802.3 NEA CTF:
CTF concerns

Peter Jones – Cisco

April 27, 2022

Background: P802.1DU PAR/CSD

- The PAR/CAD was presented in the March 2022, Virtual Plenary.
  - https://ieee802.org/1/files/public/docs2022/du-draft-PAR-0122-v01.pdf
  - https://ieee802.org/1/files/public/docs2022/du-draft-CSD-0122-v01.pdf
- 802.3 provided feedback at
  - https://www.ieee802.org/3/minutes/mar22/2022_03%20PARs%20ad%20hoc%20report.pdf
- Major issues
  - Slide 6 – Compatibility – 1.2.2 h)
    - A CTF Bridge will not be compliant with the published standards in question (i.e., IEEE Std 802.1AC and IEEE Std 802.1Q)
  - Slide 8 - Economic Feasibility 1.2.5 p)
    - CTF bridges violate the 802.3 MAC Service interface where a transmission request is an atomic action.

Notes

- The following slides are mostly extracts from 802.1AC and 802.3 showing areas I think are issues for CTF.
- This is not an exhaustive list.
- I use ***bold italic*** highlights to pick out key points.
- I use "**Calabri**" for commentry.

Slide 6 – Compatibility – 1.2.2 h

- CSD response
  - A CTF Bridge will *not be compliant* with the published standards in question (i.e., IEEE Std 802.1AC and IEEE Std 802.1Q
- Non-compliance to IEEE Std 802.1AC is non-compliance to the Media Access Control (MAC) Service Definition as defined in
  - 7.2 Service interface primitives, parameters, and frames
  - 11. Internal Sublayer Service
  - 13.1 Ethernet convergence function

802.1AC: 7.2 Service interface primitives, parameters, and frames

- "each primitive corresponding to *an atomic interaction* between the (N)-service user and the (N)-service provider"
  - "atomic interaction" is a single uninterruptable operation.
  - Related definitions
    - "An atomic transaction is ***an indivisible*** and irreducible series of database operations such that either all occurs, or nothing occurs." (https://en.wikipedia.org/wiki/Atomicity_(database_systems))
    - "An atomic operation is an operation that will always be executed without any other process being able to read or change state that is read or changed during the operation." (https://wiki.osdev.org/Atomic_operation)
  - "4A.2 Media access control (MAC) method: precise specification" uses the term "synchronous", e.g. "The ReceiveFrame operation is synchronous."
    - Synchronous programming is a programming model where operations take place sequentially. <snip> This linear behavior means that long-running operations are "***blocking***," and the program will ***halt for the duration it takes to complete them***." (https://deepsource.io/glossary/synchronous-programming/)

802.1AC: 11.1 Service primitives and parameters

- Each M_UNITDATA indication corresponds to the receipt of *an error-free MAC frame* from a LAN
- Primitive parameters
  - The mac_service_data_unit parameter is the *service user's data.*
  - The frame_check_sequence parameter is *explicitly provided with the M_UNITDATA.indication* so that it can be used in a related M_UNITDATA.request.

A frame with an error (e.g., FCS, TooLong) does not cause an indication to be generated.

802.1AC: 13.1 Ethernet convergence function

- When the convergence function receives an M_UNITDATA.request primitive, it generates a corresponding MA_DATA.request to the underlying MAC Service as follows:
  - The mac_service_data_unit is processed as follows. IEEE Std 802.3 requires that transmitted frames have a *64-octet minimum length* (3.2.8 of IEEE Std 802.3-2015), including the destination_address, source_address, *mac_service_data_unit, and frame_check_sequence.*

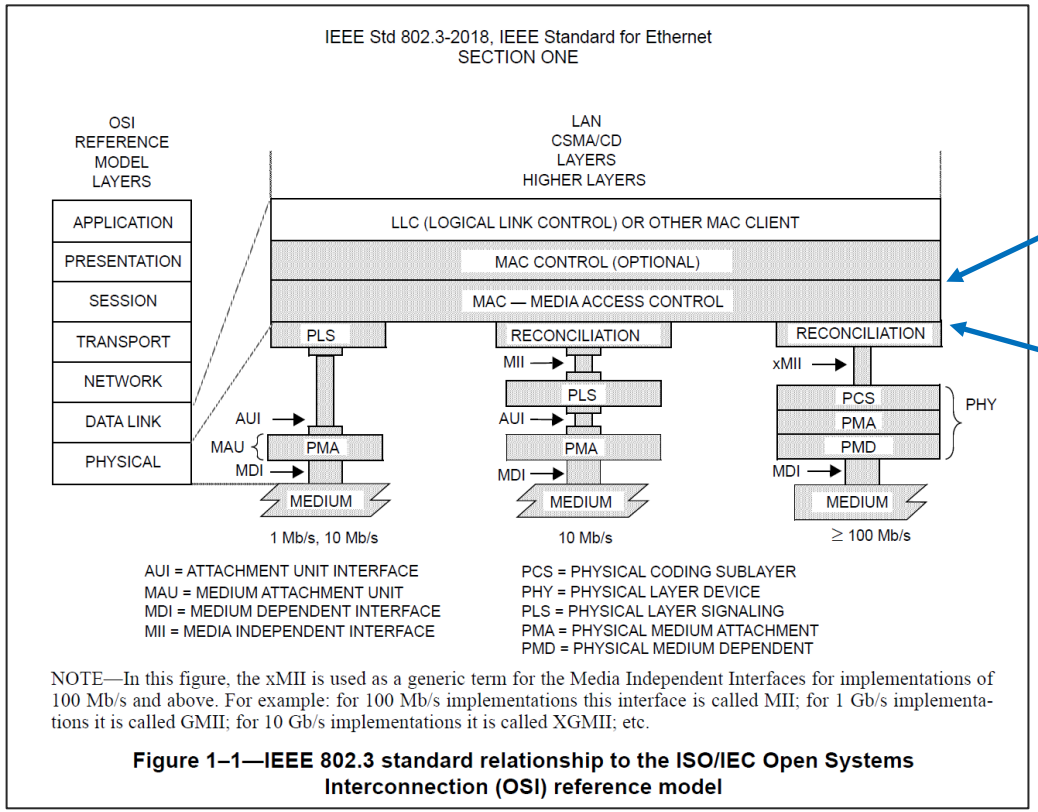
The MA_DATA.request includes the full MSDU and optionally the FCS.

802.1AC: 13.1 Ethernet convergence function (cont.)

- When the convergence function receives an MA_DATA.indication primitive from the underlying MAC Service, it generates a corresponding M_UNITDATA.indication as follows:
  - The destination_address, source_address, *mac_service_data_unit, and frame_check_sequence parameters are passed unaltered.*

The MA_DATA.indication includes the full MSDU and the FCS for a frame received without error.

802.3: 1.1.3 Architectural perspectives



NOTE—In this figure, the xMII is used as a generic term for the Media Independent Interfaces for implementations of 100 Mb/s and above. For example: for 100 Mb/s implementations this interface is called MII; for 1 Gb/s implementations it is called GMII; for 10 Gb/s implementations it is called XGMII; etc.

Figure 1–1—IEEE 802.3 standard relationship to the ISO/IEC Open Systems Interconnection (OSI) reference model

MAC Service Interface (802.3 4.3.2)

Physical Layer Interface (802.3 4.3.3)

802.3: 2.3.1 MA_DATA.request

- 2.3.1.2 Semantics of the service primitive
  - The mac_service_data_unit parameter specifies *the MAC service data unit* to be transmitted by the MAC sublayer entity
  - The receipt of this primitive will cause the MAC entity to *insert all MAC specific fields*, including DA, SA, and any fields that are unique to the particular media access method, and *pass the properly formed frame* to the lower protocol layers for transfer to the peer MAC sublayer entity or entities

The MA_DATA. request provides the full MSDU and optionally the FCS.

802.3: 2.3.2 MA_DATA.indication

- 2.3.2.2 Semantics of the service primitive
  - The mac_service_data_unit parameter specifies the MAC service data unit *as received by the local MAC entity*
  - The frame_check_sequence parameter is the cyclic redundancy check value (see 3.2.9) *as specified by the FCS field of the incoming frame*
- 2.3.2.3 When generated
  - The MA_DATA.indication is passed from the MAC sublayer entity (through the optional MAC Control sublayer, if implemented) to the MAC client entity or entities to indicate the arrival of a frame to the local MAC sublayer entity that is destined for the MAC client. Such frames are *reported only if they are validly formed, received without error*, and their destination address designates the local MAC entity

The MA_DATA.indication includes the full MSDU and the FCS for a frame received without error.

802.3: 4.3.2.1 MAC client transmit interface state diagram

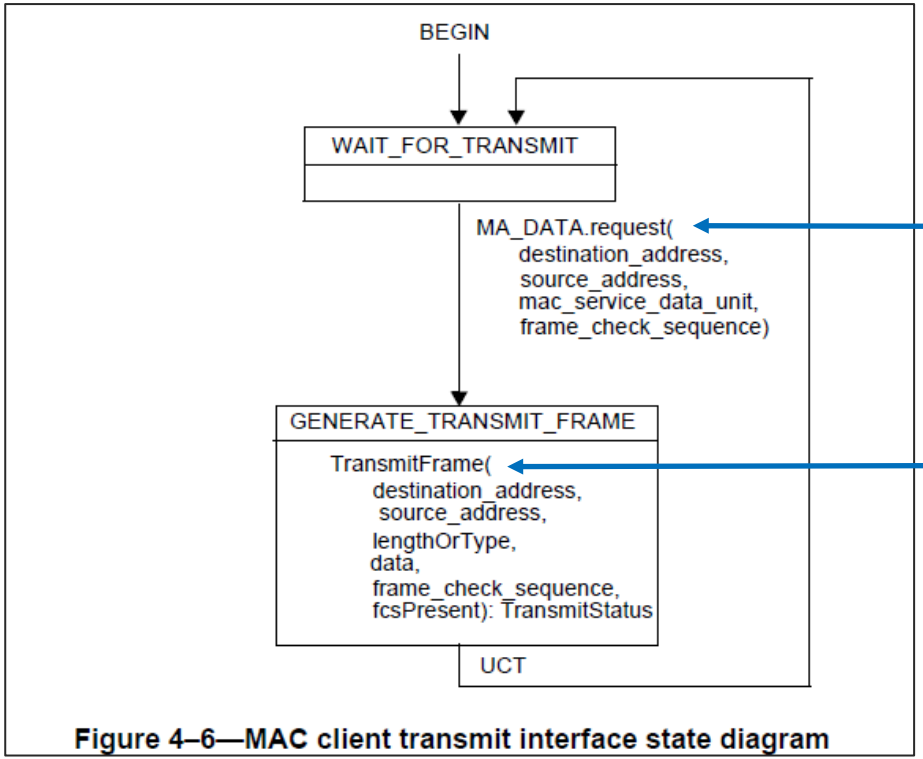


Figure 4–6—MAC client transmit interface state diagram

Atomic transmit request

Atomic transmit function

The formal MAC model defines the TransmitFrame operation as synchronous (4A.2.8 Frame transmission)

802.3: 4.3.2.2.4 MAC client receive interface state diagram

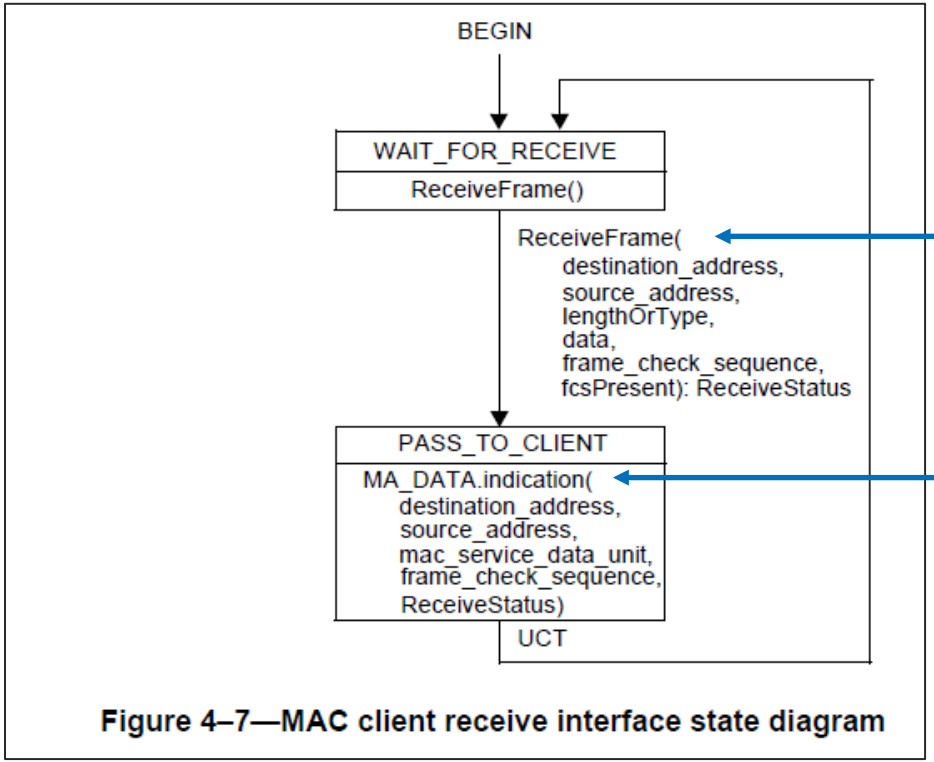


Figure 4–7—MAC client receive interface state diagram

Atomic transmit request

Atomic transmit function

The formal MAC model defines the ReceiveFrame operation as synchronous (4A.2.9 Frame reception)

802.3: 5.2.4.3 Receive variables and procedures

- procedure LayerMgmtReceiveCounters (status: ReceiveStatus);
  - Uses a case statement based on frame receive status
  - Frame and byte counters are ***only incremented in receiveOK case***
  - FCS errors are only incremented in frameCheckError case
  - Other errors (e.g., FrameToLong) are processed in the same way

802.3: 30.3.1 MAC entity managed object class

- 30.3.1.1.5 aFramesReceivedOK
  - A count of frames that are *successfully received* (receiveOK). This *does not include* frames received with frame-too-long, *FCS*, length or alignment errors, or frames lost due to internal MAC sublayer error. This counter is incremented when the *ReceiveStatus is reported as receiveOK*
- 30.3.1.1.6 aFrameCheckSequenceErrors
  - A count of receive frames that are an integral number of octets in length and *do not pass the FCS check*. <snip> This counter is incremented when the *ReceiveStatus is reported as frameCheckError*

CTF & 802.3: Non-compliant externally visible behavior

- Frames start to be transmitted before they are fully received
- FCS errored frames are forwarded
- Counters are incorrect
  - LayerMgmtReceiveCounters (status: ReceiveStatus)
    - Uses case statement based on frame receive status
    - Frame and byte counters are only incremented in receiveOK case.
    - FCS errors are only in only incremented in frameCheckError case

CTF & 802.3: Open questions

- FCS stomp
  - When a CTF bridge receives a frame with an FCS error, it is normal to “stomp” the outbound CRC on transmit.
    - This enables network users to differentiate between a “local link” FCS error, and one from a previous link.
    - There is more than one way to do this, which one or ones should be standardized?
- Counters
  - CTF is clearly inconsistent with the current definition of MAC counters.
  - What should the new behavior be?

Thank You!

Backup

802.3: 4.3.3 Services required from the Physical Layer

- **The MAC Service interface handles MSDUs (really frames), and the Physical Signaling Service handles bits).**
  - During reception, the contents of an incoming frame are retrieved from the Physical Layer by the MAC sublayer via repeated use of the ReceiveBit operation:
    - function ReceiveBit: PhysicalBit;
    - Each invocation of ReceiveBit retrieves one new bit of the incoming frame from the Physical Layer. The ReceiveBit operation is synchronous. Its duration is the entire reception of a single bit. Upon receiving a bit, the MAC sublayer shall immediately request the next bit until all bits of the frame have been received.

802.3: 4.3.3 Services required from the Physical Layer

- **The MAC Service interface handles MSDUs (really frames), and the Physical Signaling Service handles bits).**
  - During transmission, the contents of an outgoing frame are passed from the MAC sublayer to the Physical Layer by way of repeated use of the TransmitBit operation:

    procedure TransmitBit (bitParam: PhysicalBit);

    - Each invocation of TransmitBit passes one new bit of the outgoing frame to the Physical Layer. The TransmitBit operation is synchronous. The duration of the operation is the entire transmission of the bit. The operation completes, when the Physical Layer is ready to accept the next bit and it transfers control to the MAC sublayer.

End