Proposed changes for addressing comments #12, #19, #20, #25 and #26 against P802.1DU/D0.3

Author: Johannes Specht

Date: 2024-09-11

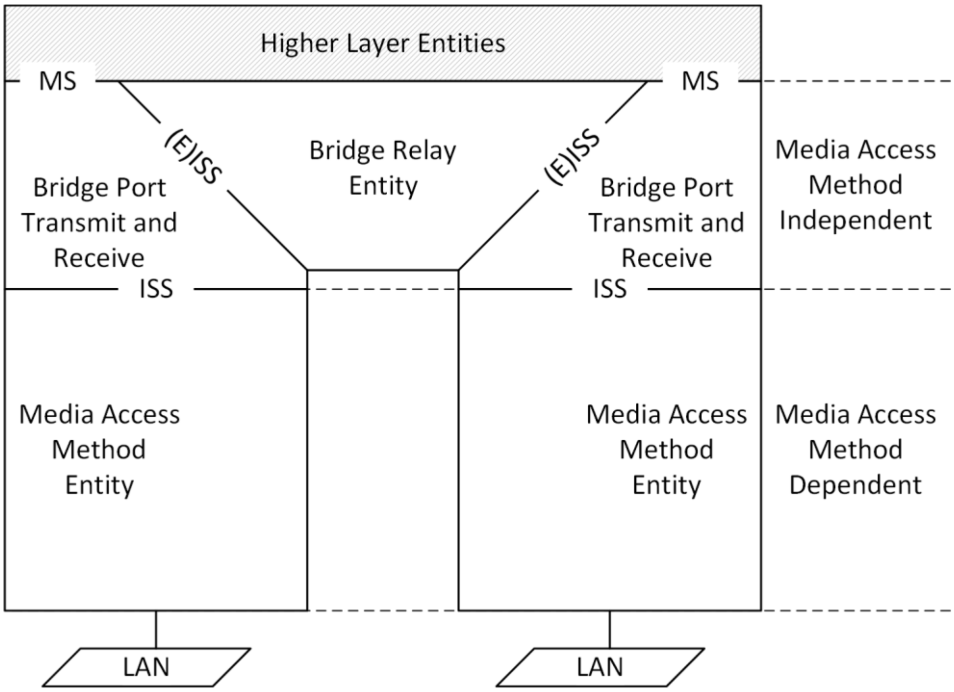
Introduction

This document contains instructions for addressing comments #12, #19, #20, #25 and #26. The instructions contain additional enhancements that were discovered during creation of this document. These enhancements include the following:

1. Clearly introduce the two invocation models used in P802.1DU.
2. Introduce “invocation intervals” for progressive invocations, in order to simplify temporal descriptions and eliminating ‘end and ‘start. The latter were helpful when describing MAM-dependent operations by state machine diagrams in IEEE 802 Nendica, but this is no longer needed in P802.1DU.
3. Remove the frame handle logic. It is only relevant for frames under transmission that originate from a frame under reception. Verbal description is sufficient here.
4. Some additional errors fixed.

Comments #19, #20

*** Replace Figure 6-1 with the following:**


Higher Layer Entities
MS
MS
(E)ISS
(E)ISS
Bridge Relay Entity
Bridge Port Transmit and Receive
Bridge Port Transmit and Receive
Media Access Method Independent
ISS
ISS
Media Access Method Entity
Media Access Method Entity
Media Access Method Dependent
LAN
LAN

*** Rename clause 6 to “CTF Bridge Architecture”**

*** In 10.1, introduce that the relay specification effectively limits to the forwarding process and retains other elements of the relay from 802.1Q unaltered.**

Comments #12, #25, #26

*** Introduce term "Media Access Method (MAM) *entity*" as the per Port share of a Media Access Method (cmp. P802Rev-C), and replace all mentionings of MACs by MAM/MAM entity.**

*** Change M_LATEERROR.indication to M_LATEERROR.ingress, and M_LATEERROR.request to M_LATEERROR.egress throughout the entire document.**

*** Replace lines 2 through 22 on page 37 as follows:**

allowing for accurate description of operations within a CTF Bridge when necessary.

Therefore, service primitive invocations follow two different temporal models:

a) Progressive invocations
b) Instantaneous invocations

Progressive invocations use a detailed temporal model, where an invocation happens over a specific time interval associated with this invocation, denoted as *invocation interval*. The information in service primitive parameters is progressively completed during that interval. In a series of sequential processing stages (e.g., a sub-process of the forwarding process in 10.1), this model allows later processing stages to access information in service primitive parameters that is progressively added by an earlier processing stage. The invocation interval can be in a temporal relationship with the externally visible time interval of the associated frame on the individual LAN (see 8.3 and Figure 7-1).

Instantaneous invocations correspond to a simplified temporal model that omits the notion of progressive parameter completion and treats invocations as instantaneous events. This model is sufficient when all parameters are complete instantaneously before an invocation.

*** Add the following definition after line 10 on page 28:**

Invocation interval: A time interval corresponding to a service primitive invocation.

*** Change the paragraph starting in line 36 on page 38 to the following:**

A parameter is said to be complete at the earliest instant of time in the invocation interval progressive service primitive invocation at which the minimal information is available to unambiguously determine the parameter's value within the valid value range of such parameter. High-level parameters derived from information contained in service primitive parameters of unit-data indication invocations are completed during the invocation interval. High-level parameters solely based on out-of-band information are completed at the start the invocation interval.

*** Replace the current M_LATERERORservice primitives by an additional parameter, lateerror, in Unit-Data indications and requests as a functional equivalent replacement of the former. The new parameters can take the three values NO_ERROR, NEW_ERROR and KNOWN_ERROR to indicate the late error status during frame transmission and reception.**

*** Change lines through 24 on page 42 to the following:**

Invocations of M_UNITDATA.indication and M_UNITDATA.request are progressive invocations (7.3) with the following temporal definitions:

d) The invocation interval of M_UNITDATA.indication invocations overlaps with the externally visible time interval of the associated frames under reception (Figure 7-1). It starts during this visible interval and ends no earlier than it.

e) The invocation interval of M_UNITDATA.request invocation overlaps with the externally visible time interval of the associated frames under transmission (Figure 7-1). The external visible interval starts no earlier than that of the invocation. The invocation interval ends no earlier than that of potential M_UNITDATA.indication invocation from a different MAM entity.

*** On page 42, delete lines 25 through 33, line 38 and line 42.**

*** Change the paragraph in lines 24ff. on page 45 to the following:**

For frames under reception, the invocation interval for M_UNITDATA.indication towards the Bridge relay entity starts when the frame is passed to the Bridge relay entity according to the aforesaid definitions and ends with that of the associated M_UNITDATA.indication invocation from the CTF sublayer.

*** Change the paragraph in lines 10ff. on page 46 to the following (‘end):**

For frames under reception, the invocation interval for EM_UNITDATA.indication towards the Bridge relay entity starts when the frame is passed to the Bridge relay entity according to the aforesaid definitions and ends with that of the associated M_UNITDATA.indication invocation from the CTF sublayer.

*** Change the paragraph in lines 6ff. on page 47 to the following:**

If CTFReceptionEnable is FALSE and CTFReceptionSupported is TRUE, processing of frames under reception falls back to S&F prior to instantaneously passing complete frames towards the Bridge relay.

*** Change the contents of 9.6.2 to the following:**

The transmit path of the CTF sublayer passes frames from the Bridge relay entity towards underlying the MAM entity instantaneously.

A particular frame results in a progressive M_UNITDATA.request invocation (7.3) to the MAM entity if that frame corresponds to a frame under reception (i.e., Cut-Through) and the underlying MAM entity for transmission supports CTF (8.3).

During that invocation interval, the transmit path of the CTF observes the late error status of the frame under reception. Any late error that is associated with that frame during the invocation interval of the M_UNITDATA.request invocation is passed via the lateerror parameter of the M_UNITDATA.request invocation towards the MAM entity instantaneously.

*** Remove frame handles (the aforesaid verbal description creates the association)**