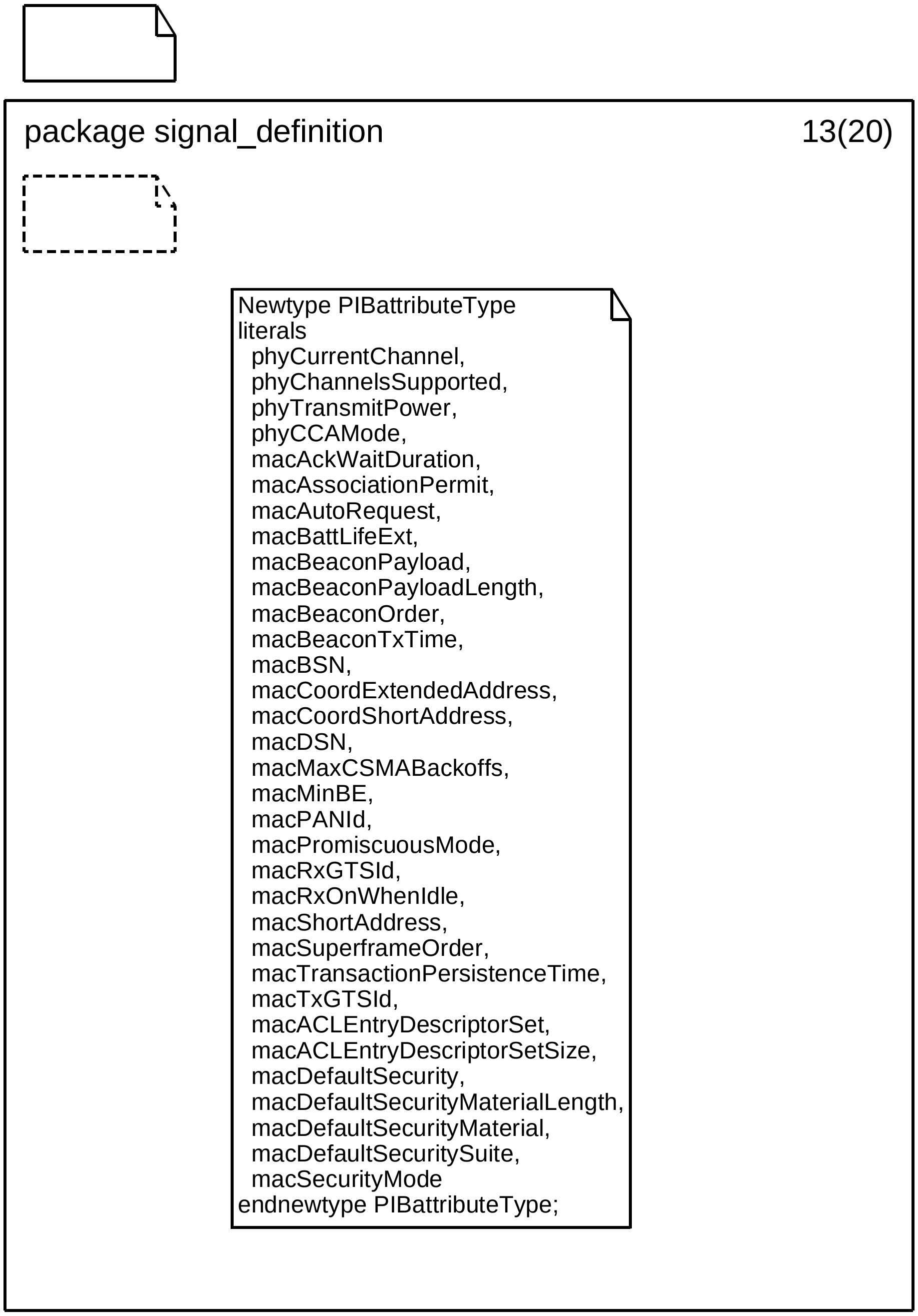


```
SIGNALLIST primfromPD =
PD_data_confirm, PD_data_indication;

SIGNALLIST primtoPD =
PD_data_request;

SIGNALLIST primfromPLME =
PLME_CCA_confirm,
PLME_ED_confirm,
PLME_GET_confirm,
PLME_SET_confirm,
PLME_SET_TRX_STATE_confirm;

SIGNALLIST primtoPLME =
PLME_CCA_request,
PLME_ED_request,
PLME_GET_request,
PLME_SET_TRX_STATE_request,
PLME_SET_request;
```

```
SIGNAL
PD_data_request(psduLength,psdu),
PD_data_confirm(phystatusType),
PD_data_indication(psduLength,psdu,ppduLinkQuality);

SIGNAL
PLME_CCA_request,
PLME_CCA_confirm(ChannelStatus),
PLME_ED_request,
PLME_ED_confirm(phystatusType,EnergyLevel),
PLME_GET_request(PIBattributeType),
PLME_GET_confirm(statusType,PIBattributeType,PIBAttributeValue),
PLME_SET_TRX_STATE_request(phystateType),
PLME_SET_TRX_STATE_confirm(phyStatusType),
PLME_SET_request(PIBattributeType, PIBAttributeValue),
PLME_SET_confirm(statusType, PIBattributeType);
```

D.4.13 Signal definition package (13)

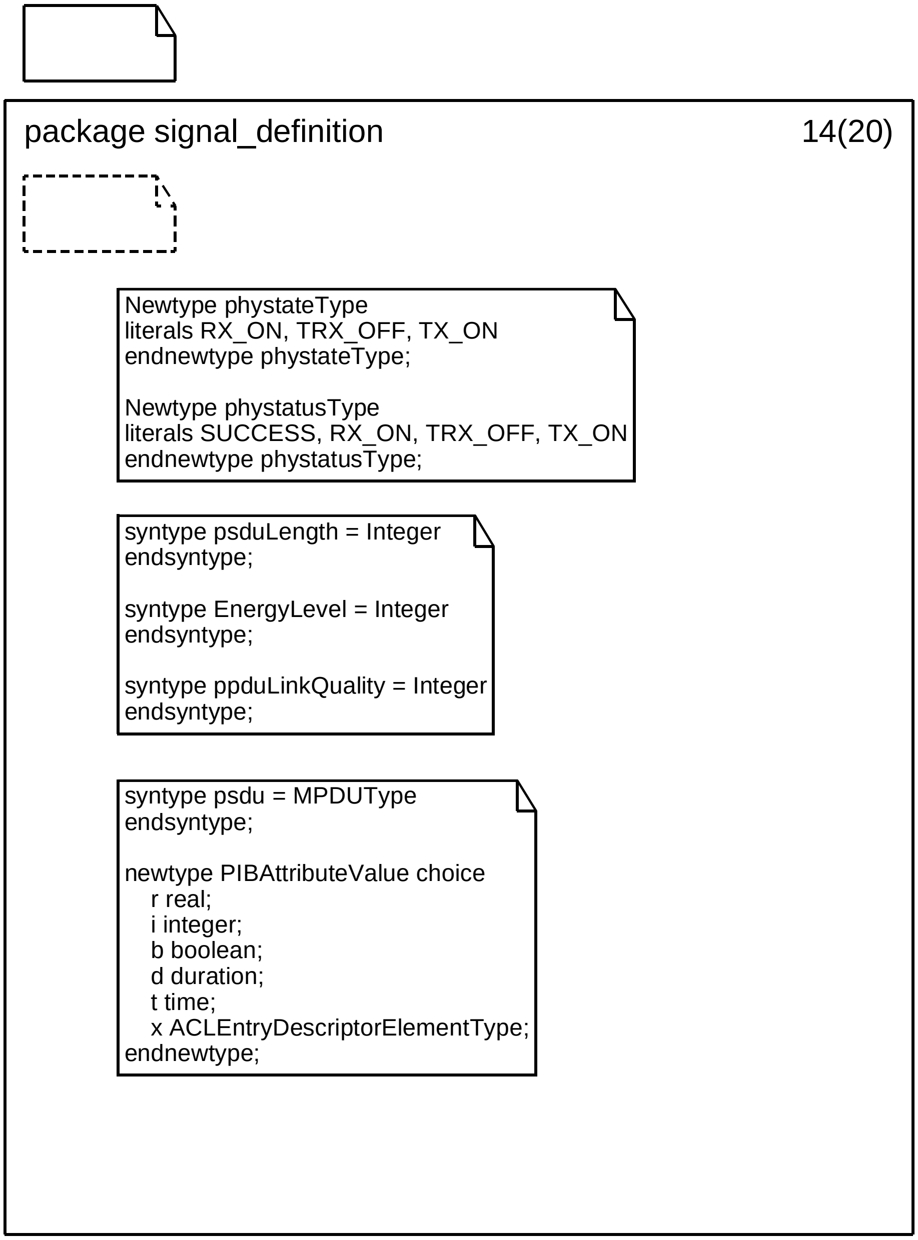
.



```
Newtype PIBattributeType
literals
  phyCurrentChannel,
  phyChannelsSupported,
  phyTransmitPower,
  phyCCAMode,
  macAckWaitDuration,
  macAssociationPermit,
  macAutoRequest,
  macBattLifeExt,
  macBeaconPayload,
  macBeaconPayloadLength,
  macBeaconOrder,
  macBeaconTxTime,
  macBSN,
  macCoordExtendedAddress,
  macCoordShortAddress,
  macDSN,
  macMaxCSMABackoffs,
  macMinBE,
  macPANId,
  macPromiscuousMode,
  macRxGTSId,
  macRxOnWhenIdle,
  macShortAddress,
  macSuperframeOrder,
  macTransactionPersistenceTime,
  macTxGTSId,
  macACLEntryDescriptorSet,
  macACLEntryDescriptorSetSize,
  macDefaultSecurity,
  macDefaultSecurityMaterialLength,
  macDefaultSecurityMaterial,
  macDefaultSecuritySuite,
  macSecurityMode
endnewtype PIBattributeType;
```

D.4.14 Signal definition package (14)

.



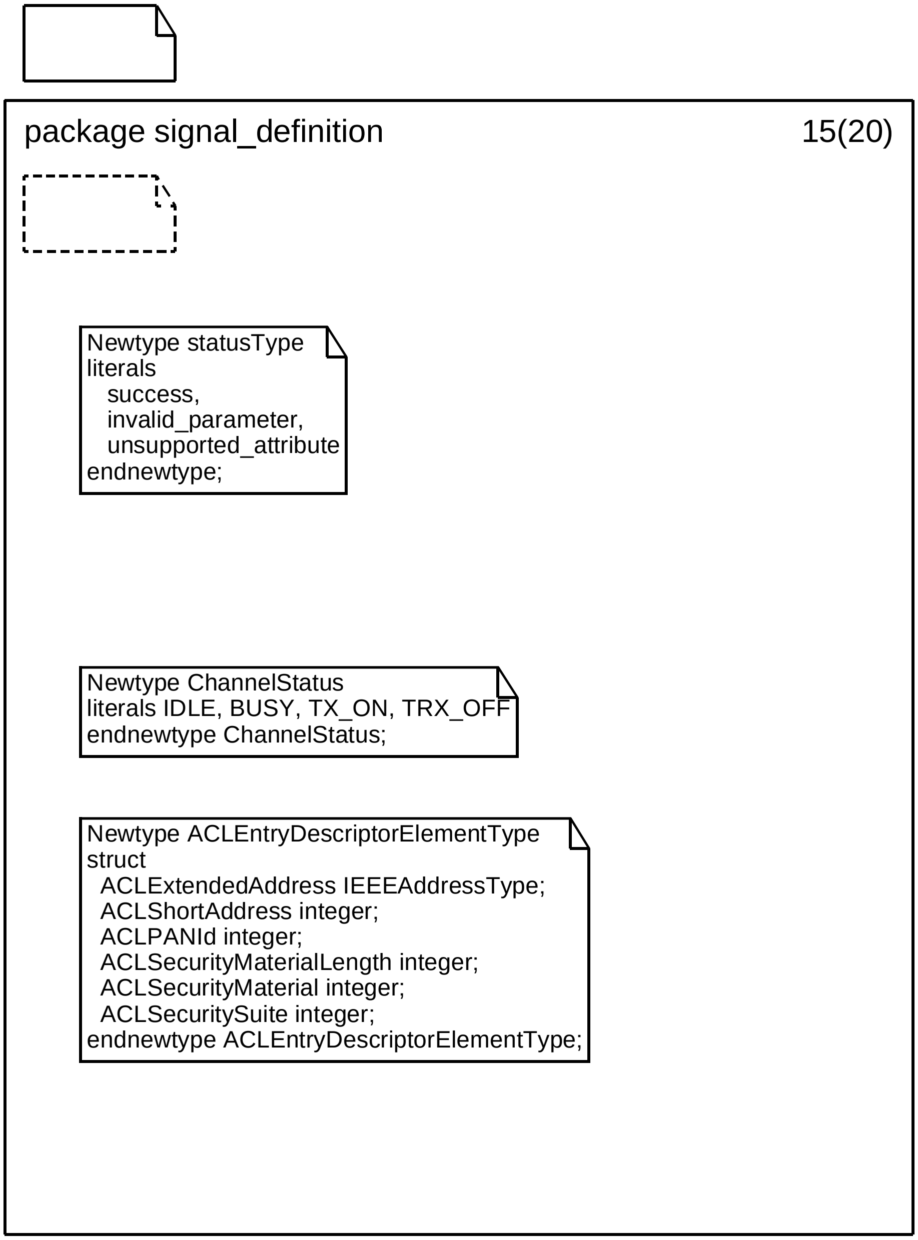
```
Newtype phystateType
literals RX_ON, TRX_OFF, TX_ON
endnewtype physstateType;

Newtype phystatusType
literals SUCCESS, RX_ON, TRX_OFF, TX_ON
endnewtype phystatusType;
```

```
syntype psduLength = Integer
endsyntype;

syntype EnergyLevel = Integer
endsyntype;

syntype ppduLinkQuality = Integer
endsyntype;
```

```
syntype psdu = MPDUType
endsyntype;

newtype PIBAttributeValue choice
    r real;
    i integer;
    b boolean;
    d duration;
    t time;
    x ACLEntryDescriptorElementType;
endnewtype;
```

D.4.15 Signal definition package (15)

s.



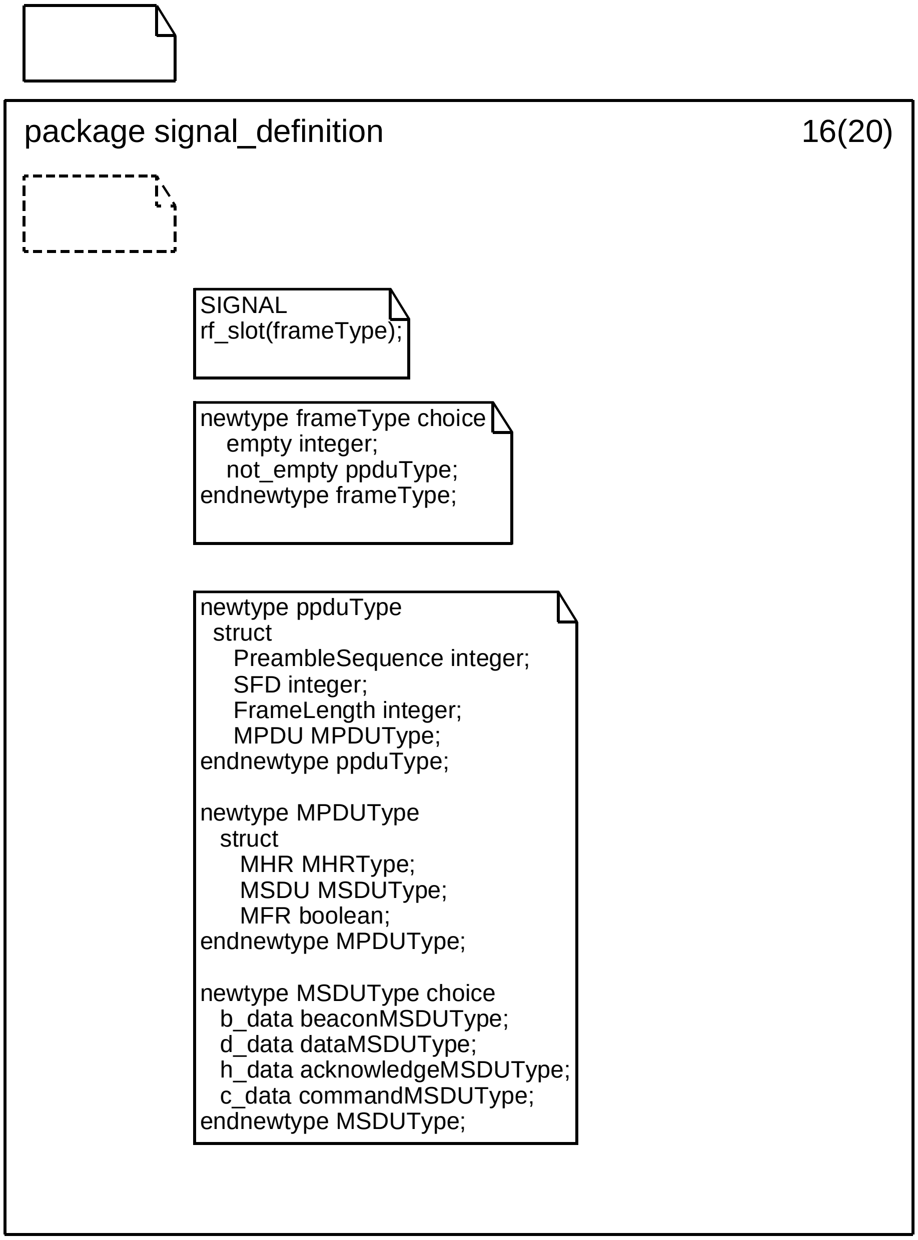
```
Newtype statusType
literals
   success,
   invalid_parameter,
   unsupported_attribute
endnewtype;
```

```
Newtype ChannelStatus
literals IDLE, BUSY, TX_ON, TRX_OFF
endnewtype ChannelStatus;
```

```
Newtype ACLEntryDescriptorElementType
struct
  ACLExtendedAddress IEEEAddressType;
  ACLShortAddress integer;
  ACLPANId integer;
  ACLSecurityMaterialLength integer;
  ACLSecurityMaterial integer;
  ACLSecuritySuite integer;
endnewtype ACLEntryDescriptorElementType;
```

D.4.16 Signal definition package (16)

.



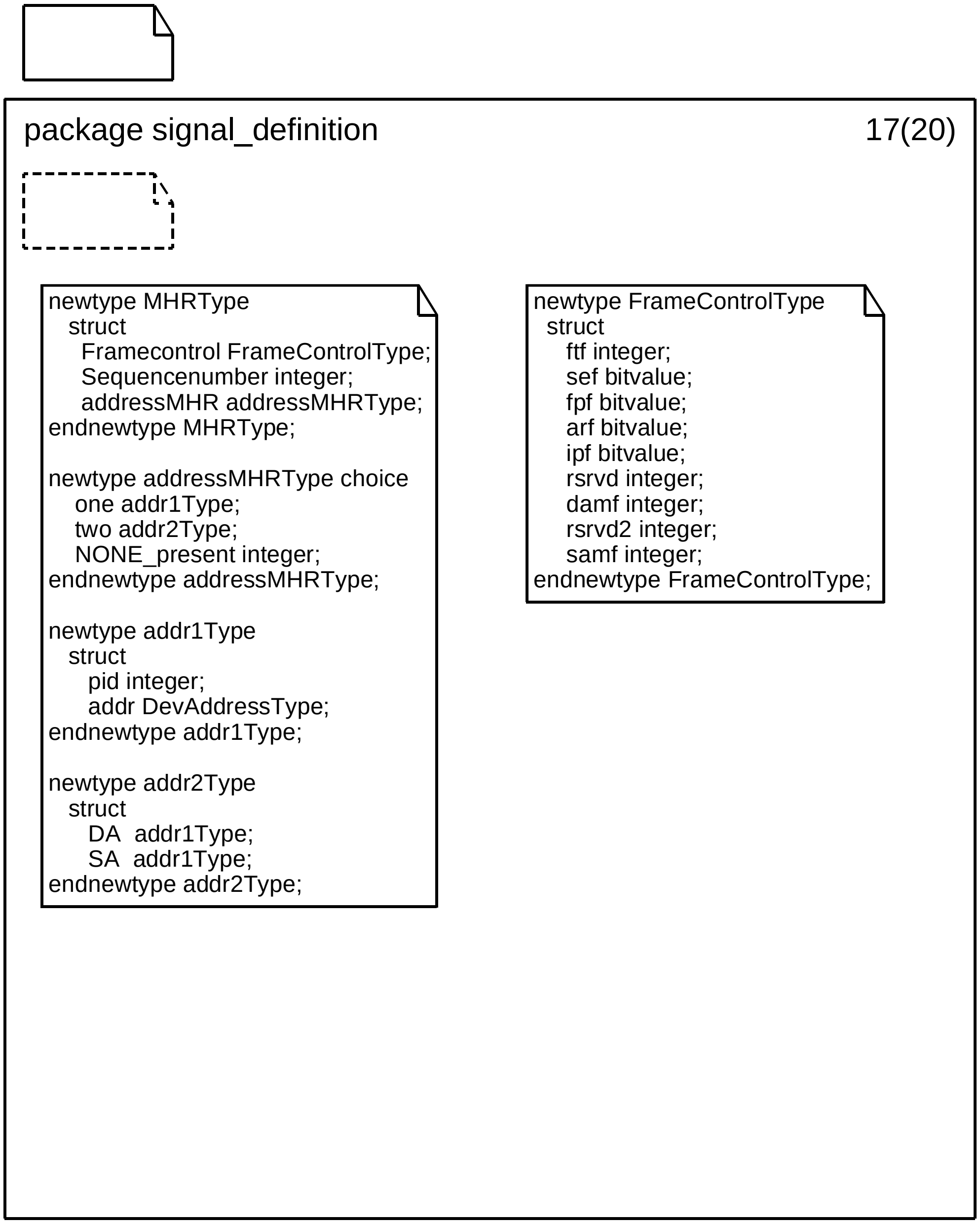
```
SIGNAL
rf_slot(frameType);
```

```
newtype frameType choice
    empty integer;
    not_empty ppduType;
endnewtype frameType;
```

```
newtype ppduType
  struct
     PreambleSequence integer;
     SFD integer;
     FrameLength integer;
     MPDU MPDUType;
endnewtype ppduType;

newtype MPDUType
   struct
      MHR MHRType;
      MSDU MSDUType;
      MFR boolean;
endnewtype MPDUType;

newtype MSDUType choice
   b_data beaconMSDUType;
   d_data dataMSDUType;
   h_data acknowledgeMSDUType;
   c_data commandMSDUType;
endnewtype MSDUType;
```

D.4.17 Signal definition package (17)

.

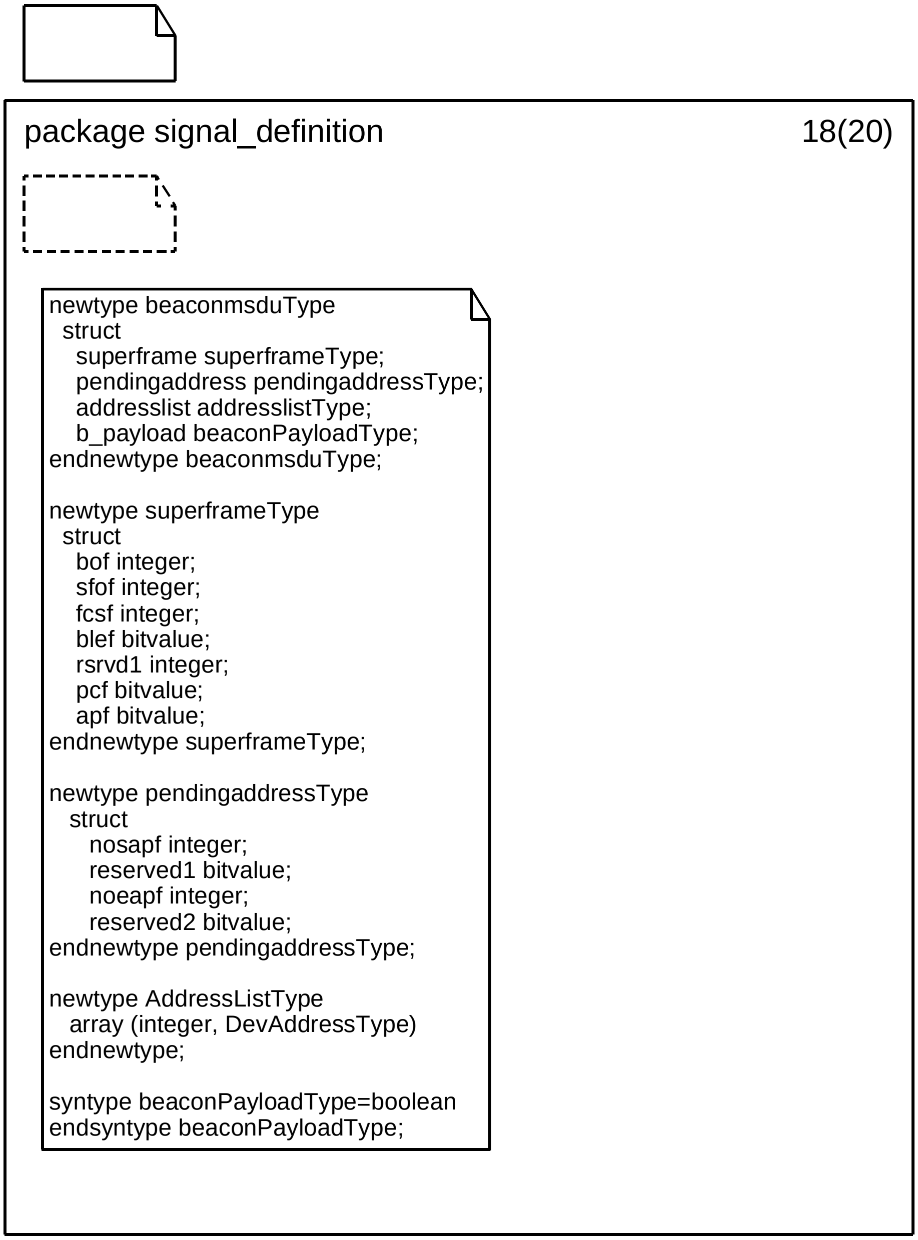


```
newtype MHRType
   struct
     Framecontrol FrameControlType;
     Sequencenumber integer;
     addressMHR addressMHRType;
endnewtype MHRType;

newtype addressMHRType choice
    one addr1Type;
    two addr2Type;
    NONE_present integer;
endnewtype addressMHRType;

newtype addr1Type
   struct
      pid integer;
      addr DevAddressType;
endnewtype addr1Type;

newtype addr2Type
   struct
      DA  addr1Type;
      SA  addr1Type;
endnewtype addr2Type;
```

```
newtype FrameControlType
  struct
     ftf integer;
     sef bitvalue;
     fpf bitvalue;
     arf bitvalue;
     ipf bitvalue;
     rsrvd integer;
     damf integer;
     rsrvd2 integer;
     samf integer;
endnewtype FrameControlType;
```

D.4.18 Signal definition package (18)

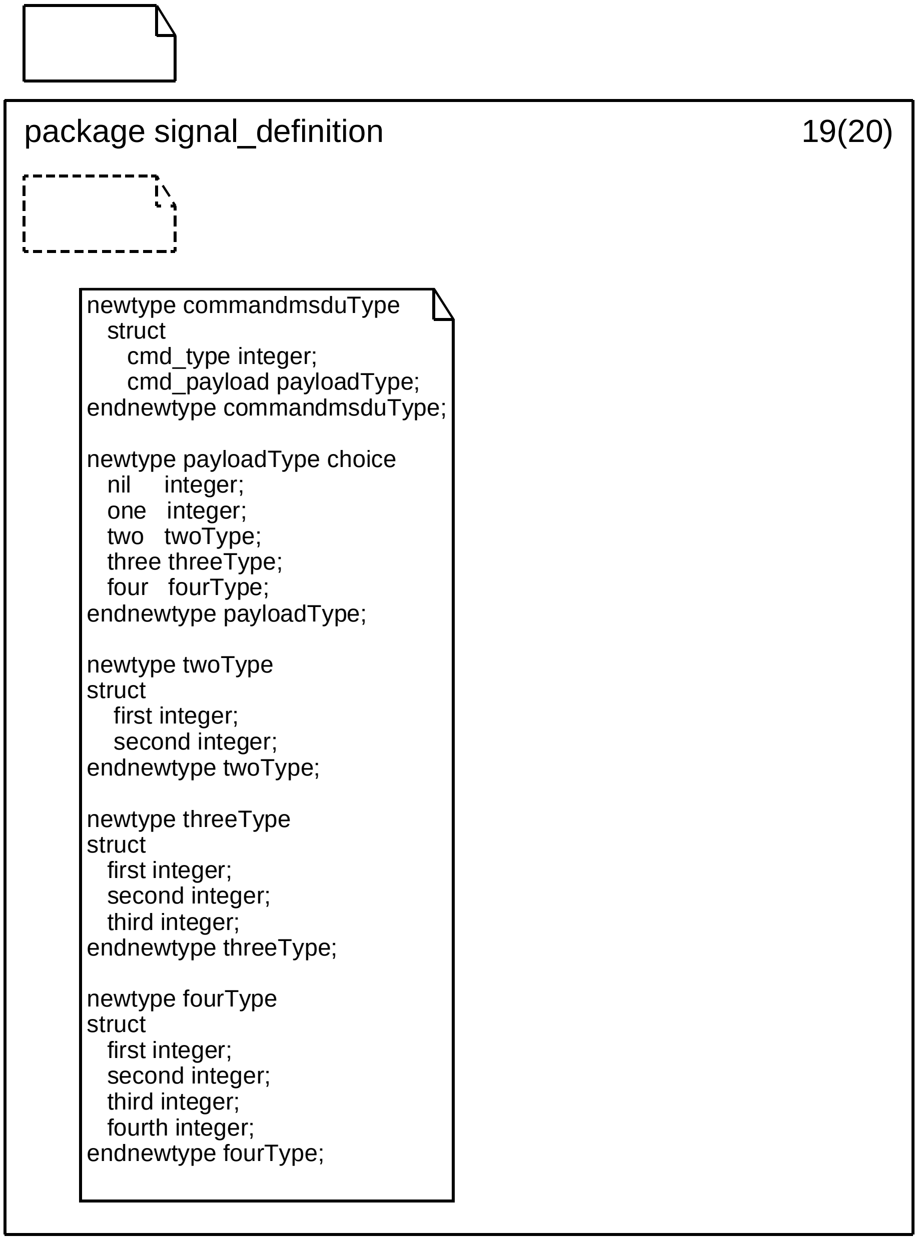
.



```
newtype beaconmsduType
  struct
    superframe superframeType;
    pendingaddress pendingaddressType;
    addresslist addresslistType;
    b_payload beaconPayloadType;
endnewtype beaconmsduType;

newtype superframeType
  struct
    bof integer;
    sfof integer;
    fcsf integer;
    blef bitvalue;
    rsrvd1 integer;
    pcf bitvalue;
    apf bitvalue;
endnewtype superframeType;

newtype pendingaddressType
   struct
      nosapf integer;
      reserved1 bitvalue;
      noeapf integer;
      reserved2 bitvalue;
endnewtype pendingaddressType;

newtype AddressListType
   array (integer, DevAddressType)
endnewtype;

syntype beaconPayloadType=boolean
endsyntype beaconPayloadType;
```

D.4.19 Signal definition package (19)

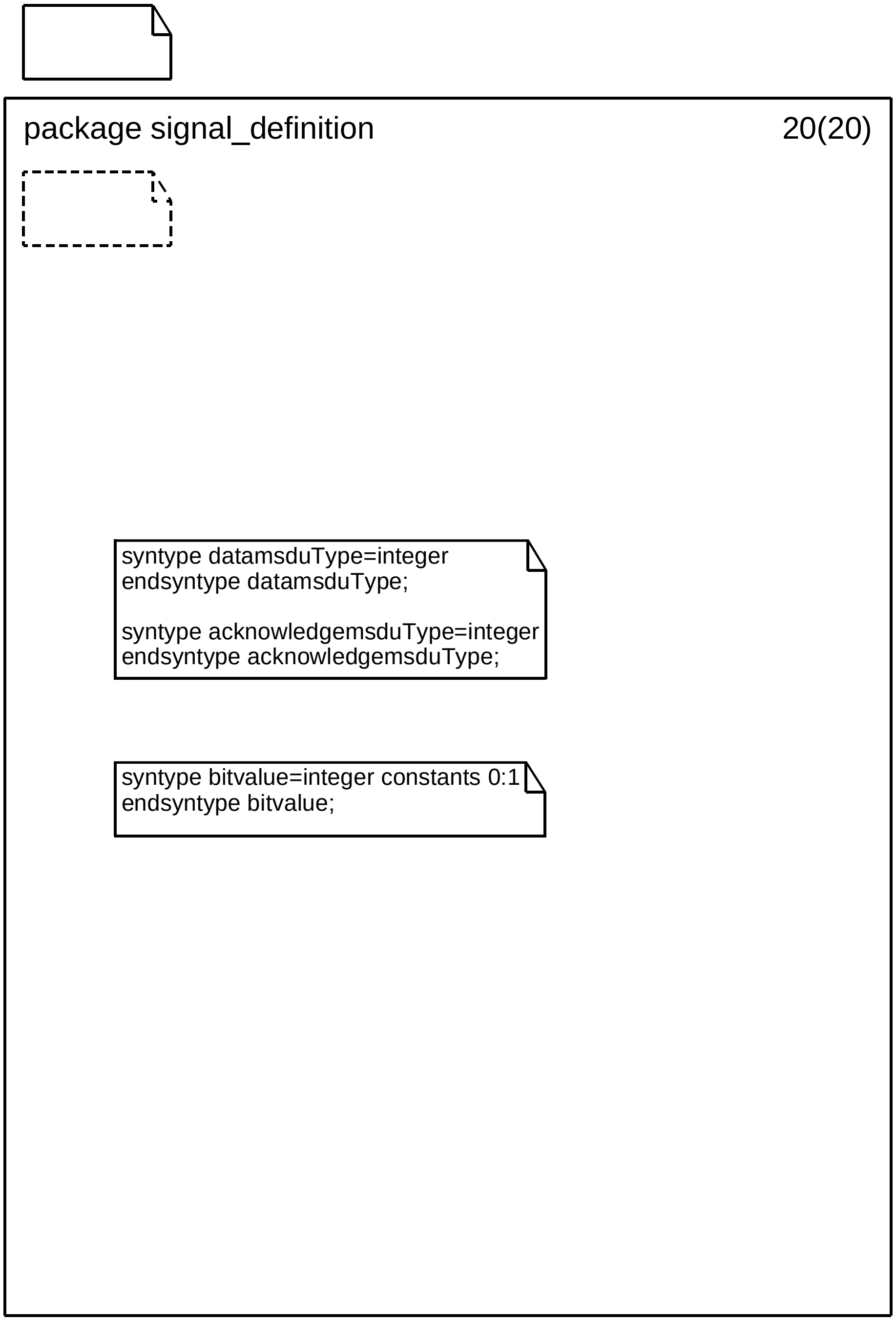
.



```
newtype commandmsduType
   struct
      cmd_type integer;
      cmd_payload payloadType;
endnewtype commandmsduType;

newtype payloadType choice
   nil     integer;
   one    integer;
   two    twoType;
   three threeType;
   four   fourType;
endnewtype payloadType;

newtype twoType
struct
    first integer;
    second integer;
endnewtype twoType;

newtype threeType
struct
   first integer;
   second integer;
   third integer;
endnewtype threeType;

newtype fourType
struct
   first integer;
   second integer;
   third integer;
   fourth integer;
endnewtype fourType;
```

D.4.20 Signal definition package (20)

.



```
syntype datamsduType=integer
endsyntype datamsduType;

syntype acknowledgemsduType=integer
endsyntype acknowledgemsduType;
```

```
syntype bitvalue=integer constants 0:1
endsyntype bitvalue;
```

Annex E

(informative)

Coexistence with other IEEE standards and proposed standards

While not required by the specification, IEEE 802.15.4 devices can be reasonably expected to “coexist,” that is, to operate in proximity to other wireless devices. This annex considers issues regarding coexistence between IEEE 802.15.4 devices and other wireless IEEE-compliant devices.

E.1 Standards and proposed standards characterized for coexistence

This clause enumerates IEEE-compliant devices that are characterized and the devices that are not characterized for operation in proximity to IEEE 802.15.4 devices.

As described in 6.1.2, the IEEE 802.15.4 PHYs are specified for operation in 27 channels. Channel 0 through channel 10 reside in frequencies in the 868 MHz and 915 MHz bands and, therefore, do not significantly interact with other wireless devices covered under the IEEE 802 wireless standards. Channel 11 through channel 26 span frequencies from 2405 MHz to 2480 MHz and, therefore, may interact with other IEEE-compliant devices operating in those frequencies.

Standards and proposed standards characterized in this annex for coexistence are

- IEEE Std 802.11b-1999 (2400 MHz DSSS)
- IEEE Std 802.15.1-2002 [2400 MHz frequency hopping spread spectrum (FHSS)]
- IEEE P802.15.3 (2400 MHz DSSS)

Standards not characterized in this annex for coexistence are:

- IEEE Std 802.11, 1999 Edition, frequency hopping (FH) (2400 MHz FHSS)
- IEEE Std 802.11, 1999 Edition, infrared (IR) (333GHz AM)
- IEEE Std 802.16-2001 (2400 MHz OFDM)
- IEEE Std 802.11a-1999 (5.2GHz DSSS)

E.2 General coexistence issues

IEEE Std 802.15.4-2003 provides several mechanisms that enhance coexistence with other wireless devices operating in the 2400 MHz band. This subclause provides an overview of the mechanisms that are defined in the standard. These mechanisms include

- CCA
- Dynamic channel selection
- Modulation
- ED and LQI
- Low duty cycle
- Low transmit power
- Channel alignment
- Neighbor piconet capability

These mechanisms are described briefly in E.2.1 through E.2.7.

E.2.1 CCA

IEEE 802.15.4 PHYs provide the capability to perform CCA in its CSMA-CA mechanism (see 6.7.9). The PHYs require at least one of the following three CCA methods: ED over a certain threshold, detection of a signal with IEEE 802.15.4 characteristics, or a combination of these methods. Use of the ED option improves coexistence by allowing transmission backoff if the channel is occupied by any device, regardless of the communication protocol it may use.

E.2.2 Modulation

The 2400 MHz PHY specified for IEEE Std 802.15.4-2003 uses a quasi-orthogonal modulation scheme, where each symbol is represented by one of 16 nearly orthogonal PN sequences. This is a power-efficient modulation method that achieves low signal-to-noise ratio (SNR) and signal-to-interference ratio (SIR) requirements at the expense of a signal bandwidth that is significantly larger than the symbol rate. A typical low-cost detector implementation is expected to meet the 1% PER requirement as SNR values of 5-6 dB.

Relatively wideband interference, such as IEEE 802.11b and IEEE P802.15.3, would appear like white noise to an IEEE 802.15.4 receiver. The detector performance in this case is similar to noise performance, but the overall SIR requirement is 9 dB to 10 dB lower because only a fraction of the IEEE 802.11b or IEEE P802.15.3 signal power falls within the IEEE 802.15.4 receiver bandwidth.

The use of PN sequences to represent each symbol in IEEE 802.15.4 offers DSSS-like processing gains to interferers whose bandwidth is smaller than the bandwidth of IEEE 802.15.4. For example, this processing gain helps to reduce the impact of an IEEE 802.15.1 interferer, whose 20 dB bandwidth is roughly 50% smaller than the bandwidth of IEEE 802.15.4. Whereas the SNR requirement is 5 dB to 6 dB for 1% PER in noise, the equivalent SIR requirement for an IEEE 802.15.1 signal centered within the pass band of the IEEE 802.15.4 receiver is only 2 dB.

In terms of interference to others, IEEE 802.15.4 appears as wideband interference to IEEE 802.15.1, and only a fraction (~50%) of the IEEE 802.15.4 signal power falls within the IEEE 802.15.1 receiver bandwidth. Furthermore, due to the bandwidth ratios and to the frequency hopping used in IEEE 802.15.1, IEEE 802.15.4 transmissions will interfere with approximately 3 out of the 79 hops, or approximately 4%. To an IEEE 802.11b receiver, IEEE 802.15.4 looks like a narrowband interferer, and the processing gain resulting from the spread-spectrum techniques in IEEE 802.11b will help reduce the impact of the IEEE 802.15.4 interferer.

E.2.3 ED and LQI

The IEEE 802.15.4 PHYs include two measurement functions that indicate the level of interference within an IEEE 802.15.4 channel. The receiver ED measurement (see 6.7.7) is an estimate of the received signal power within an IEEE 802.15.4 channel and is intended for use as part of a channel selection algorithm at the network layer. The LQI (see 6.7.8) measures the received energy level and/or SNR for each received packet. When energy level and SNR information are combined, they can indicate whether a corrupt packet resulted from low signal strength or from high signal strength plus interference.

E.2.4 Low duty cycle

The specifications of IEEE Std 802.15.4-2003 are tailored for applications with low power and low data rates (a maximum of 250 kb/s and down to 20 kb/s). Typical applications for IEEE 802.15.4 devices are anticipated to run with low duty cycles (under 1%). This will make IEEE 802.15.4 devices less likely to cause interference to other standards.

E.2.5 Low transmit power

Although operation in the 2400 MHz band under Section 15.247 of FCC CFR47 [B14] rules allow transmission powers up to 1 W, IEEE 802.15.4 devices will likely operate with much lower transmit power. A key metric of IEEE Std 802.15.4-2003 is cost, and achieving greater than 10 dBm transmit power in a low-cost system on chip, while feasible, will be economically disadvantageous. Furthermore, European regulations (ETSI EN 300 328) for out-of-band emissions make it difficult to transmit above 10 dBm without additional, expensive filtering. These factors limit the distribution of devices with greater than 10 dBm transmit power to a few specialized applications.

At the low end, the IEEE 802.15.4 PHY specifies that devices must be capable of at least –3 dBm transmit power. At this level, actual transmit power represents a small fraction of the overall power consumed by the transmitter, so there is little benefit in terms of energy savings to operate below this level. However, the standard does encourage operating with lower transmit power, when possible, to minimize interference (see 6.7.5).

Thus the majority of IEEE 802.15.4 devices are expected to operate with transmit powers between –3 and 10 dBm, with 0 dBm being typical. IEEE 802.11b devices also operate under Section 15.247 of FCC CFR47 [B14], where up to 1 W of transmit power is allowed; however, most devices in the market today operate at transmit powers between 12 dBm and 18 dBm. IEEE P802.15.3 devices operate under Section 15.249 of FCC CFR47 [B14], which limits transmit power to 8 dBm EIRP. The EIRP measurement for the IEEE P802.15.3 PHY includes the antenna gain; therefore, a 1 dB increase antenna gain requires a 1 dB decrease in transmit power. In contrast, devices operating under Section 15.247 of FCC CFR47 [B14] are allowed up to 6 dB of antenna gain without modifications to the transmit power.

Assuming moderate antenna gain (~0 dBi) for typical implementations, the discussion in this subclause implies that a nominal IEEE 802.15.4 transmitter would operate about 8 dB less than the IEEE P802.15.3 transmitter and about 12 dB to 18 dB less than a typical IEEE 802.11b implementation.

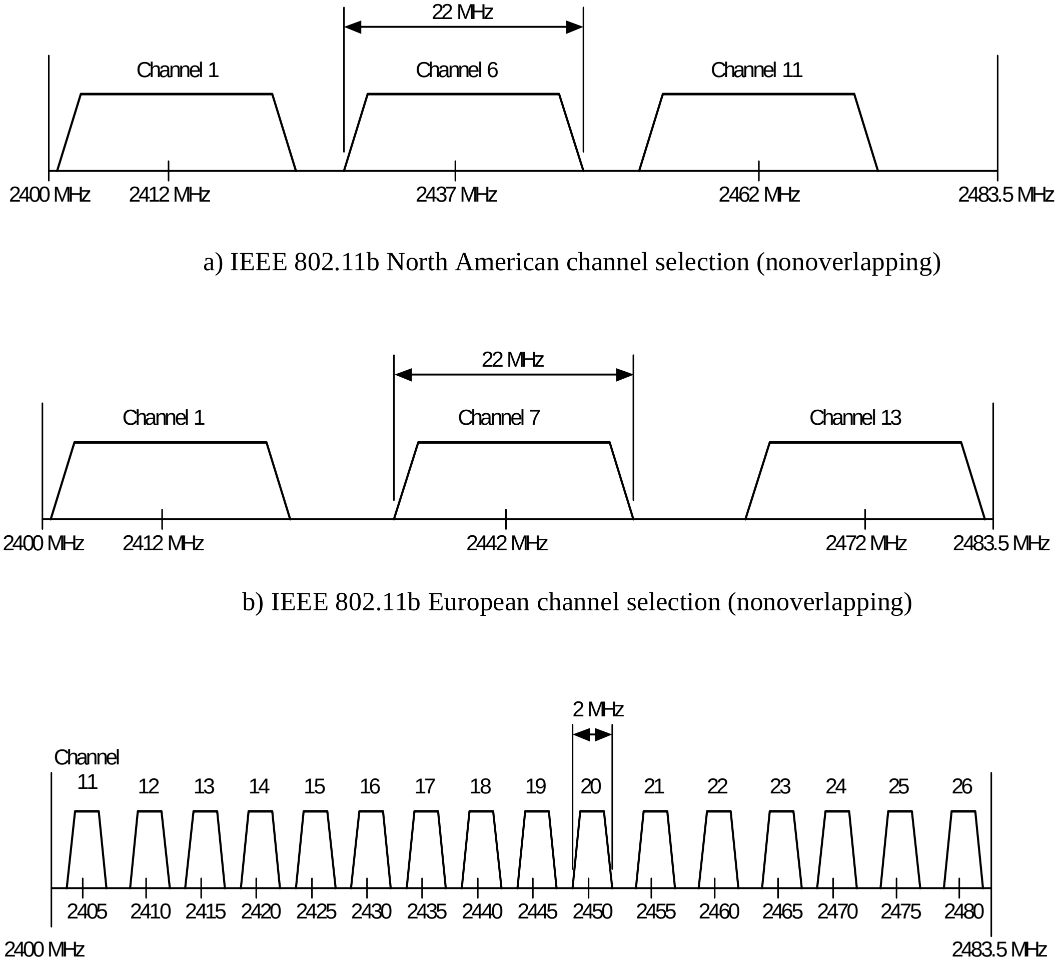
E.2.6 Channel alignment

The alignment between IEEE 802.11b (nonoverlapping sets) and IEEE 802.15.4 channels are shown in Figure E.1. There are four IEEE 802.15.4 channels that fall in the guard bands between (or above) the three IEEE 802.11b channels (*n* = 15, 20, 25, 26 for North America; *n* = 15, 16, 21, 22 in Europe). While the energy in this guard space will not be zero, it will be lower than the energy within the channels; and operating an IEEE 802.15.4 network on one of these channels will minimize interference between systems.

When performing dynamic channel selection, either at network initialization or in response to an outage, an IEEE 802.15.4 device will scan a set of channels specified by the ChannelList parameter. For IEEE 802.15.4 networks that are installed in areas known to have high IEEE 802.11b activity, the ChannelList parameter can be defined as the above sets in order to enhance the coexistence of the networks.

E.2.7 Neighbor piconet capability

Interoperability with other systems is beyond the scope of IEEE Std 802.15.4-2003. However, certain schemes may be envisaged for this purpose, for example, the PAN coordinator can set aside GTSs specifically for use by other systems. This type of neighbor piconet support capability may further alleviate interference with other systems.



a) IEEE 802.11b North American channel selection (nonoverlapping)

b) IEEE 802.11b European channel selection (nonoverlapping)

c) IEEE 802.15.4 channel selection (2400 MHz PHY)

Figure E.1—IEEE 802.15.4 (2400 MHz PHY) and IEEE 802.11b channel selection

E.3 Coexistence performance

The assumptions made across all standards characterized for coexistence are described in E.3.1. Subclauses E.3.2 and E.3.3 describe the assumptions made for individual standards and quantify their predicted performance when coexisting with IEEE 802.15.4 devices.

E.3.1 Assumptions for coexistence quantification

The assumptions in E.3.1.1 through E.3.1.9 are made to determine the level of coexistence.

E.3.1.1 Channel model

The channel model is based on the IEEE 802.11 specification used by IEEE P802.15.2 and IEEE P802.15.3.

$$d = 10^{\frac{(P_t - P_r - 40.2)}{20}} \qquad \text{for } d < 8\text{m}$$

$$d = 8 \times 10^{\frac{(P_t - P_r - 58.5)}{33}} \qquad \text{for } d > 8 \text{ m}$$

E.3.1.2 Receiver sensitivity

The receiver sensitivity assumed is the reference sensitivity specified in each standards as follows:

a) –76 dBm for IEEE 802.11b 11 Mb/s CCK
b) –70 dBm for IEEE 802.15.1
c) –75 dBm for IEEE P802.15.3 22 Mb/s DQPSK
d) –85 dBm for IEEE 802.15.4

E.3.1.3 Transmit power

The transmitter power for each coexisting standard has been specified as follows:

a) 14 dBm for IEEE 802.11b
b) 0 dBm for IEEE 802.15.1
c) 8 dBm for IEEE P802.15.3
d) 0 dBm for IEEE 802.15.4

E.3.1.4 Receiver bandwidth

The receiver bandwidth is as required by each standard as follows:

a) 22 MHz for IEEE 802.11b
b) 1 MHz for IEEE 802.15.1
c) 15 MHz for IEEE P802.15.3
d) 2 MHz for IEEE 802.15.4

E.3.1.5 Transmit spectral masks

The maximum transmitter spectral masks are assumed for the calculations. This assumption is the absolute worst-case scenario; in most cases, the transmitter spectrum will be lower.

Table E.1—Transmit mask for IEEE 802.11b (see 18.4.7.3 in IEEE 802.11b)

Frequency	Relative limit
f_c – 22 MHz < f < f_c – 11 MHz and f_c + 11 MHz < f < f_c + 22 MHz	–30 dBr
f < f_c – 22 MHz and f > f_c + 22 MHz	–50 dBr

Table E.2—Transmit mask for IEEE 802.15.1 (see 7.2.3.1 in IEEE 802.15.1)

Frequency offset	Transmit power
± 500 kHz	–20 dBc
|M – N| = 2	–20 dBm
|M – N| ≥ 3	–40 dBm
The transmitter is transmitting on channel M, and the adjacent channel power is measured on channel number N.	

Table E.3—Transmit mask for IEEE P802.15.3 (see 11.5.3 in IEEE P802.15.3)

Frequency offset	Relative limit
7.5 MHz < $\lvert f - f_c \rvert$ < 15 MHz	–30 dBr
15 MHz < $\lvert f - f_c \rvert$ < 22 MHz	–1 /7[$\lvert f - f_c$ (MHz)$\rvert$ + 13] dBr
22 MHz < $\lvert f - f_c \rvert$	–50 dBr

Table E.4—Transmit mask for IEEE 802.15.4 (see 6.5.3.1 in IEEE 802.15.4)

Frequency	Relative limit	Absolute limit
$\lvert f - f_c \rvert$ >3.5 MHz	–20 dBr	–30 dBm

E.3.1.6 IEEE 802.11b transmit PSD

Because IEEE 802.11 implementations will generally meet FCC requirements, they will achieve an absolute power of less than –41.3 dBm/MHz at a separation of 22 MHz from the carrier frequency. The reason for this is that there is a restricted band that ends at 2.39 GHz, which is 22 MHz from the center of the lowest channel used for the FCC regulatory domain (see 18.4.6.2 in [G3]). Thus, the relative power for greater than 22 MHz separation would be +14 dBm – (–41.3 dBm) = 55.3 dB.

E.3.1.7 Interference characteristics

The effect of the interfering signal on the desired signal is assumed to be similar to additive white Gaussian noise (AWGN) in the same bandwidth.

E.3.1.8 Bit error rate (BER) calculations

The BER calculations are as described in 5.3 of IEEE P802.15.2:

1) BER for IEEE 802.11b at 1 Mb/s = $Q(11 \times SINR)^{\frac{1}{2}}$

2) BER for IEEE 802.11b at 2 Mb/s = $\left(5.5 \times \frac{SINR}{2}\right)^{\frac{1}{2}}$

3) BER for IEEE 802.11b at 5.5 Mb/s = $\frac{8}{15} \times \left(14 \times Q(8 \times SINR)^{\frac{1}{2}} + Q(16 \times SINR)^{\frac{1}{2}}\right)$

4) BER for IEEE 802.11b at 11 Mb/s =

$$\frac{128}{255} \times \left(24 \times Q(4 \times SINR)^{\frac{1}{2}} + 16 \times Q(16 \times SINR)^{\frac{1}{2}} + 174 \times Q(8 \times SINR)^{\frac{1}{2}} \ldots\right.$$

$$\left.16 \times Q(10 \times SINR)^{\frac{1}{2}} + 24 \times Q(12 \times SINR)^{\frac{1}{2}} + Q(16 \times SINR)^{\frac{1}{2}}\right)$$

5) BER for IEEE 802.15.1 = $0.5 \times e^{\frac{-SINR}{2}}$

6) BER for IEEE P802.15.3 at 11 Mb/s= $Q\left(SINR^{\frac{1}{2}}\right)$

7) BER for IEEE 802.15.4 = $\frac{8}{15} \times \frac{1}{16} \times \sum_{k=2}^{16} -1^{k}\binom{16}{k} e^{\left(20 \times SINR \times \left(\frac{1}{k} - 1\right)\right)}$

E.3.1.9 PER

To convert between BER and PER, the following average packet lengths are assumed:

a) Average frame for IEEE 802.11b = 1024 bytes
b) Average frame for IEEE 802.15.1 = 1024 bytes
c) Average frame length for IEEE P802.15.3 = 1024 bytes
d) Average frame length for IEEE 802.15.4 = 22 bytes

E.3.2 BER model

This subclause presents the BER for standards characterized for coexistence. The BER results were obtained using the analytical model from IEEE P802.15.2. The calculation follows the approach outlined in 5.3.2 of that document and the conversion from SNR to BER uses the formulas in 5.3.6 of that document. Figure E.2 illustrates the relationship between BER and SNR for IEEE 802.11b, IEEE P802.15.3 base rate, IEEE 802.15.1, and IEEE 802.15.4.

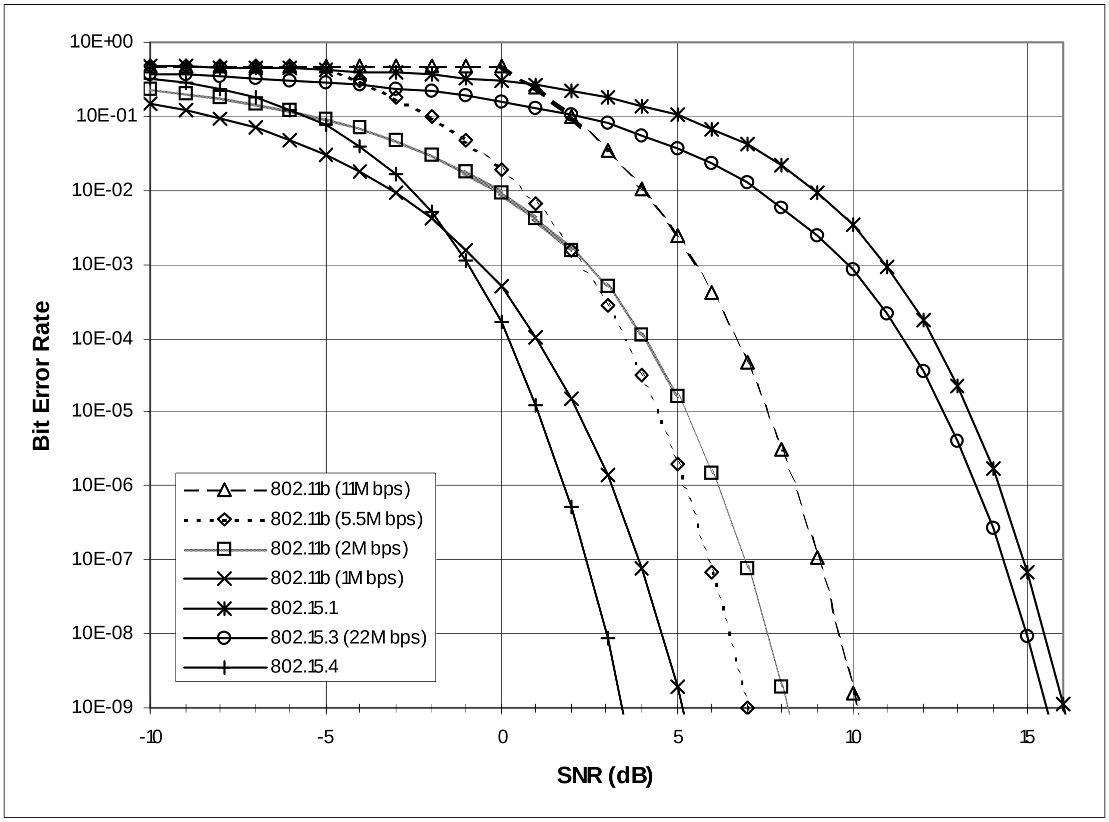


Figure E.2—BER Results for IEEE 802.11, IEEE 802.15.1, IEEE P802.15.3 and IEEE 802.15.4

E.3.3 Coexistence simulation results

Using the assumptions outlined in E.3.2, an analytical simulation tool was developed to quantify the effect of interference between neighboring devices. For each of the cases studied, the receiver under test was presented with a desired signal at 10 dB above the required sensitivity (see E.3.1.2) and a single interfering device with appropriate transmit power (see E.3.1.3). The amount of received interference power was determined using the propagation model (see E.3.1.1) as well as the transmit PSD (see E.3.1.5) and receiver bandwidth (E.3.1.4), and the resulting SIR level was used to estimate the achievable PER.

The simulation output (see Figure E.3 through Figure E.8) shows the PER versus separation distance and frequency offset for various combinations of devices. When comparing the results, some obvious features stand out. First, for the nonhopping systems, large frequency offsets allow close-proximity coexistence (less than 2 m separation), while low-frequency offsets, or co-channel interference, require separation distances in the tens of meters. Therefore, as expected, the ability to detect channel occupancy and perform dynamic channel selection is an important mechanism for coexistence.

A second observation is that transmit power level is the dominant factor in co-channel interference situations. When a low-power IEEE 802.15.4 device is moved toward an IEEE 802.11b or IEEE P802.15.3 device, the IEEE 802.15.4 device is the first to degrade. IEEE 802.15.1 and IEEE 802.15.4 have similar transmit powers, and their interference effects on each other are similar.

Even with its low transmit power level, the results presented here suggest that an IEEE 802.15.4 device can cause degradation to the other devices in co-channel situations with separation distances below 20 m. However, in practice, several IEEE 802.15.4 coexistence features (which were not included in this PHY simulation) will help to further reduce the occurrence and severity of co-channel interference. These include the very low duty cycle operation for typical IEEE 802.15.4 applications, as well as the use of CCA prior to transmission (CSMA-CA).

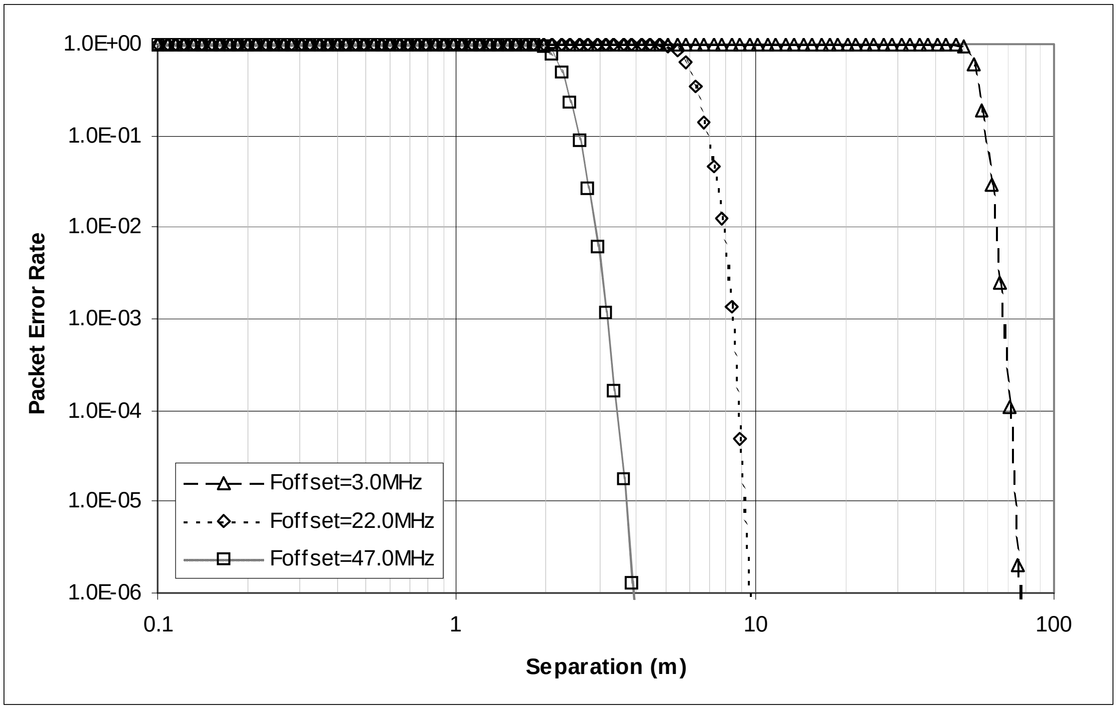


Figure E.3—IEEE 802.15.4 receiver, IEEE 802.11b interferer

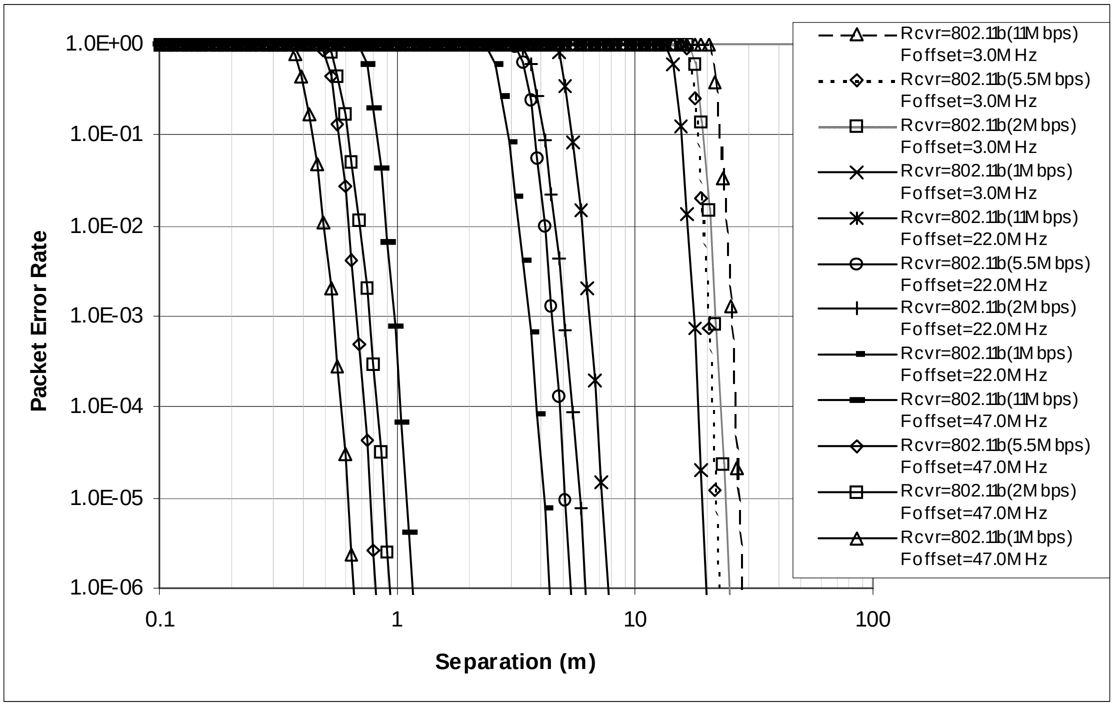


Figure E.4—IEEE 802.11b receiver, IEEE 802.15.4 interferer

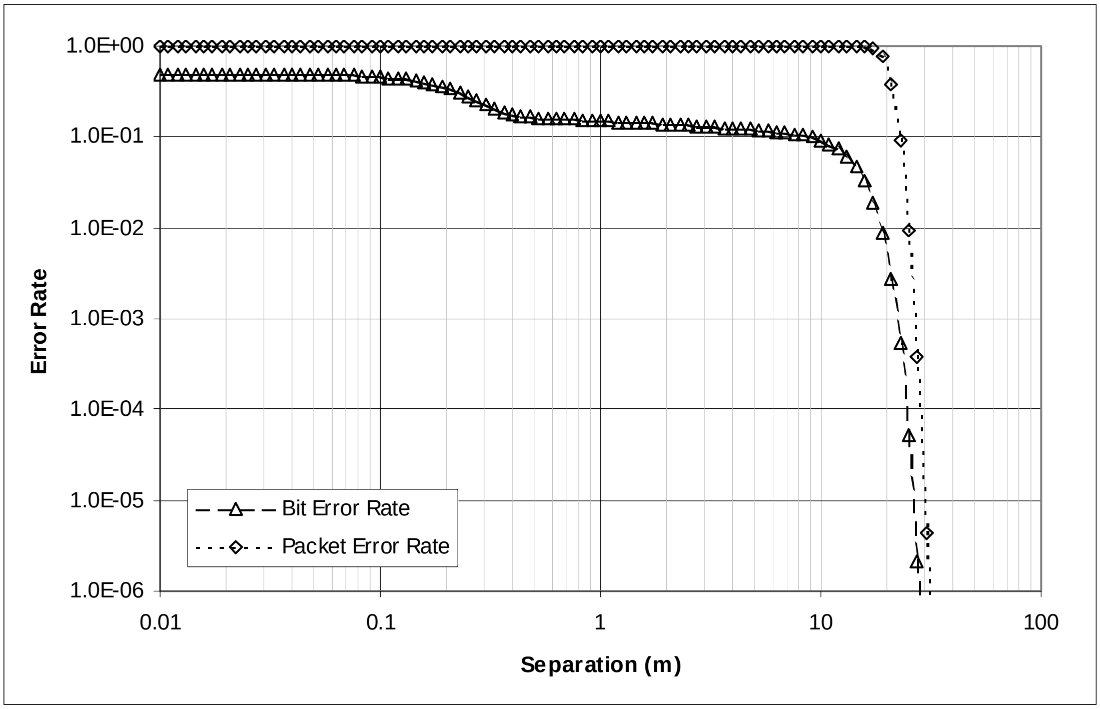


Figure E.5—IEEE 802.15.4 receiver, IEEE 802.15.1 interferer

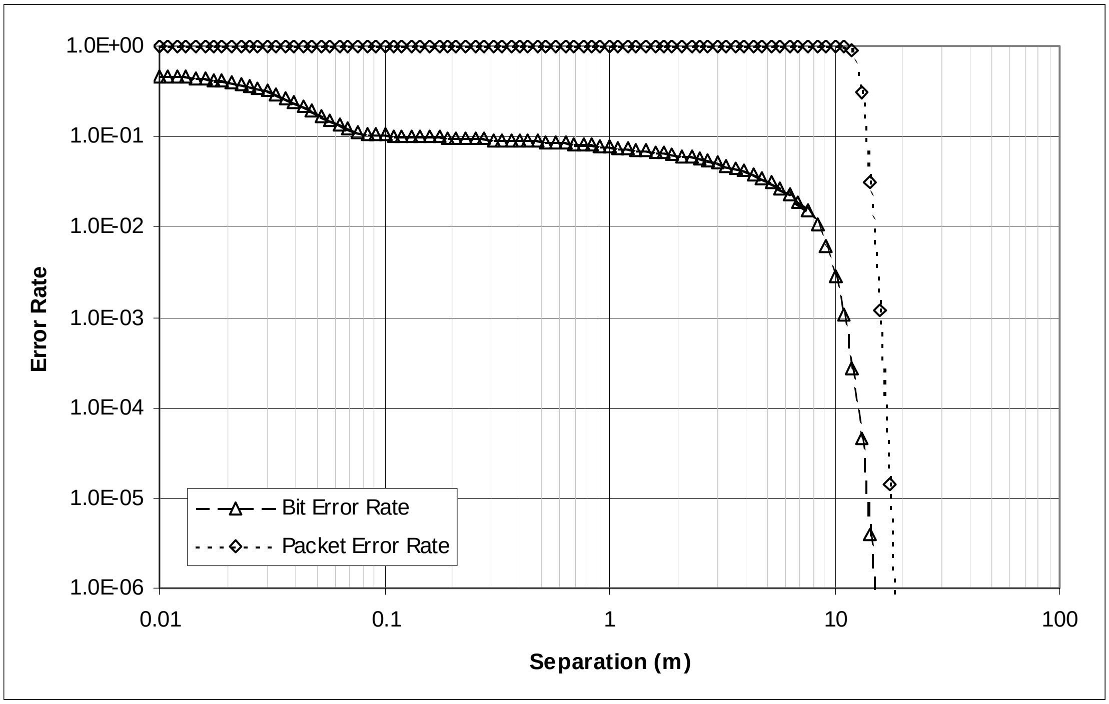


Figure E.6—IEEE 802.15.1 receiver, IEEE 802.15.4 interferer

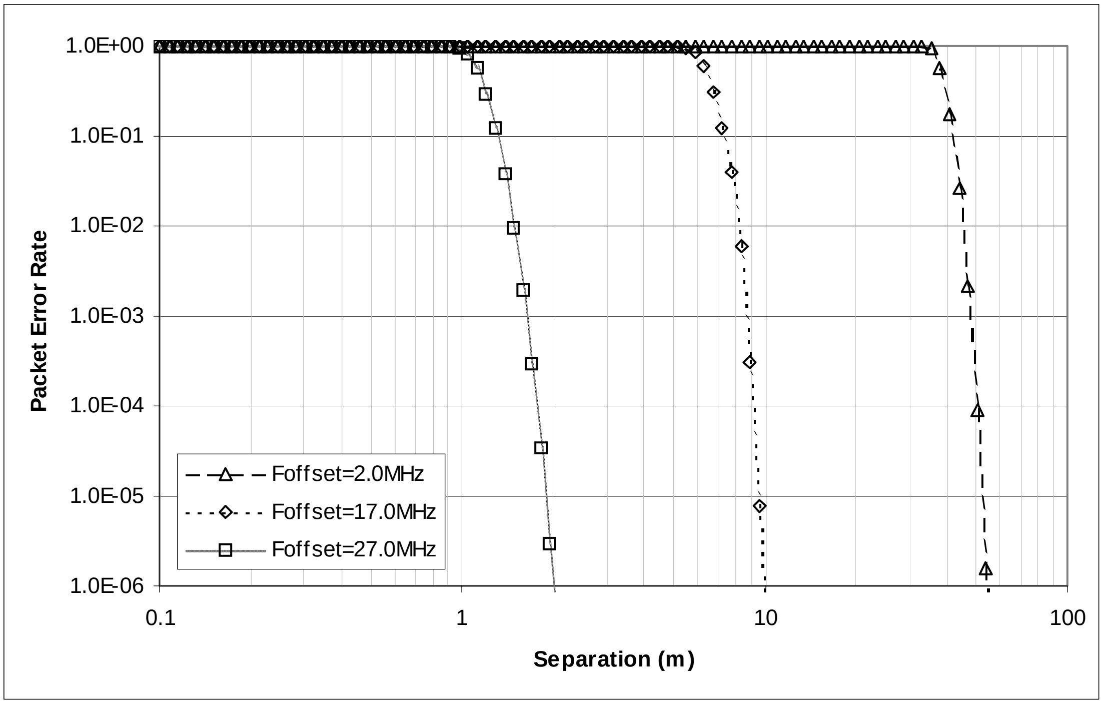


Figure E.7—IEEE 802.15.4 receiver, IEEE P802.15.3 interferer

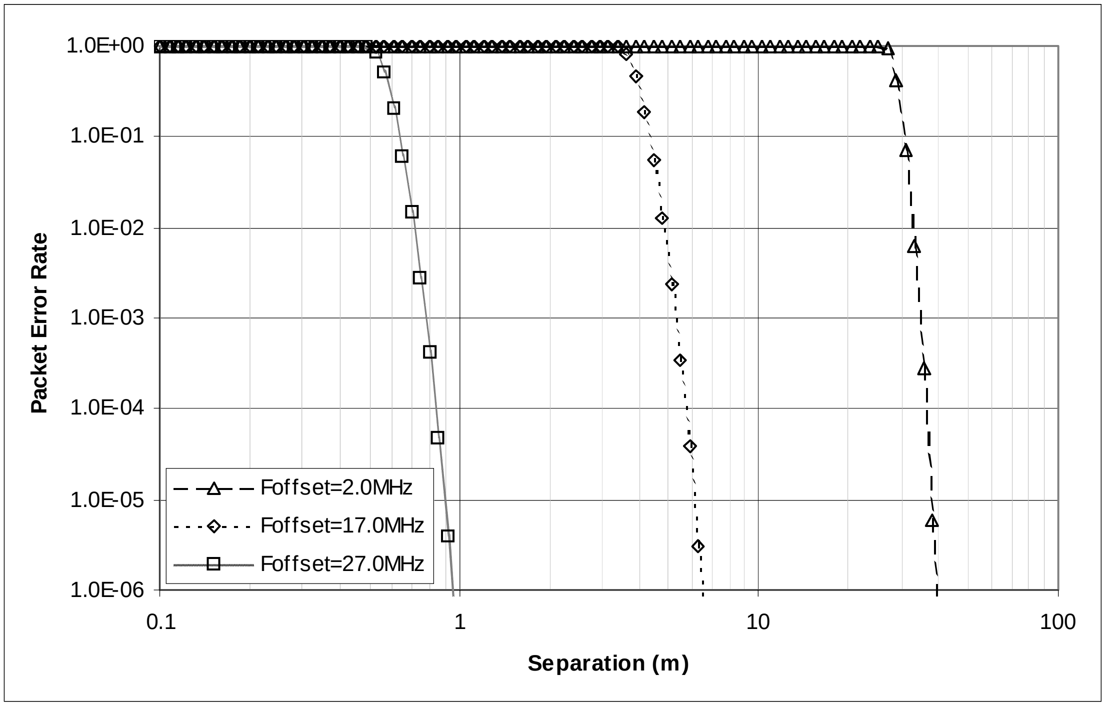


Figure E.8—IEEE P802.15.3 receiver, IEEE 802.15.4 interferer

E.4 Notes on the calculations

The calculations for this annex were based on the formulas and descriptions from IEEE P802.15.2.

Annex F

(informative)

IEEE 802.15.4 regulatory requirements

F.1 Introduction

This annex provides informational references to and key summary of the more critical regulatory requirements, with analysis of the impact on the specifications and design of IEEE 802.15.4 products where appropriate. Due to the great breadth and variety of worldwide regulations it is not possible to cover all nations in the available time and space. Fortunately, the 2400 MHz band is standardized for unlicensed operation nearly worldwide, and it is believed that the material contained in this annex captures the key requirements to fielding a product that can meet the rules in most major markets. However, the published rules are not always clear. Despite some degree of standardization, there are many differences between nations; and in some cases the various rules may appear to contradict each other without it being clear which rule has the force of law. Common practice and historical interpretation also often impact the situation, and the rules and their interpretations are also in a continuous state of flux. A good faith effort has been made here to provide accurate summary and interpretation of the rules and to predict impact upon system design, but no warranty is made other than a sincere effort to be helpful and to save time for users of the standard. The final word on regulatory requirements is under the jurisdiction of the individual nations in which a product is fielded. Manufacturers must take the responsibility to check specifications against the latest regulations of nations into which they market, using the measured results of certified electromagnetic compatibility (EMC) laboratories.

It is very enabling to the business of short-range radio and WPANs that these applications are granted allocations for certified or unlicensed operation where the manufacturer ensures that the product meets technical requirements and the end user does not require a license. There are regulatory allowances for this type of operation in the United States; most nations of Europe, Japan, and Canada; and many other nations around the world. Because this type of operation is generally uncoordinated as to frequency usage in a geographic region, it is usually understood that systems operating in these bands must accept interference from other users without regulatory recourse. The emphasis on competent design practice within the rules encourages good immunity to interference. The European rules go further than U.S. (FCC) rules by actually mandating certain performance standards.

In the United States, the regulations are given under FCC CFR47 [B14]. FCC rules contained there provide allocations in the 260–470 MHz range (see Section 15.231 of FCC CFR47), and in the 902–928 MHz and 2400–2483.5 MHz industrial, scientific, and medical (ISM) bands. In the ISM bands, Section 15.249 of FCC CFR47 provides for narrowband operation of up to approximately 1 mW effective radiated power (ERP) (if wideband, then 1 mW per 100 kHz below 1000 MHz, and 1 mW/MHz above 1000 MHz), and Section 15.247 of FCC CFR47 provides for wideband operation at up to 1 W transmitted power. In these FCC bands, the 260–470 MHz range is generally restricted on power, transmit duty cycle, and application so that it is primarily used for control and security applications, such as keyless entry. The ISM bands allow continuous transmission at higher power levels and are thus used for higher end applications. Section 15.247 of FCC CFR47 is the service category under which IEEE 802.15.4 equipment would most often be certified. Changes to this FCC rule that became effective May 30, 2002, eliminated the requirement in the United States for spread spectrum if the combination of data rate, coding, and modulation method has a 6 dB bandwidth greater than 500 kHz and a maximum transmitted spectral density of less than +8 dBm/3 kHz. However, either DSSS or FHSS may still be used to meet the requirements under Section 15.247 of FCC CFR47.

Canada provides for the same ISM bands and general operating modes as the United States. Canadian requirements are so similar to FCC requirements that in general for 902–928 MHz and 2400–2483.5 MHz band ISM operation the rules may be considered equivalent except for occasional time lag for the Canadian rules to be modified following changes to U.S. rules. The interpretations made later concerning allowed emissions in the United States as compared to Europe may be considered to also apply when comparing Canada to Europe. Section 6.2.2 (m) Canadian RSS-210 provides numerical requirements for narrowband operation in 902–928 MHz and 2400–2483.5 MHz and associated harmonic limits that are identical to operation according to Section 15.249 of FCC CFR47. As of the time of this writing, the Canadian requirements allowances for spread spectrum operation given in Section 6.2.2 (o) of Canadian RSS-210 are equivalent to the previous (prior to May 30, 2002) requirements of Section 15.247 of FCC CFR47. These requirements allow for carrier power up to 1 W for DSSS systems in both 902–928 MHz and 2400–2483.5 MHz if direct sequence processing gain is a minimum of 10 dB. However, because Canada has apparently adopted a strategy of being closely aligned with U.S. regulations, it is likely that Canada will in the near future adopt the changes recently made in the United States regarding digital modulation, which eliminated the processing gain requirement.

In Europe the band segment from 433.05 MHz to 434.79 MHz is the common control and security band and is primarily limited to these applications because of a general 10% duty cycle limit (see ERC 70-03 [B13]). Europe does not offer an ISM band from 902 MHz to 928 MHz, but does offer a limited band from 868 MHz to 870 MHz (see ERC 70.03 for general rules and Table F.7 in this annex for transmit duty cycle limits). The 868–870 MHz band may be expanded in the near future, as explained later in this annex under the more detailed European rules description. The 2400–2483.5 MHz ISM band is provided for also in ERC 70-03 for short-range devices (SRDs) (any digital modulation form), and also in ETSI EN 300 328 for spread spectrum devices with data rates equal to or greater than 250 kb/s, which includes IEEE 802.15.4. While this document does not provide the direct force of law within the European Community, it is generally followed by the regulations of each nation. However, individual nations may have variances that must be checked for fielding of equipment in a particular European country.

For operation in Japan in the 2400 MHz band, the governing document is generally ARIB STD-T66 [B14]. Although ARIB is an industry association, it has been chartered by the Japanese government to perform certain quasi-governmental functions in support of efficient use of the radio spectrum. Its standards are intended to capture both regulatory mandates and extensions over minimum government requirements that provide for efficient use of the radio spectrum.

Thus 2400–2483.5 MHz is the only worldwide allocation of spectrum for unlicensed usage without any limitations on applications and transmit duty cycle. It provides up to 1 W transmit power in spread spectrum modes in the United States, up to 100 mW in Europe, and up to 10 mW/MHz in Japan. Based on these regulatory opportunities, 2400–2483.5 MHz has been selected as the primary IEEE 802.15.4 band. Although IEEE 802.15.4 equipment is generally envisioned to operate with a maximum transmit power of approximately 0 dBm, the international community generally allows a minimum of +10 dBm in the 2400 MHz band. Although harmonic and spurious requirements vary, they may generally be met with –20 to –40 dBc of rejection relative to carrier power in the 0 to +10 dBm range. Spurious suppression requirements vary considerably not only by nation, but also by test methodology and the use of averaging. Where such information is clearly supplied by regulatory agencies, it is included and analyzed in this annex, but no claim to completeness or accuracy is made herein.

The European 868–870 MHz and the U.S. 902–928 MHz bands, being lower frequency and thus accessible in lower capability integrated circuit processes, are special cases of regional bands that are still highly useful even though they are not allocated worldwide. Given that antennas remain approximately omnidirectional, they also provide larger antenna aperture than 2400 MHz, so at a given transmit power, data rate, receiver sensitivity, and reliability level they will provide greater range. Because of their usefulness, a separate air interface specification is provided for these bands, and detailed discussion of regulatory impact is provided in this annex for these bands as well.

Table F.1 shows several websites that may be accessed for direct regulatory information. It is recommended that manufacturers review the latest regulations and utilize the latest test and certification methodologies before fielding new equipment.

Table F.1—Regulatory-related web sites

Web Site	Comment
http://www.iec.ch	Main website for the IEC, keepers of the CISPR 16 measurement standard.
http://www.fcc.gov/oet/info/rules	Specific area of the main FCC website that points to a site where Section 15 rules may be downloaded.
http://strategis.ic.gc.ca/sc_mrksv/spectrum/engdoc/spect1.html	Website for Canadian regulatory authority ,Industry Canada. Major documents include GL-36 [B15] governing 2400 MHz operation, RSP-100 for certification procedures, and Canadian RSS-210 governing 902–928 MHz and 2450 MHz.
http://www.etsi.org	Europe, may download ETSI EN 300 220-1 [B10] (test methodology) and ETSI EN 300 683 (EMC compliance)
http://www.ero.dk	Europe, may download ERC 70-03E [B13], general description of allowed applications, bands, powers, etc., adopted by most European nations.
http://www.tele.soumu.go.jp/e/	English language version of Japanese Radio Law.
http://www.telec.or.jp/ENG/Index_e.htm	English language version of Japanese regulatory certification requirements
http://www.arib.or.jp/english/html/overview/st_e.html	Listing of Japanese technical radio standards documents. ARIB STD-T66 [B14] is the document relative to IEEE 802.15.4.

F.2 Applicable U.S. (FCC) rules

The FCC rules officially govern operation only in the United States, but are followed in varying degrees by many other nations in the Americas and the Pacific rim. As mentioned earlier, the Canadian requirements are usually effectively identical except for time lag to align the Canadian rules following U.S. rule changes. There are five specific FCC rules of high interest to designers of IEEE 802.15 class systems. These sections are 15.35, 15.209, 15.205, 15.247, and 15.249 of FCC CFR47 [B14].

F.2.1 Section 15.35 of FCC CFR47

The rule of Section 15.35 of FCC CFR47 gives the requirements for detector and averaging functions for certification measurements. These issues often have a significant effect on the other rules by which equipment is certified, so much so that understanding this rule and how it is interpreted is critical to developing the highest performance and most cost-effective product. FCC section 15.35 (b) specifies use of a “CISPR quasi-peak” detector function when measuring field strength levels at frequencies below 1000 MHz or, alternately, a peak detector function using the same bandwidth as the quasi-peak detector. Part 15 does not directly define this quasi-peak detector, but instead references CISPR 16 of the IEC. Section 4.2 of Canadian RSS-210 also specifies use of the “CISPR-16” detector. ETSI EN 300 220-1 [B10] governing European test requirements similarly refers to the use of the CISPR 16 detector. FCC 15.35 (b) states that unless otherwise stated (in a separate rule applicable to a special case), an averaging detector shall be used above 1000 MHz and shall employ a minimum resolution bandwidth of 1 MHz. Section 5.8 of Canadian RSS-210 similarly requires the 1 MHz detector bandwidth above 1000 MHz.

A first-order explanation of the functioning of the peak, quasi-peak, and averaging detectors may be of use to the reader. Although averaging and peak detectors require specification to exactly describe, their definitions are as most readers would generally expect. The peak detector has a fast attack and very slow fade and is sometimes also called an “envelope detector.” The fast attack time of the peak detector function may allow for reducing the test time of a broadband test. The averaging detector captures the average power level over a long period of time. One way of generating an approximate average detector is to use a spectrum analyzer with the video bandwidth set much lower than the lowest expected modulation frequency, although the linearity of the spectrum analyzer may limit the accuracy of this measurement.

The quasi-peak detector function requires further explanation. This function was developed to correct for the subjective human response to pulse interference on audio channels. This allows for setting useful limits to sources that may cause interference to analog voice communications or television systems. Human hearing suffers less perceived interference from low pulse repetition frequencies (PRFs); therefore, this function deemphasizes the peak response of lower PRFs by use of a longer attack time constant and a shorter decay time constant than a pure peak detector. For digital communications systems, this function is less useful, but it is still commonly specified and used below 1000 MHz, in both Europe and the United States, because this lower frequency range was historically used for analog voice systems. A pre-detector overload factor (see Table F.2) captures the required linearity of the measurement system. Because the peak power in the measured channel can be much greater than the quasi-peak power, but still has to be correctly reacted to by the measurement system, the system must not compress on peaks that exceed the quasi-peak measurement by the overload factor values shown in Table F.2 in each subband.

Table F.2—CISPR-16 quasi-peak detector specifications

	9–150 kHz	0.15–30 MHz	30–1000 MHz
6 dB bandwidth (kHz)	0.2	9	120
Charge time constant (ms) (attack time)	45	1	1
Discharge time constant (ms) (decay time)	500	160	550
Predetector overload factor (required excess linearity) (dB)	24	30	43.5

Detector bandwidths result in emissions really being in spectral density form rather than in total power form, although for narrowband emissions they are effectively the same. The 902–928 MHz band, due to its 100 kHz detector bandwidth, rather easily benefits from direct sequence spreading that allows higher transmit power than Section 15.249 of FCC CFR47 (the “narrowband rule,” see below in this Clause) directly indicates. For 2400 MHz operation under Section 15.249 of FCC CFR47, the 1 MHz detector bandwidth could also be used in the case of very wide bandwidth systems to allow more total power. This was useful for justifying higher carrier power for nonspread spectrum transmitters before the May 30, 2002, rule change that eliminated the direct sequence requirement for wideband systems. For example, IEEE 802.15.3 high-rate WPAN systems were initially planned to operate under the interpretation of about 1 mW/MHz instead of 1 mW total, allowing an approximately 10 dB increase in total transmit power over what a narrowband-only interpretation of Section 15.249 of FCC CFR47 would have allowed. The 1 MHz detector bandwidth above 1000 MHz is also useful for reducing the harmonic requirements of both 900 MHz and 2400 MHz systems, by taking advantage of the frequency spreading of the harmonics. This provides a relaxing of harmonic requirements for IEEE 802.15.4 systems for both the 2400 MHz and 900 MHz air interfaces, which will be detailed later in this Clause.

The rule in Section 15.35 of FCC CFR47 also allows increases in allowed transmit power (in some cases) or spurious emissions (in all cases) via time averaging. However, in the ISM bands peak carrier powers are specified, so the provisions of Section 15.35 of FCC CFR47 are limited to harmonics and other spurious emissions only. Specifically, Section 15.35 (c) of FCC CFR47 states that averaging of radiated emission limits may be used and that allowed peak emissions may be as much as 20 dB in excess of stated limits so long as the maximum averaging time does not exceed 100 ms. The worst-case 100 ms in a longer transmission must be used. Design formulas and examples relative to IEE 802.15.4 will be given later in this annex. It must be kept in mind that the FCC rules are referring to “emissions” in terms of radiated rms electric field strength, not directly in terms of power. This averaging rule is often helpful in reducing the difficulty in meeting the low general field strength levels given in Section 15.209 of FCC CFR47.

Section 6.5 of Canadian RSS-210 also allows for averaging with the same numerical specifications as Section 15.35 of FCC CFR47. Specifically, the averaging shall apply over the worst-case 100 ms period, and peak power shall not exceed 20 dB more than the allowed average power limit.

F.2.2 Section 15.209 of FCC CFR47

The so-called “general” rule restricts the RF energy that electronic equipment may parasitically emit. The specific level of emissions is 200 uV/m at 3 m test range below 960 MHz and 500 uV/m above. These field strengths are approximately equivalent to –49.2 and –41.2 dBm ERP, respectively. The formula used to make this conversion from root-mean-square (rms) electric field E_{rms} to transmitted ERP P_{terp} at range R m is

$$P_{terp} = 0.03333R^2E_{rms}^2 \tag{F.1}$$

Section 15.209 (d) of FCC CFR47 calls out the same detector functions over frequency as called out in Section 15.35 (b) of FCC CFR47, namely that CISPR quasi-peak detectors are used below 1000 MHz and averaging detectors above 1000 MHz. Thus these field strength levels are actually field strength spectral density rather than total field strength. The densities are per 100 kHz below 1000 MHz and per 1 MHz = above 1000 MHz.

This rule governs the general spurious nonharmonic emissions that IEEE 802.15.4 equipment can emit in the United States, and where harmonics fall into restricted bands, it also limits harmonics. The 2nd and 3rd harmonics of the 2400 MHz ISM band do fall into restricted bands as given in Section 15.205 of FCC CFR47, as do the 3rd and 5th harmonics of the 900 MHz ISM band. Again, the Canadian regulations outlined in Section 6.2.1 of Canadian RSS-210 are nearly identical.

F.2.3 Section 15.205 of FCC CFR47

This clause documents the restricted bands (see Table F.3) where only spurious emissions are allowed and where those emissions must meet the general levels of Section 15.209 of FCC CFR47. Above 1000 MHz, averaging according to Section 15.35 of FCC CFR47 may be used.

Note that the 2nd and 3rd harmonics of equipment in the 2400 MHz ISM band fall into restricted bands, as do the 3rd and 5th harmonics of 902–928 MHz. The restricted band from 2483.5 MHz to 2500 MHz is also a cause for potential concern for close-in spurious emissions from IEEE 802.15.4 transmissions, with the highest center channel setting at 2480 MHz being only 3.5 MHz away.

The list of restricted bands used in Canada are almost identical to the ones shown in the Table F.3. See Table 2 of Canadian RSS-210 to note the differences.

Table F.3—Partial list of restricted frequencies under Section 15.205 of FCC CFR47

Restricted frequency range
240–285 MHz
322–335.4 MHz
399.9–410 MHz
608–614 MHz
960–1240 MHz
1300–1427 MHz
1435–1626.5 MHz
1645.5–1646.5 MHz
1660–1710 MHz
1718.8–1722.2 MHz
2200–2300 MHz
2310–2390 MHz
2483.5–2500 MHz
2655–2900 MHz
3260–3267 MHz
4.5–5.15 GHz
7.25–7.75 GHz
10.6–12.7 GHz

The limit of 500 uV/m at 3 m is approximately –41 dBm/MHz ERP; therefore, for approximately –1 dBm narrowband continuous transmission, at least 40 dB harmonic suppression is required. For the 2400 MHz direct sequence IEEE 802.15.4 system with approximately 1 MHz 3 dB power bandwidth, there is a moderate relaxation of this requirement due to spreading. At the 2nd harmonic, the power bandwidth is approximately 2 MHz; therefore, for a 1 MHz detector bandwidth, the requirement is relaxed 3 dB to yield a –38 dBc harmonic suppression requirement. For a 3 MHz power bandwidth at the 3rd harmonic of 2400 MHz, the relaxation is 4.8 dB (see F.5). For the 900 MHz U.S. (40 kb/s, 600 kchip/s) IEEE 802.15.4 air interface, the 3rd and 5th harmonics fall into these restricted bands. The fundamental 3 dB bandwidth of the 900 MHz PHY is about equal to the chip rate and, therefore, is about 600 kHz. At the 3rd harmonic this allows a relaxation of about 2.5 dB; and at the 5th harmonic, a relaxation of about 4.8 dB. Even with these relaxations, these harmonic requirements are fairly difficult for low-cost equipment; but the use of the averaging rules of Section 15.35 of FCC CFR47 applied to the actual transmit times of IEEE 802.15.4 packets will relax them considerably more. This is quantified in sections 5 and 6.

F.2.4 Section 15.247 of FCC CFR47

Section 15.247 of FCC CFR47 is the primary category for U.S. operations of IEEE 802.15.4 equipment. This service category provides the potential for the highest performance of all the unlicensed service categories, allowing freedom from licensing, transmit powers up to 1 W, and no limitations for application or transmit duty cycle. This rule is applicable to the ISM bands, which are 902 MHz to 928 MHz, 2400 MHz to 2483.5 MHz, and 5725 MHz to 5850 MHz. Of these bands, only the 2400–2483.5 MHz band provides a nearly worldwide available band of suitable power and freedom from applications restrictions for IEEE 802.15.4.

Until May 30, 2002, this service category required either FHSS or DSSS, or a combination. However, at that time the requirement for direct sequence was replaced by digital modulation, which was defined as modulation that satisfied the following two requirements:

a) 6 dB bandwidth of 500 kHz or more
b) Spectral density not to exceed +8 dBm/3 kHz

Note that the 2 Mchip/s O-QPSK waveform of the IEEE 802.15.4 2400 MHz air interface, with 6 dB bandwidth of about 1.5 MHz, and the 600 kc/s BPSK waveform of the 902–928 MHz air interface, with bandwidth of about 600 kHz, both meet the definition of digital modulation.

Allowed transmit power is up to 1 W delivered at the antenna port, with a maximum antenna gain up to 6 dBi. If antenna gain is greater than 6 dBi, then transmit power must be reduced by an amount equal to the decibel measure of how much the antenna gain exceeds 6 dBi. However, for 2400 MHz fixed point-to-point operations, the transmit power need only be reduced by 1 dB for every 3 dB that antenna gain exceeds 6 dBi.

The spurious and harmonic requirement of Section 15.247 of FCC CFR47 is a rather easily achieved –20 dBc, as measured in a detector bandwidth 100 kHz wide and compared to the 100 kHz of the modulated carrier that contains the highest power. However, it must be noted that the 2nd and 3rd harmonics of 2400 MHz and the 3rd and 5th harmonics of 900 MHz fall into restricted bands as given in Section 15.205 of FCC CFR47 and thus have stricter requirements. As discussed above, these requirements can then be mitigated by the 1 MHz detector bandwidth and averaging effects of Section 15.35 of FCC CRF47. The resulting requirements will be quantified in F.5 and F.6.

See Section 6.2.2 (o) of Canadian RSS-210 for the very similar Canadian rules allowing operation in both 902 MHz to 928 MHz and 2400 MHz to 2483.5 MHz. As of the time of this writing (November 2002), these rules did not yet allow for the nondirect sequence digital modulation, but it is likely that a Canadian rule change to bring section 6.2.2 into agreement with Section 15.247 of FCC CFR47 will shortly be made.

F.2.5 Section 15.249 of FCC CFR47

The service category of Section 15.249 of FCC CFR47 provides for narrowband (nonspread spectrum) operation in the same ISM band of 902 MHz to 928 MHz, 2300 MHz to 2383.5 MHz, and 5725 MHz to 5850 MHz. Operation is allowed up to 50 mV rms of electric field strength from a transmitter at a 3 m test range. This is equivalent to 0.75 mW ERP [see Equation (F.1)]. Under Section 15.35 of FCC CRF47, these field strengths are actually interpreted as field strength spectral *densities* and are per 100 kHz for the 902–928 MHz band and per 1 MHz = for the two higher bands. The harmonic requirement is a difficult –40 dBc relative to the maximum allowed power, and the restricted bands can require additional suppression. Fortunately the provisions of Section 15.35 of FCC CRF47 can be applied to reduce these difficult requirements.

IEEE 802.15.4 equipment would not generally be certified under this category, but it could be if the product manufacturer chose to do so. For 902–928 MHz operation, it is not as disadvantageous as it might seem because the interpretation of this section as a transmitted density allows higher 900 MHz powers for IEEE 802.15.4 equipment than for narrowband equipment. Because the detector bandwidth used for certification testing in this band is 100 kHz wide, the 902–928 MHz air interface with about 600 kHz bandwidth could be certified under this service category at up to about +6.5 dBm. However, because the detector bandwidth at 2400 MHz is 1 MHz and the 3 dB bandwidth of the 2400 GHz air interface is also about 1 MHz, the carrier power is still limited to the narrowband limit of about –1 dBm. Concerning harmonics, because the 2nd and 3rd harmonics of 2400 MHz and the 3rd and 5th harmonics of 900 MHz fall into restricted bands anyway, only the 2nd harmonic of 900 MHz really benefits from the relaxed –20 dBc harmonic requirement given in Section 15.247 of FCC CFR47. With that exception, the harmonic requirements of Section 15.247 and Section 15.249 of FCC CFR47 are practically the same because they happen to have restricted band limits for low-order harmonics.

See Section 6.2.2 (m) of Canadian RSS-210 for the 902–928 MHz and 2400–2483.5 MHz narrowband authorizations that are identical to Section 15.249 of FCC CFR47.

F.3 Applicable European rules

The over 40 member nations of the European Conference of Postal and Telecommunications Administrations (CEPT) have established a fairly high degree of standardization throughout Europe on the operation of low power radio equipment. Most disagreements are in the nature of allowed modes and transmit duty cycles that may be accounted for in software control, allowing the same hardware and technical standards to be used throughout Europe. The ETSI develops technical standards for CEPT countries. Requirements are spread across multiple documents, as shown in Table F.4 and are influenced by other documents as well. It is not always clear when European Community advisory and recommendation documents have attained the force of law, so the regulatory requirements of a particular nation must be checked before planning to market equipment there.

Table F.4—Primary European rules documents

Specification Number	Title	Comment
CEPT CISPR 16-1: 1999	Specifications for radio disturbance and immunity measuring apparatus and methods: Part 1: Radio disturbance and immunity measuring apparatus.	Specifications on test methods and equipment, including EMC antennas and the CISPR-16 quasi-peak detector that are referenced, but not provided in other sources. Available from the IEC.
ERC 70-03 [B13]	Relating to the Use of Short Range Devices (SRDs), April 2002	General recommendations that are loosely followed by most European nations.
ETSI EN 300 328-1 [B11]	Electromagnetic Compatibility and Radio Spectrum Matters (ERM); Wideband Transmission Systems; Part 1	European rules for spread spectrum systems in ISM bands. Available from ETSI.
ETSI EN 300 328-2 [B12]	Electromagnetic Compatibility and Radio Spectrum Matters (ERM); Wideband Transmission Systems; Part 2	European rules for spread spectrum systems in ISM bands. Available from ETSI.
ETSI EN 300 220-1 [B10]	Electromagnetic Compatibility and Radio Spectrum Matters (ERM); Short Range Devices (SRDs); Part 1	Provides details on European test methodology to confirm compliance.

The most fundamental document to use as a starting point in understanding general low-power radio operation in Europe is ERC 70-03E [B13] on SRD, downloadable from www.ero.dk. This document gives a general description of recommended applications, frequencies, powers, and other specifications. It provides advisory recommendations for narrowband operation in the 2400 MHz ISM band in the table of Annex 1 and allows up to 10 mW EIRP in narrowband or spread spectrum mode. ERC 70-03E is particularly applicable to IEEE 802.15.4 equipment operating in the European 868.0–868.6 MHz band segment. However, for ISM band spread spectrum operation, such as IEEE 802.15.4 equipment operating at 2400 MHz with bit rates greater than or equal to 250 kb/s, ETSI EN 300 328 is in general the formal governing document (although it may be overruled by the regulatory documents of a specific nation). For devices with less than 10 mW EIRP, the rules of Section 5 of ETSI EN 300 328 may be used or the rules of Annex 1 of ERC 70-03E. For power levels between 10 mW and 100 mW in the 2400 MHz band, only ETSI EN 300 328 applies, and DSSS or FHSS must be used.

Details on test methodology to confirm compliance are given in CISPR 16-1: 1999 and in ETSI EN 300 220-1 [B10]. An important general note is the bandwidths in which carrier and spurious emissions are measured. The specification of bandwidth actually means that what is commonly referred to as power is really PSD, and total power over wider bandwidth than the detector may be higher than the level allowed within a detector bandwidth. Fortunately, the standards chosen are basically the standards also specified in the FCC rules. Table 5 in Section 6.6 of ETSI EN 300 220-1 specifies that below 1000 MHz a test spectrum analyzer bandwidth of 100 kHz shall normally be used, and above 1000 MHz a bandwidth of 1 MHz shall be used. This is in general identical to the provisions of FCC 15.35. However, a slight variance exists in that Table 5 also states that below 25 MHz a bandwidth of 10 kHz shall be used, and below 150 kHz a bandwidth of 1 kHz shall be used. The zone between 150 kHz and 25 MHz is one in which clock and digital noise does have a potential to cause difficulty; therefore, certification laboratories will normally conduct testing over that band. Note that the ETSI EN 300 220-1 frequency boundary for changing detector bandwidths is 25 MHz as opposed to the 30 MHz given in CISPR 16-1: 1999.

An important difference in the European rules as compared to the FCC rules are provisions that go beyond preventing interference to other systems, which is a primary goal of the FCC rules, into attempting to guarantee acceptable system performance. This is captured in law in that most electronic equipment sold in the European Union must comply with EMC Directive 89/336/EEC, and carry the CE mark that claims compliance. After April 8, 2000, compliance with these requirements could be self certified by certain procedures (see www.ero.dk). Another document governing required performance its ETSI EN 300 683. This document’s performance requirements are centered on interference immunity from both outside electromagnetic fields and disturbances on power supply and control inputs. These documents reference other documents as well, the full set of which must be reviewed by manufacturers.

F.3.1 European 2400 MHz band rules

Basic 2400 MHz band parameters extracted from ETSI EN 300 328 are given in Table F.5.

Table F.5—Basic general 2400 MHz band parameters from ETS EN 300 328

Parameter	Specifications	Comments
Carrier Frequency	2400 MHz to 2483.5 MHz	
Transmit Power	100 mW ERP maximum	
Modulation	Frequency hopping: At least 20 channels with 0.4 s maximum dwell time Direct sequence: ERP limited to 10 mW/MHz maximum	In addtion to spread spectrum, data rates greater than or equal to 250 kb/s are required.
Narrowband spurious emissions from 30 MHz to 1 GHz	–36 dBm ERP operating –57 dBm ERP standby	

Table F.5—Basic general 2400 MHz band parameters from ETS EN 300 328 *(continued)*

Parameter	Specifications	Comments
Narrowband spurious emissions from 1 GHz to 12.75 GHz	–30 dBm operating –47 dBm standby	Approximately 10 dB more relaxed on 2nd and 3rd harmonic than U.S. requirement
Narrowband spurious emissions 1.8 GHz to 1.9 GHz and 5.15 GHz to 5.3 GHz	–47 dBm operating –47 dBm standby	
Wideband spurious emissions from 30 MHz to 1 GHz	–87 dBm/Hz ERP operating –107 dBm ERP standby	
Wideband spurious emissions from 1 GHz to 12.75 GHz	–80 dBm/Hz operating –97 dBm/Hz standby	Note that the interpretation of the transmitted modulation sidebands and phase noise out of band must meet these levels.
Narrowband spurious emissions 1.8 GHz to 1.9 GHz and 5.15 GHz to 5.3 GHz	–97 dBm/Hz operating –97 dBm/Hz standby	

Note that the wideband spurious emission may be interpreted so that transmitted phase noise, as spread by the DSSS modulation, must meet –80 dBm/Hz general wideband level at the band edge.

The allowed emission by the receiver portion of the device, which is presumably interpreted as during the receive time of a transceiver, are given in Clause 5.3.2 of ETSI EN 300 328 and documented in Table F.6.

Table F.6—Allowed receiver spurious emissions for 2400 MHz band from ETSI EN 300 328

Parameter	Specification	Comments
Narrowband RX emissions 30 MHz to 1 GHz	–57 dBm	
Narrowband RX emissions 1 GHz to 12.75 GHz	–47 dBm	
Wideband RX emissions 30 MHz to 1 GHz	–107 dBm/Hz	
Narrowband RX emissions 1 GHz to 12.75 GHz	–97 dBm/Hz	

There are country-by-country exceptions to the 2400 MHz rules that change over time. The latest exceptions may be noted by referring to the current revision of ERC 70-03E [B13], Annex 1 Band L Non Specific Short-Range Devices 2400-2483.5 MHz. Other applications that could cause potential interference to IEEE 802.15.4 2400 MHz band operation are noted below.

Railway autoidentification systems in Europe are allocated at 2446 MHz to 2454 MHz and, because they are fairly powerful, could cause interference to IEEE 802.15.4 systems using these frequencies. There are five narrowband channels of 1.5 MHz each over this band, at transmit powers up to 500 mW in the presence of the train. See Annex 4 of ERC 70-03E for more details.

Equipment for detecting movement is authorized in 2400 MHz to 2483.5 MHz at power levels up to 25 mW. See Annex 6 of ERC 70-03E.

F.3.2 European 868–870 MHz band rules

This information is provided in support of the version of the 900 MHz air interface that has been developed for the European 868–870 MHz band. This air interface provides 20 kb/s raw data rate using 300 kc/s and BPSK modulation. The information shown in Table F.7 is for general purpose SRDs and is extracted from Annex 1 of ERC 70-03E and Table 2 of Appendix 1 of ERC 70-03E [B13] .

Table F.7—European 868–870 MHz band rules from Annex 1 of ERC 70-03E [B13] (nonspecific SRDs)

Frequency (MHz)	TX power	Duty cycle	Comment
868.0–868.6	25 mW	< 1%	
868.7–869.2	25 mW	< 0.1%	
869.4–869.65	500 mW	< 10%	25 kHz channel spacing required
869.7–870.0	5 mW	100%	

For the particular case of nonspecific SRDs in the 868.0–868.6 MHz band, ETSI EN 300 220-1 [B10] and ERC 70-03 [B13] apply. The fundamental parameters of this band are summarized in Table F.8.

Table F.8—Fundamental 868.0–868.6 MHz band parameters from ETSI 300 220-1 [B10] and ERC 70-03 [B13]

Parameter	Specification	Comment
Carrier frequency	868.0–868.6 MHz	
Transmit power	25 mW ERP maximum	
Transmit duty cycle	< 1%	In any 1 h period
Maximum TX on time	3.6 s	Advisory only as per Appendix 1 of ERC 70-03E.
Minimum TX off time	1.8 s	Advisory only as per Appendix 1 of ERC 70-03E.
Antenna	Integral or dedicated required	Type approved with the equipment

The 1% Transmit Duty Cycle is the main limitation on operation within the 868.0–868.6 MHz band. However, as stated in Annex E, IEEE 802.15.4 devices are intended for low duty cycle operation. It is the responsibility of the higher protocol layers to ensure that these parameters are satisfied.

The Electronic Communications Committee (ECC) within the CEPT is currently studying a number of changes to the SRD rules. Changes that are generally more friendly to SRD operation are proposed for both 2400 MHz and 868 MHz, but the largest benefit is a likely expansion of the limited 868–870 MHz band down to 863 MHz. This would provide multiple additional channels to the single channel now allowed by the European rules for the European implementation of the IEEE 802.15.4 900 MHz PHY. A broad description of changes proposed at the time of this writing (November 2003) is captured in the report titled “Strategic Plans for the Future Use of the Frequency Bands 862-870 MHz and 2400-2483.5 MHz for Short Range Devices”, May 2002. This report is available at www.ero.dk, under “ECC Activities” and the Frequency Management and Short Range Devices links.

Among the suggested 868 MHz band changes are to extend the band to cover from 863 MHz to 870 MHz for nonspecific SRDs using spread spectrum with power levels up to 25 mW. Some relaxation of the duty cycle limits now applying in 868 MHZ to 870 MHz is contemplated. The use of interference avoidance techniques (e.g., frequency agility, dynamic channel assignment, listen before transmit) are encouraged. The band from 862 MHz to 863 MHz, despite its appearance in the title of the document, is not recommended for SRDs due to its current usage for security services.

The changes for 2400 MHZ to 2483.5 MHz have less impact on IEEE 802.15.4, but one change of note is the possible relaxation of the 250 kb/s minimum data rate to qualify for certification under ETSI EN 300 328. This would bring more low-data-rate devices into the band and increase the probability of interference. However, also proposed are more extensive methods of interference avoidance. These are generally in keeping with the interference avoidance and coexistence methodologies already planned for IEEE 802.15.4.

The reader is advised to remain alert to possible rapid changes in the European rules that may occur after this writing. The above noted website is the best known source of information.

F.4 Known Japanese rules

The Japanese regulatory situation has proven difficult to understand and summarize in depth due to an unfortunate lack of extensive Japanese participation in the early development of the standard, and the resulting language barrier. The following information is summarized from ARIB STD-T66 [B14] in an attempt to be as complete as possible. Although not a direct representation of Japanese regulatory law, this widely used industry standard document is believed by the committee to provide accurate, although incomplete, information. The major known incompleteness is in the areas of detector bandwidth, averaging statistics, and test methodology that can impact the simple data provided. It is also not clear from the standard exactly which specifications carry the force of law and which are recommendations or whether antennas must be permanently fixed to the equipment or have nonstandard connectors (as mandated by the FCC and ETSI).

Table F.9 below summarizes the emissions allowed under ARIB STD-T66 with respect to the 2400–2483.5 MHz band. Other requirements given in that standard include

— Frequency tolerance of ± 50 ppm.
— Maximum antenna gain of 2.14 dB at full power of 10 mW/MHz, although higher gain antennas may be used with a commensurate decrease in maximum transmit power.
— Minimum “spreading factor” of 5, which is most likely another term for processing gain.
— 90% power bandwidth in spread spectrum mode of 500 kHz or more.

F.5 Emissions specification analysis with respect to known worldwide regulations

Based on the above rules, a set of spurious emission requirements that allows meeting the regulations of the majority of the world’s nations may be developed. Where provided, detector specifications and averaging effects may sometimes to be used to relax requirements. Meeting requirements may have a noticeable effect

Table F.9—Japanese 2400 MHz rules as supplied by ARIB STD-T66 [B14]

Emission type	Power level	Comment
DSSS emission 2400–2483.5 MHz	10 mW/MHz +10 dBm/MHz	Detector type and specifications not supplied.
FHSS or FHSS + DSSS 2400–2483.5 MHz	3 mW/MHz +4.77 dBm/MHz	
FHSS or FHSS + DSSS 2400–2483.5 MHz excluding 2427–2470.75 MHz	10 mW/MHz +4.77 dBm/MHz	
Nonspread spectrum 2400–2483.5 MHz	10 mW +10 dBm	
Spurious emissions 2387–2400 MHz and 2483–2496.5 MHz	25uW or less –16.02 dBm or less	“Average” applicable to close in modulation sidebands and phase noise. Averaging statistics and detector bandwidth not given, but presumaby could be relaxed by averaging if these become known. Without detector bandwidth, phase noise requirements cannot be accurately inferred.
Spurious emissions *f* < 2387 MHz and *f* > 2496.5 MHz	2.5 uW or less –26.02 dBm or less	“Average” applicable to far out spurs such as harmonics. Averaging statistics and detector bandwidth not given, but presumaby could be relaxed by averaging if these become known.
RX emissions *f* < 1000 MHz	4 nW –53.98 dBm	Applicable to receive mode spurs such as local oscillator leakage.
RX emissions *f* > 1000 MHz	20 nW –46.99 dBm	

on build of materials (BOM) cost and design choices; therefore, best efforts have been made to infer minimum requirements that support the lowest possible manufacturing cost.

F.5.1 General analysis and impact of detector bandwidth and averaging rules

The use of averaging and detector bandwidths that allow some relaxation of specifications is described well in the FCC rules and similarly in the Canadian rules. European rules do not allow for averaging, but do provide for detector bandwidths that provide some relaxation also. Japanese detector specifications and averaging allowances are unknown to the committee at the time of this writing. The U.S. rules as related to the restricted bands as described in Section 15.205 of FCC CFR47 appear to be the most restrictive, although analysis using the detector and averaging provisions of Section 15.35 of FCC CFR47 with the parameters of the transmitted data leads to the conclusion that European limits are usually the worst case.

The critical facts relative to the U.S. rules are as follows:

- The allowed ERP of spurious emissions (including harmonics falling in restricted bands) is approximately –41 dBm.
- The detector bandwidth used for certification testing is 1 MHz, and the 3 dB modulation bandwidth at the 2400 MHz fundamental planned under the standard is slightly less than 1 MHz.
- The averaging provisions of Section 15.35 of FCC CRF47 apply, which allow averaging over a 100 ms period with maximum increase in peak power limited to 20 dB. The units of measure are

electric field strength; and although rms is not explicitly stated, rms units are standard for FCC measurements.

The mathematics of FCC duty cycle averaging are not given in the regulations, but may be quickly derived, keeping in mind that units are rms electric field strength (not power). The following definitions are made:

D_c = highest carrier duty cycle over a 100 ms period.

E_{ss} = field strength steady state = allowed rms field strength at a particular frequency without averaging.

E_{pa} = field strength peak allowed when averaging = allowed peak field strength at a particular duty cycle and frequency when averaging. Note that according to FCC convention this “peak” is not the true RF peak. It is the rms carrier strength in volts per meter at the peak of the envelope.

$\mathrm{P_{ss}}$ = steady-state power allowed when not averaging.

$\mathrm{P_{pa}}$ = peak power allowed when averaging.

The duty cycle is reduced by an average of -3 dB for amplitude shift key (ASK) modulation, but for the 2400 MHz air interface a nearly constant envelope form of modulation (O-QPSK) is used. Thus duty cycle is simply time on a particular channel relative to 100 ms:

$$D_c = \frac{T_{oc}}{0.1} \tag{F.2}$$

The maximum spurious field peak field strength allowed under the rules will then be

$$E_{pa} = \frac{E_{ss}}{D_c} \tag{F.3}$$

This value is the maximum allowed peak of the envelope of a carrier measured in rms electric field strength, up to a limit of 10 times the steady state allowed rms field (E_{ss}).

Equation (F.3) can be solved for the allowed duty cycle given peak field strength that a system is capable of and put it into power terms as follows:

$$D_C = \frac{E_{ss}}{E_{pa}} = \sqrt{\frac{P_{ss}}{P_{pa}}} = \sqrt{\frac{AllowedPowerSteadyState}{PeakPowerAllowed}} \tag{F.4}$$

For peak power allowed when averaging using narrowband emitters (bandwidth less than regulatory detector bandwidth), compute

$$P_{pa} = \frac{P_{ss}}{D_c^2} \qquad \text{(narrowband emitters)} \tag{F.5}$$

where the minimum value of D_c^2 is 0.01 (maximum boost of 20 dB when duty cycle = 10% or less).

If the protocol is less than 100% duty cycle over 100 ms, then the averaging effect illustrated in Equation (F.5) may be used to reduce restricted band harmonic attenuation requirements, where P_{pa} is understood to mean peak harmonic power allowed during carrier on times and P_{ss} is the steady state allowed power level in the detector bandwidth. The 500 uV/m allowed steady-state level in restricted bands corresponds to 7.5 x 10^8 W ERP, or –41.2 dBm ERP, which may be a significant level of attenuation to attain in low-cost equipment. A 50% duty cycle over 100 ms could, for example, reduce this difficult requirement by 6 dB.

Beyond averaging, additional help on the required harmonic attenuation levels for restricted band harmonics results from taking the transmitted bandwidth BW_h at the harmonic frequency and regulatory compliance detector bandwidth BW_d (1 MHz above 1000 MHz, and 100 kHz below 1000 MHz) into account. This case of wider band emissions in Equation (F.5) may be extended to

$$HP_{pa} = M\frac{P_{ss}}{D_c^2} \quad \text{(wideband and narrowband emitters)} \tag{F.6}$$

Here HP_{pa} is harmonic power peak allowed and refers to the total harmonic power and not necessarily the harmonic power captured within the detector bandwidth. M is a factor given by

$$M = \begin{cases} 1 & \text{if } \mathrm{BW_h} <= \mathrm{BW_d} \\ \dfrac{BW_h}{BW_d} = \dfrac{hBW_f}{BW_d} & \text{if } \mathrm{BW_h} > \mathrm{BW_d} \end{cases} \tag{F.7}$$

where

h is the harmonic number,
BW_f is the effective power bandwidth (approximately the 3 dB bandwidth) at the fundamental.

The total harmonic peak power allowed of Equation (F.6) may be expressed in decibels relative to 1 mW where duty cycle D_c is between 0.1 and 1.0 as given below:

$$HP_{pa(dBm)} = 10\log M - 10\log D_c^2 - 41.2 \tag{F.8}$$

When duty cycle is less than 10%, the total peak harmonic power allowed in decibels relative to 1 mW ERP is given by

$$HP_{pa(dBm)} = 10\log M - 21.2 \tag{F.9}$$

These results may be used to calculate the total harmonic ERP for IEEE 802.15.4 packets as given in F.5.2.

F.5.2 Frequency spreading and averaging effects specific to IEEE 802.15.4

This subclause is primarily applicable to FCC rules in the United States, but there is also a modest relaxation of European harmonic requirements that may be inferred from detector bandwidth and direct sequence spreading. The United States has more relaxed harmonic requirements except when harmonics fall in restricted bands, in which case, the requirements are then more difficult than Europe, but are then mitigated by the averaging allowances of Section 15.35 of FCC CFR47. Analysis to be shortly reviewed will show that the resulting U.S. requirement is about –20 dBm and is thus approximately 10 dB more relaxed than the –30 dBm European requirement.

Under Section 15.247 of FCC CFR47, harmonics that do not fall in restricted bands have a requirement of –20 dBc as measured in a 100 kHz detector bandwidth when compared to the 100 kHz segment within the fundamental transmission that has the highest power. Because a smaller frequency at the fundamental spreads out to provide the 100 kHz at the harmonic, the requirement is actually even more relaxed that the –20 dBc would indicate. For example, at the 2nd harmonic, if the transmitter harmonic level and filtering effectiveness for a narrowband transmission provided –17 dBc, then once well spread with DSSS, only 50 kHz at the fundamental will spread out to 100 kHz at the 2nd harmonic. The –17 dBc that the transmitter gives narrowband would be increased to –20 dBc when spread. A formula giving the narrowband transmitter harmonic performance that will meet the –20 dBc requirement when spread, where h is the harmonic number, is

$$FCCNarrowBandHarmReq(dBc) = 20 - 10\log h \tag{F.10}$$

These relaxed harmonic requirements are part of the attractiveness of DSSS. Unfortunately, low-order harmonics of both 900 MHz and 2400 MHz fall into restricted bands where they must meet the much more stringent requirements of Section 15.209 of FCC CFR47. However, as described in F.5.1, the detector bandwidth and averaging allowances of Section 15.35 of FCC CFR47 may be applied to relax what would otherwise be quite onerous requirements for low-cost equipment. This is done through the application of Equation (F.6) through Equation (F.9) and the general character of the phase shift key (PSK) modulation used in this standard. Filtered BPSK generally has a 3 dB bandwidth about equal to the data rate or to the chip rate in the case of DSSS. O-QPSK generally has a 3 dB bandwidth of about half the data or chip rate. The 2400 MHz air interface with a 2 Mchip/s chip rate has approximately a 1 MHz 3dB fundamental power bandwidth. The 2nd, 3rd, and 5th harmonics of 2400 MHz fall into restricted bands where the general limit is –41.2 dBm ERP. The 900 MHz IEEE 802.15.4 air interface uses 600 kc/s BPSK; therefore, has about a 600 kHz 3 dB bandwidth. The 3rd and 5th harmonics fall into restricted bands. Both bands benefit from the spreading of harmonics beyond the 1 MHz detector bandwidth used above 1 MHz; thus M > 1 applies. Time averaging may be combined with frequency spreading via Equation (F.6), Equation (F.8), and Equation (F.9) to further reduce harmonic requirements if a particular application can guarantee transmissions less than 100 ms. The packet lengths used in IEEE Std 802.15.4-2003 do allow transmissions of less than 100 ms with low duty cycle. The packet length varies depending on the data carried, with a payload that may vary from 0 to 127 bytes, always prefixed with 6 preamble and header bytes. The 2400 MHz PHY has a maximum packet time of 4.256 ms, and the 915 MHz PHY has a maximum packet time of 26.6 ms.

Table F.10 captures the *total* harmonic power in decibels relative to 1 mW ERP that may be transmitted for the two U.S. PHYs for harmonics that land in the restricted bands. The cases shown are for maximum length packets with 127 bytes of data and for nominal length packets with 40 bytes of data.

Table F.10—Harmonic allowed worst cases interpreted from FCC rules and IEEE 802.15.4 duty cycles

PHY	Harmonic/M	Dc maximum/nomimal packet	Allowed harmonic ERP for maximum packet (dBm)	Allowed harmonic ERP for nominal packet (dBm)
900 MHz	3/1.8	0.266/0.92	–27.1[a]	–18.6[a]
900 MHz	5/3	0.266/0.92	–24.9[a]	–16.4[a]
2400 MHz	2/2	0.04256/0.01472	–18.2[b]	–18.2[c3]

Table F.10—Harmonic allowed worst cases interpreted from FCC rules and IEEE 802.15.4 duty cycles *(continued)*

PHY	Harmonic/M	Dc maximum/nomimal packet	Allowed harmonic ERP for maximum packet (dBm)	Allowed harmonic ERP for nomimal packet (dBm)
2400 MHz	3/3	0.04256/0.01472	–16.4[b]	–16.4[3]
2400 MHz	5/5	0.04256/0.01472	–14.2[b]	–14.2[3]

[a] The 900 MHz harmonic powers assume a single packet per 100 ms. The maximum packet time is 26.6 ms.

[b] The 2400 MHz harmonic power for maximum length packets applies up to 2 packets per 100 ms. Because the maximum packet time is 4.256 ms, the 2 packet case still meets the less-than-10% duty cycle requirement for maximum boost of 20 dB in harmonic relaxation requirements.

[c] The 2400 MHz harmonic power for nominal length packets applies up to 6 packets per 100 ms. Because the nominal packet time is 1.472 ms, the 6 packet case still meets the less-than-10% duty cycle requirement for maximum boost of 20 dB in harmonic relaxation requirements.

When these relaxations are taken into account, it must also be considered that whip antennas will generally re-resonate on odd harmonics and show a higher directivity than they do at the fundamental. Several decibels of safety margin must be left to deal with this factor. It must also be understood that, in giving the results of Table F.10 in total harmonic power, less harmonic power will be measured when using the standard measurement bandwidth of 1 MHz, due to the fact that the harmonics of the DSSS transmission are greater than 1 MHz. The factor 10logM for each harmonic gives the approximate decibel measure of how much less the harmonic measured at the center of the transmitted harmonic is than the total harmonic power.

Note that the European requirement for harmonics of 2400 MHz is –30 dBm in a 1 MHz detector bandwidth, with no provision for time averaging. The actual harmonic requirement of the transmitter system is similar to Equation (F.10) and may be written as

$$EuropeanNarrowBandHarmReq(dBm) = -30 + 10\log h \tag{F.11}$$

To use this equation, one turns on a steady state carrier without modulation or chipping and measures harmonic performance either via cable or via open air link (taking antenna aperture into account). If the measured performance is better than the requirement given in Equation (F.11), then the system should pass.

In general the European requirement for 2400 GHz is about 10 dB better than the FCC requirements, taking restricted bands and averaging into account; therefore, a product that passes European harmonic requirements should have a comfortable margin for the U.S. market.

F.6 Summary of out-of-band spurious emissions limits

Table F.11 summarizes the allowed spurious emissions for both U.S. and European limits. The levels below 30 MHz are provided because radiation from clocks and other digital noise sources can sometimes exceed allowed levels. Known Japanese limits were given in Table F.9.

All FCC emissions are steady-state limits and, when above 1000 MHz, could be relaxed via 15.35 time averaging and detector bandwidth frequency spreading. For European operation, similar detector bandwidths apply, but not the time averaging effects. The –30 dBm European harmonic requirement, relaxed as given in Equation (F.11), provides a worst-case level of required harmonic performance that is in general about 10 dB more stringent that the U.S. requirements when analyzed in light of Section 15.35 of FCC CFR47.

Table F.11—Out-of-band spurious emissions power limits for the United States and Europe

Frequency band or harmonics	FCC	ETSI active mode	ETSI standby mode
9–490 kHz	2400/kHz uVrms/m/100 kHz @300 m		
490kHz–1.705 MHz	2400/kHz uVrms/m/100 kHz @30 m		
1.705–30 MHz	30 uVrms/m/100 kHz@30 m –45.7 dBm EIRP		
30–88 MHz	100 uVrms/m/100 kHz@ 3 m –55.2 dBm EIRP	–30 dBm EIRP/100 kHz	–57 dBm EIRP/100 kHz
88–216 MHz	150 uVrms/meter100 kHz @ 3 m –51.7 dBm EIRP	–30 dBm EIRP/100 kHz	–57 dBm EIRP/100 kHz
216–960 MHz	200 uVrms/m/100 kHz @ 3 m –49.2 dBm EIRP	–30 dBm EIRP/100 kHz	–57 dBm EIRP/100 kHz
960–1000 MHz	500 uVrms/m/100 kHz @ 3 m –41.2 dBm EIRP	–30 dBm EIRP/100 kHz	–57 dBm EIRP/100 kHz
1GHz–12.5 GHz	500 uVrms/m/MHz@ 3 m –41.2 dBm EIRP	–30 dBm EIRP/MHz	–47 dBm EIRP/MHz
1.8–1.9 GHz and 5.15–5.3 GHz	500 uVrms/m/MHz@ 3 m –41.2 dBm EIRP	–47 dBm EIRP/MHz	–47 dBm EIRP/ MHz

F.7 Phase noise requirements inferred from regulatory limits

Regulatory spurious emissions limits with known detector bandwidths may be inferred to set a phase noise limit on the transmitter carrier generating synthesizer system.

For European 2400 MHz operation the wide spurious limit under Section 5.2.4 of ETSI EN 300 328 is –80 dBm/Hz from 1 GHz to 12.5 GHz. Because the nearest channel to the band edge in the 2400 MHZ to 2483.5 MHz band is 2480 MHz, this rule sets a worst-case phase noise at 3.5 MHz offset of –80 dBm/Hz. This is converted to phase noise relative to the carrier by taking into account the direct sequence spreading and the maximum carrier power. The phase noise is relaxed by the spreading gain (about 9 dB for the 2400 MHz air interface), allowing a maximum unspread phase noise of –71 dBm/Hz at 3.5 MHz offset. This specification must then improve 1 dB for each decibel the maximum transmitted power can exceed 0 dBm. If the maximum transmit power is +10 dBm, the 2400 MHz product marketed in Europe must show a total phase noise at 2400 MHz before spreading of about –81 dBc/Hz at 3.5 MHz offset. This is a relaxed phase noise requirement, and it will be shown below that U.S. rules will set a worst case for the 2400 MHz air interface.

For the European 868 MHz band operation, the transmit center frequency is 868.3 MHz. Neglecting the slight frequency error of the crystal timebase, the allowed unspread transmitted phase noise is thus –80 dBm/Hz at 300 kHz. The spreading gain of the 20 kb/s, 300 kc/s BPSK 868 MHz air interface is about 15 dB (300 kHz RF bandwidth divided by 10 kHz baseband bandwidth). The allowed phase noise relative to carrier for a 0 dBm transmit power is thus about –65 dBc/Hz at 300 kHz offset. For a +10 dBm maximum transmit power, the required unspread phase noise of the 868 MHz air interface is thus about –75 dBc/Hz at 300 kHz offset. Because the 868 MHz offset is so much less, this requirement is more difficult than the –81 dBc/Hz at 3.5 MHz requirement of the 2400 MHz air interface. However, this level is still fully within the achievable performance of an integrated design.

U.S. requirements on phase noise for 2400 MHz are more restrictive, although still highly attainable with a fully integrated design. There is no general “wideband” limit, only the spurious levels appropriate to a particular service and the general level over the restricted bands. However, there is a restricted band from 2483.5 MHz to 2500 MHz, and the general level of 500 uV/m at 3 m in a 1 MHz detector bandwidth applies. The emission level allowed there is thus –41.2 dBm/MHz ERP or an average of –101.2 dBm/Hz. The spreading gain of 9 dB raised the allowed transmitted phase noise for a 0 dBm transmission to about –92 dBm/Hz at 3.5 MHz offset (the highest channel is 2480 MHz). At +10 dBm, the phase noise requirement is thus about –102 dBc/Hz at 3.5 MHz offset.

For the U.S. 902–928 MHz band the nearest restricted band edge is 960 MHz, the same frequency where the general level rises from 200 to 500 uV/m. The detector bandwidth used is 100 kHz, so the transmitted phase noise is limited to about –41 dBm/100 kHz, or –91 dBm/Hz, at 32 MHz offset. The processing gain for the U.S. 900 MHz band is about 15 dB; therefore, the transmitted phase noise for a 0 dBm transmitter is limited to about –76 dBm/Hz at 32 MHz offset, or –86 dBc/Hz for a +10 dBm transmit power level. This requirement is very relaxed.

Examining the effect of general spurious requirements in the United States (Section 15.247 (c) of FCC CFR47), the spurious limit that falls out of band but not in the restricted bands is –20 dBc/100 kHz, as compared to the 100 kHz segment in band with the most power. Because the comparison uses the same 100 kHz bandwidth, when interpreted as phase noise the limit is –20 dBc at the band edge. This level is too relaxed to set a worst case and may thus be ignored.

The Japanese phase noise requirement cannot be determined without knowledge of test detector bandwidths used to meet the close-in spurious requirements of ARIB STD-T66 [B14]. However, if the detector bandwidth above 1000 MHz is the commonly used international standard of 1 MHz, then an accurate estimate can be made. The Japanese requirement from 2483.5 MHz to 2496.5 MHz is –16.02 dBm. In a 1 MHz bandwidth, this would translate to a phase noise of –76 dBm/Hz at 3.5 MHz offset, reduced to –67 dBm by the spreading gain. Because it would take a detector bandwidth of over 100 MHz for the Japanese requirement to be more severe than the U.S. one, the U.S. requirement almost certainly remains the worst case.

All the phase noise numbers given above are slightly conservative in that they assume a flat phase noise over the detector bandwidth, whereas the actual phase noise continues to decline over the detector bandwidth.

In summary, for 2400 MHz operation, the U.S. rules set a worst case for unspread carrier phase noise of about –102 dBc/Hz at 3.5 MHz offset for a +10 dBm transmit power level. For the 900 MHz air interfaces, the European rules set a worst case of about –75 dBc/Hz at 300 kHz offset for a +10 dBm power level.

F.8 Summary of transmission power levels

Table F.12 summarizes the known maximum transmit power levels for the targeted frequency bands in various geographical regions.

Table F.12—Maximum transmit power levels

Frequency band	Geographical region	Maximum conductive power/ radiated field limit	Regulatory document
2400 MHz	Japan	10 mW/MHz	ARIB STD-T66 [B14]
	Europe (except Spain and France)	100 mW EIRP or 10 mW/MHz peak power density	ETSI EN 300 328
	United States	1000 mW	Section 15.247 of FCC CFR47 [B14]
	Canada	1000 mW (with some limitations on installation location)	GL-36 [B15]
902–928 MHz	United States	1000 mW	Section 15.247 of FCC CFR47 [B14]
868 MHz	Europe	25 mW	ETSI EN 300 220 [B10]

Annex G

(informative)

Bibliography

G.1 General

[B1] IEEE 100™, *The Authoritative Dictionary of IEEE Standards Terms*, Seventh Edition. [8]

[B2] IEEE Std 802.11b-1999, IEEE Standard for Information Technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific Requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band.

[B3] IEEE P802.15, Draft Standard for Information Technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific Requirements—Part 15: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for High Rate Wireless Personal Area Networks (WPAN).

[B4] IEEE Std 802.15.1-2002, IEEE Standard for Information Technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific Requirements—Part 15.1: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs).

[B5] IEEE Std 802.15.2-2003, IEEE Standard for Information Technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific Requirements—Part 15.2: Coexistence of Wireless Personal Area Networks with Other Wireless Devices Operating in Unlicensed Frequency Bands.

[B6] IEEE Std 802.15.3-2003, IEEE Standard for Information Technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific Requirements—Part 15.3: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for High Rate Wireless Personal Area Networks (WPAN).

[B7] IEEE Std 802.16-2001, Information Technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific Requirements—Part 16: Air Interface for Fixed Broadband Wireless Access Systems.

[B8] Sklar, Bernard, *Digital Communications: Fundamentals and Applications*. New Jersey: Prentice Hall, 1988.

G.2 Regulatory documents

[B9] ARIB STD-T66, Second Generation Low Power Data Communication System/Wireless LAN System 1999.12.14 (H11.12.14) Version 1.0. Association of Radio Industries and Businesses (ARIB). Japan.[9]

[8] IEEE publications are available from the Institute of Electrical and Electronics Engineers, Inc., 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA (http://standards.ieee.org/).

[9] ARIB publications are available at (http://www.arib.or.jp).

[B10] ETSI EN 300 220-1, Electromagnetic Compatibility and Radio Spectrum Matters (ERM); Short Range Devices (SRDs); Radio equipment to be used in the 25 MHz to 1 000 MHz frequency range with power levels ranging up to 500 mW; Part 1: Technical characteristics and test methods.[10]

[B11] ETSI EN 300 328-1, Electromagnetic Compatibility and Radio Spectrum Matters (ERM); Wideband Transmission systems; Data transmission equipment operating in the 2,4 GHz ISM band and using spread spectrum modulation techniques; Part 1: Technical characteristics and test conditions.

[B12] ETSI EN 300 328-2, Electromagnetic Compatibility and Radio Spectrum Matters (ERM); Wideband Transmission systems; Data transmission equipment operating in the 2,4 GHz ISM band and using spread spectrum modulation techniques; Part 2: Harmonized EN covering essential requirements under article 3.2 of the R&TTE Directive.

[B13] ERC Recommendation 70-03, Relating to the use of Short Range Devices (SRDs), April 2002.[11]

[B14] FCC Code of Federal Register (CFR), Part 47, Section 15.35, Section 15.205, Section 15.209, Section 15.231, Section 15.247, and Section 15.249. United States.[12]

[B15] Industry Canada (IC) Document: GL-36. Canada.[13]

[10] ETSI publications are available from the European Telecommunications Standards Institute (http://etsi.org).

[11] ERC publications are available from Global Engineering Documents, 15 Inverness Way East, Englewood, Colorado 80112, USA (http://global.ihs.com/).

[12] FCC publications are available from Global Engineering Documents, 15 Inverness Way East, Englewood, Colorado 80112, USA (http://global.ihs.com/).

[13] IC publications are available from Global Engineering Documents, 15 Inverness Way East, Englewood, Colorado 80112, USA (http://global.ihs.com/).