Cut-Through Forwarding (CTF): Towards an IEEE 802.1 Standard

Johannes Specht

(Self; Analog Devices, Inc.; Mitsubishi Electric Corporation; Phoenix Contact GmbH & Co. KG; PROFIBUS Nutzerorganisation e.V.; Siemens AG; Texas Instruments Inc)

Introduction

Proposed Activities

1) IEEE 802 Nendica: Already ongoing

- Discuss administrative and detailed technical aspects of potential lower layer modelling across IEEE 802.3 and IEEE 802.1 with support for CTF.
- Requires sufficient need/interest in IEEE WG 802.1 and IEEE WG 802.3 to work on such a modelling.

2) IEEE WG 802.1: Proposed

- Motion to develop PAR&CSD for an IEEE 802.1 project to standardize CTF as standalone IEEE 802.1 standard (not amendments to 802.1 Standards). Proposed items included in a scope:
  - Support for IEEE Std 802.3-2018 compatible ***real implementations***.
  - Incorporate/standardize IEEE 802.1 aspects of a joint model across IEEE WG 802.1 and 802.3 with support for CTF, if such a ***model*** becomes available during the proposed IEEE 802.1 Stds development project.

3) IEEE 802 Nendica/IEEE WG 802.1: Now … before the proposed motion in item 2)

- Discuss individual contributions that may support the aforementioned motion, and may be considered during development of the PAR&CSD [item 2)] by IEEE WG 802.1.

Note: IEEE 802 Nendica appears to be a good place for item 3), but this may be discussed due ongoing discussion on the processes between IEEE 802 Nendica and IEEE WG 802.1.

A closer Look

Proposal to IEEE WG 802.1

- Motion to develop PAR&CSD for an IEEE 802.1 project to standardize CTF as standalone IEEE 802.1 standard (not amendments to 802.1 Standards).

- Proposed items included in a PAR scope:
    1. Support for IEEE Std 802.3-2018 compatible ***real implementations***.
    2. Incorporate/standardize IEEE 802.1 aspects of a joint model across IEEE WG 802.1 and 802.3 with support for CTF, if such a ***model*** becomes available during the proposed IEEE 802.1 Stds development project.

A clear specification of CTF in the scope of IEEE WG 802.1 appears technically feasible.

- Of course not the entire proposed scope…
- **Options** allowed by scope, although **not** pre-conditions/requirements for the suggested 802.1 Stds development project.

Difference may be small, but worth to talk about (figuratively):

"A MAC …"

v.s.

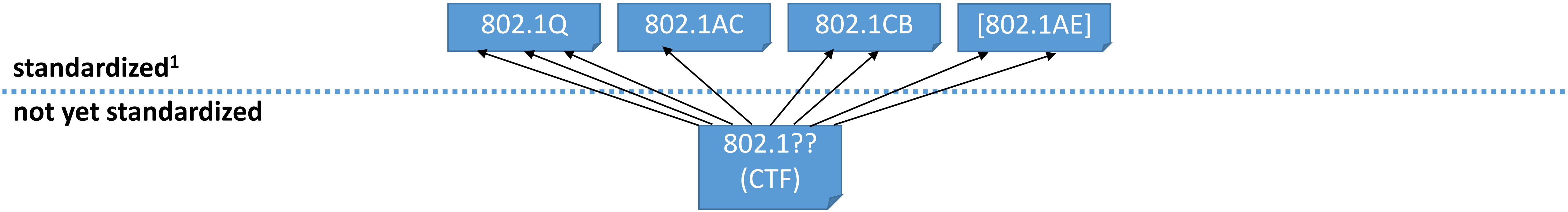
"A MAC with the externally visible behavior specified in IEEE Std 802.3-2018…"

- If both WGs see the technical need to drive this, the model should become part of a running IEEE WG 802.1 project!
- The other way around, decisions documented in associated IEEE 802.1 Stds drafts could be helpful input for development of such a model.
- **IEEE 802 Nendica – contributions are welcome!**

CTF as a standalone 802.1 Standard, not amendments to existing 802.1 standards

See also https://mentor.ieee.org/802.1/dcn/21/1-21-0037-00-ICne-ieee-802-tutorial-cut-through-forwarding-ctf-among-ethernet-networks.pdf

Recap: Proposed Location in IEEE 802.1 Standards

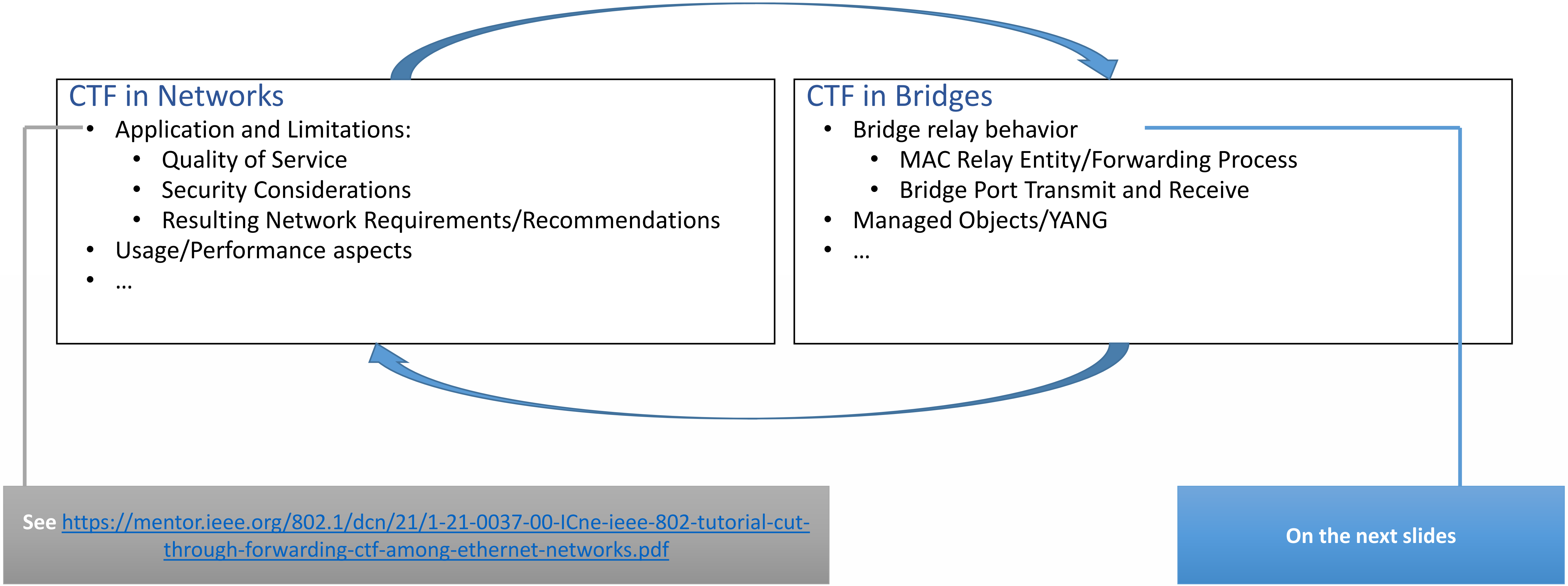


Separate stand alone IEEE 802.1 base standard for CTF

- **Single document**
  Avoids distribution of CTF across existing standards (compared to multiple amendment projects).
- **Exclusion, inclusion/re-use and adjustment of existing protocols**
  - Existing protocols not referred to are basically beyond specification.
  - <u>If no adjustments for CTF are needed:</u>
    Inclusion by reference (e.g., *"as specified in x.y.z of IEEE Std 802.1Xxx-20XX"*) can be sufficient.
  - <u>If adjustments for CTF are needed:</u>
    - Additional description of the differences can be sufficient.
    - Adjustments apply for CTF only; no side effects on existing protocols in absence of CTF support.

[1] The latest published base standards, corrigenda, and a subset of the published amendments.

Recap: Proposed Content Categories and (some) Contents



CTF in Networks

- Application and Limitations:
  - Quality of Service
  - Security Considerations
  - Resulting Network Requirements/Recommendations
- Usage/Performance aspects
- …

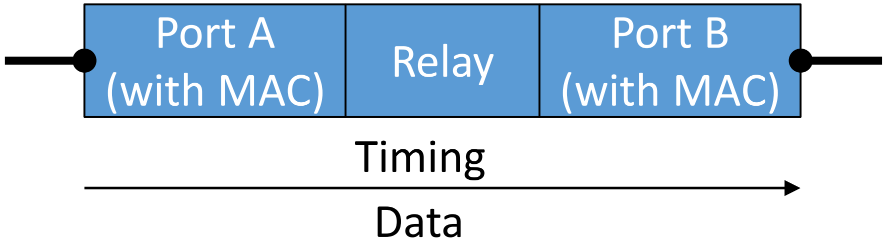
CTF in Bridges

- Bridge relay behavior
  - MAC Relay Entity/Forwarding Process
  - Bridge Port Transmit and Receive
- Managed Objects/YANG
- …

See https://mentor.ieee.org/802.1/dcn/21/1-21-0037-00-ICne-ieee-802-tutorial-cut-through-forwarding-ctf-among-ethernet-networks.pdf

On the next slides

Aspects of externally visible Behavior
(on the relevant path from frame reception to frame transmission)



Two types of aspects define the externally visible behavior of a bridge

1. Data

- What data goes into a bridge, and comes out of the bridge?
  - Ports: What data in frames is transmitted by a Port B, as a result of data in frames received on a Port A?
  - Management: What are the management interfaces, and how can management parameters change/be set?

2. Timing

- When are frames transmitted at a Port B as a result of frame reception at a Port A?

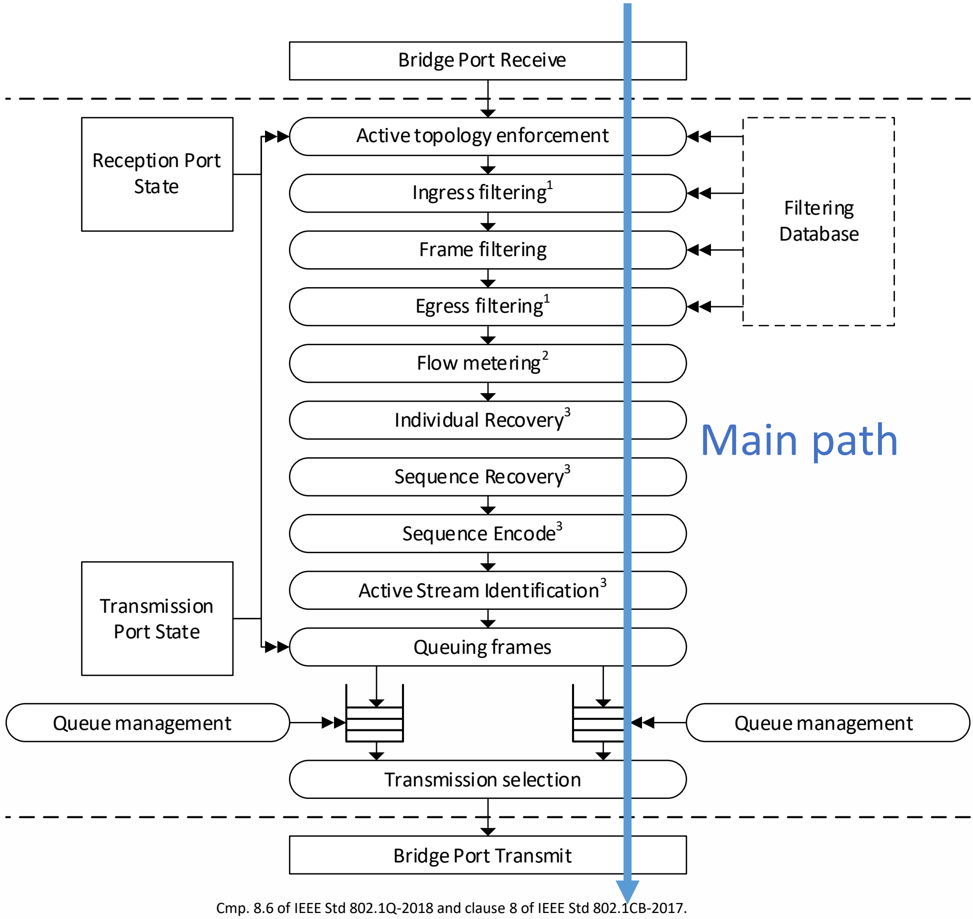
Recap: CTF in Bridges - Relay with CTF Support

Coexistence of CTF & S&F

- **Traffic separation**
  - Decoded priority (VLAN-Tags, IPV assignment)
  - Ports/traffic classes (FDB & decoded priority)
- **Enabling/disabling CTF**
  - Per reception port (the entire port)
  - Per transmission port per traffic class

The standardized model extended

- **Flow**
  (Incomplete) frames pass through processing stages, remain visible to earlier stages.
- **Stalls (incomplete frames)**
  Waiting for more data of an incomplete frame before passing it to the next stage.
- **Stall until completed (incomplete frames)**
  Stalling incomplete frames until completed before passing it to the next stage → Fallback to S&F
- **Late errors (incomplete frames)**
  - Causes (earlier stage)
  - Handling (same or later stage)



Cmp. 8.6 of IEEE Std 802.1Q-2018 and clause 8 of IEEE Std 802.1CB-2017.
1) Not present in MAC Bridges
2) Not present if PSFP and CI are unsupported
3) Not present if FRER is unsupported

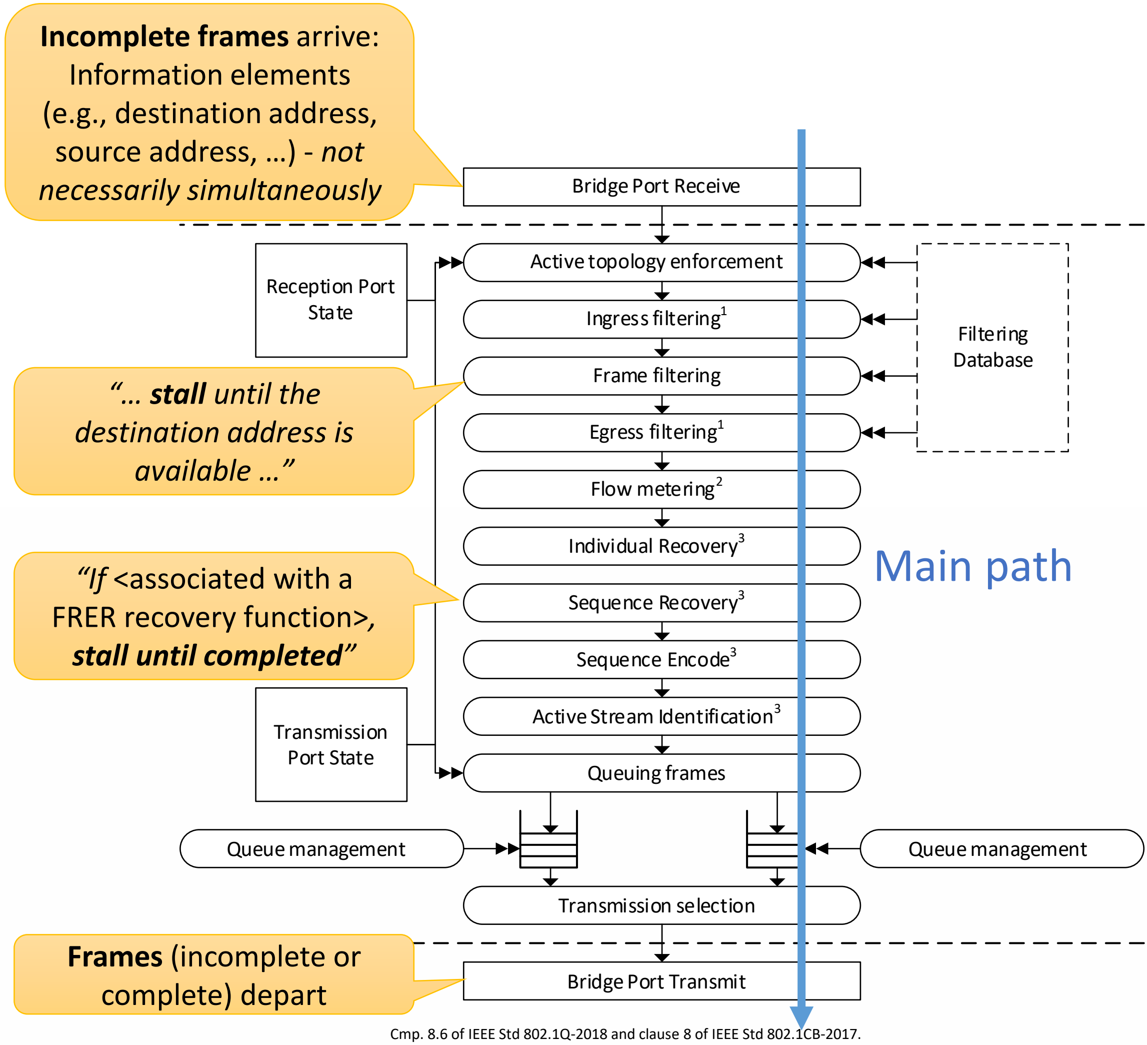
CTF in Bridges: Data and Flow of the Model Illustrated

Upfront: Relevant implications of an IEEE 802.1 standalone Standard for CTF

- **No** (***full***) conformance with IEEE Stds 802.1Q, 802.1AC, etc. required.
- ***Partial*** re-use of elements from existing IEEE 802.1 Stds is reasonable (e.g., individual data transformations), omitting others is possible (e.g., MAC service definition without ISO/IEC 10731 service primitives, handle information elements in frames individually, both)

Resolution of Information Elements

- Information elements of (incomplete) frames at the resolution necessary for the behavior in stages.
  → Keep definitions at a minimum.
- It may be discussed in detail if higher resolution (e.g., octets) are helpful/needed …
- … it likewise appears that such detailed discussions can/should happen during a potential IEEE 802.1 Stds development project according to the defined process.



Incomplete frames arrive: Information elements (e.g., destination address, source address, …) - *not necessarily simultaneously*

*"… **stall** until the destination address is available …"*

"If <associated with a FRER recovery function>, ***stall until completed****"*

Frames (incomplete or complete) depart

Cmp. 8.6 of IEEE Std 802.1Q-2018 and clause 8 of IEEE Std 802.1CB-2017.

1) Not present in MAC Bridges
2) Not present if PSFP and CI are unsupported
3) Not present if FRER is unsupported

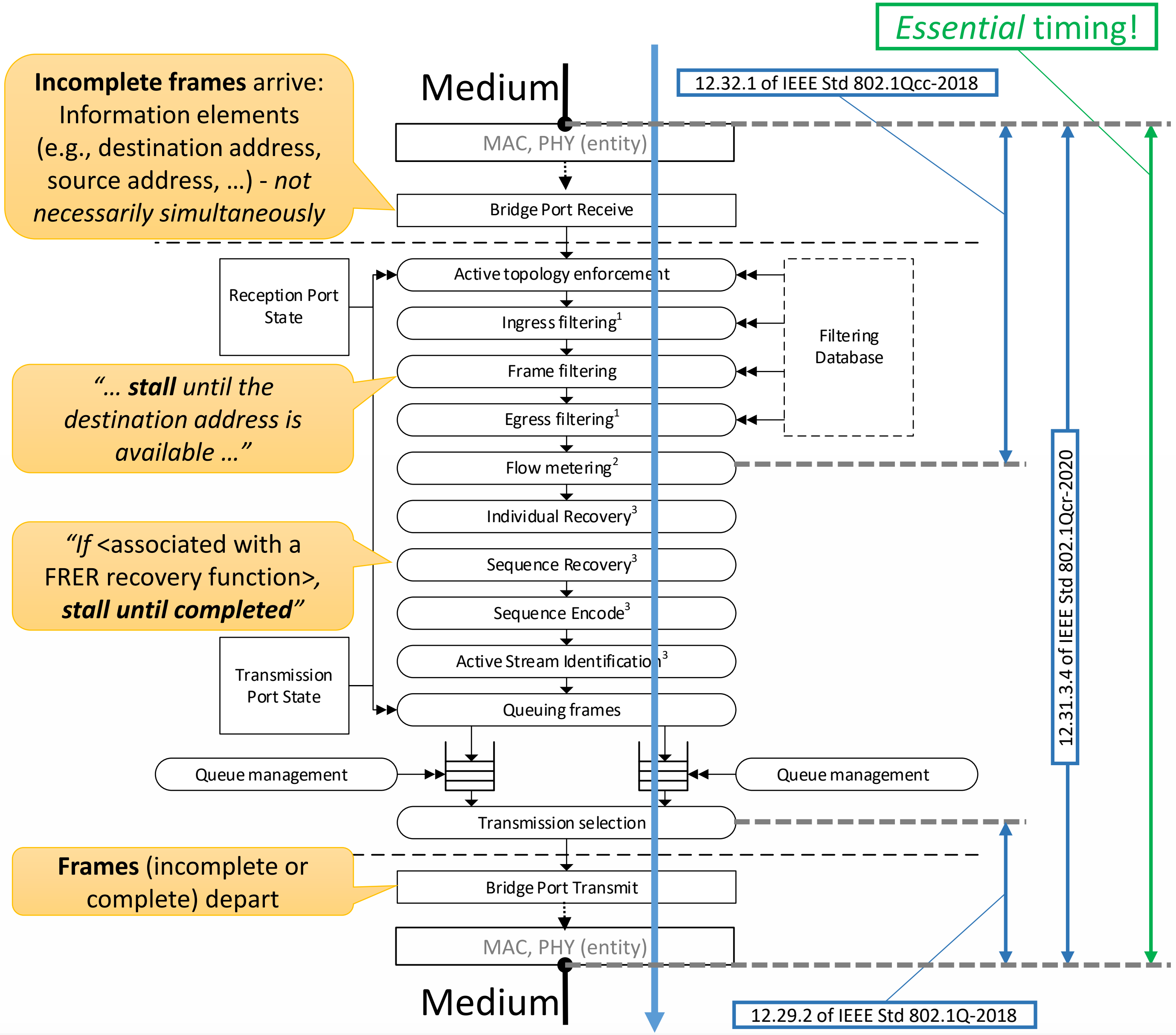
Timing

Essential Timing

- Required for clear specification of the externally visible behavior on the path *medium→ bridge→ medium* with CTF.

Can it be specified in an IEEE 802.1 Std?

- An IEEE 802.1 Std is the right place to do this:
  - A perception of this timing only exists if there is “*something”* (i.e., bridge) between the network media.
  - ➢ For S&F Bridges, the essential timing (and more) is found in existing 802.1 standards.



Incomplete frames arrive: Information elements (e.g., destination address, source address, …) - *not necessarily simultaneously*

*“… **stall** until the destination address is available …”*

“If <associated with a FRER recovery function>, ***stall until completed”***

Frames (incomplete or complete) depart

Essential timing!

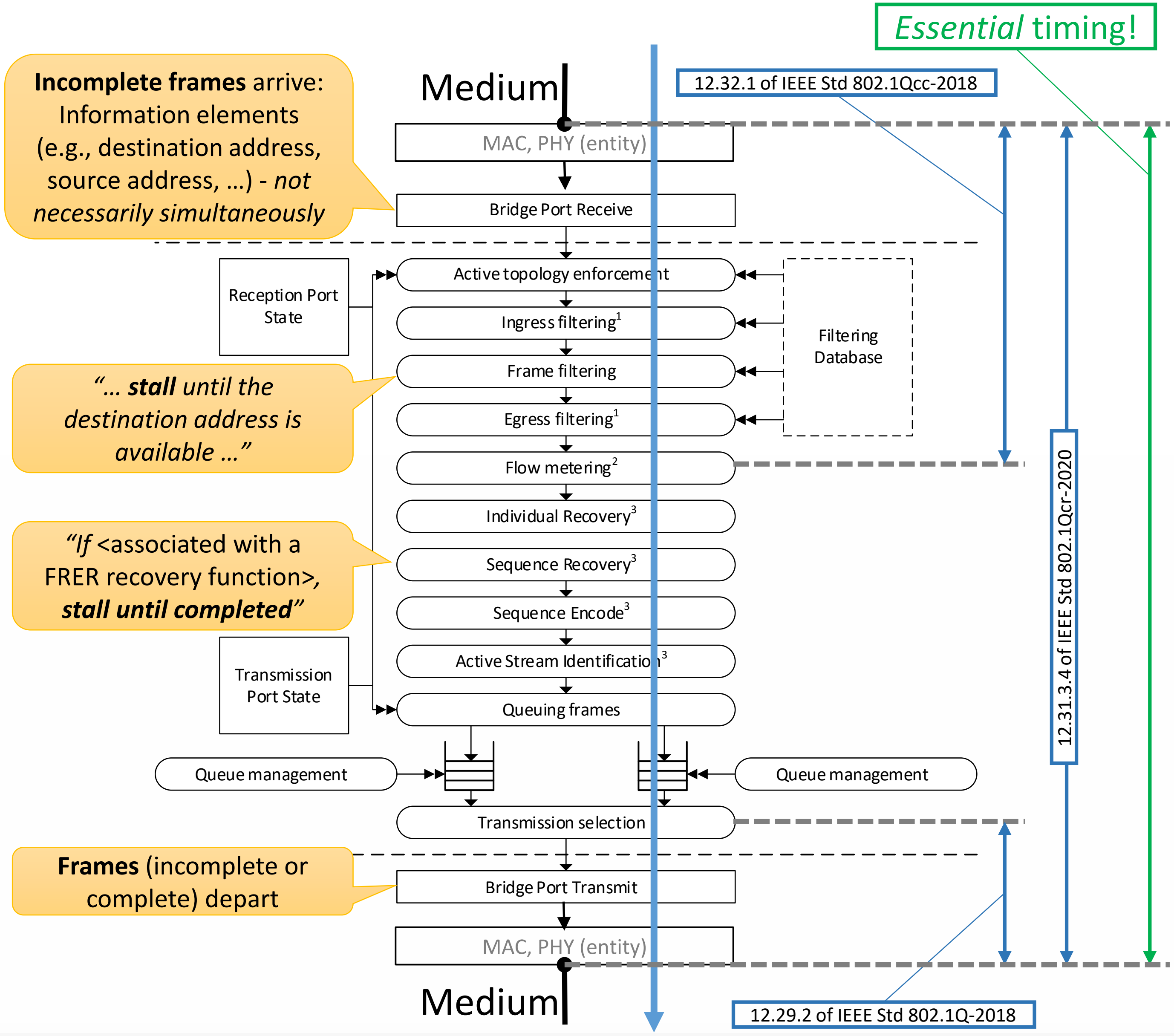
Intermediate Discussion

Technical questions

- Fundamentally missing elements that prevent clear specification of the behavior of a bridge with support for CTF?
- Items beyond the scope of IEEE WG 802.1?

Non-technical questions

- Comments for getting a better impression about the “for and against” of standardizing CTF?
- Other non-technical aspects?



Incomplete frames arrive: Information elements (e.g., destination address, source address, …) - *not necessarily simultaneously*

*“… **stall** until the destination address is available …”*

“If <associated with a FRER recovery function>, ***stall until completed****”*

Frames (incomplete or complete) depart

Essential timing!

Support for real implementations of IEEE Std 802.3-2018 possible?

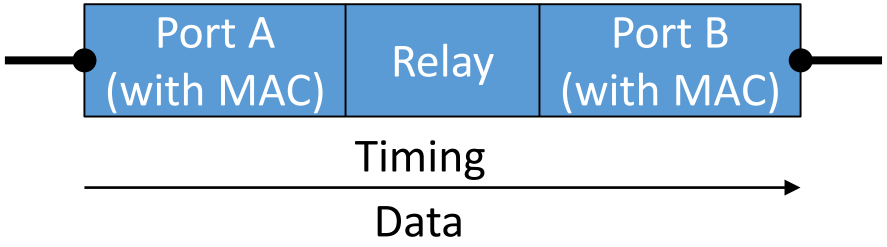
Considerations with the focus on externally visible behavior

Model v.s. Implementations (1)

> *It is important to distinguish, however, between the model and a real implementation. The **model** is **optimized for simplicity and clarity** of presentation, while any realistic **implementation** shall place heavier **emphasis on** such constraints as **efficiency and suitability** to a particular implementation technology or computer architecture.* [4A.2.2 of IEEE Std 802.3-2018, “Overview of the Procedural Model”]

> … ***it is the behavior of any MAC sublayer implementations that shall match the standard**, not their internal structure. The internal details of the procedural model are useful only to the extent that they help specify that behavior clearly and precisely.* [item b) in 4A.2.2.1 of IEEE Std 802.3-2018, “Ground rules for the procedural model”]

Model v.s. Implementations (2)



Model

The handling of incoming and outgoing frames is rather stylized in the procedural model, in the sense that ***frames are handled as single entities*** *by most of the MAC sublayer and are only serialized for presentation to the Physical Layer.* [item c) in 4A.2.2.1 of IEEE Std 802.3-2018, "Ground rules for the procedural model"]

Implementations

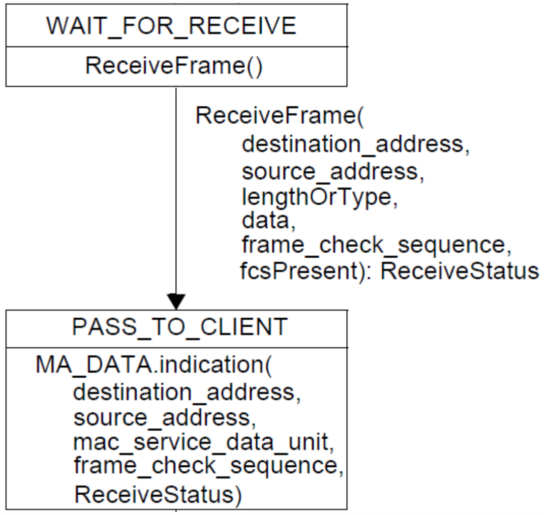
In reality, ***many implementations will instead handle frames serially on a bit, octet or word basis****. This approach has not been reflected in the procedural model, since this only complicates the description of the functions without changing them in any way.* [item c) in 4A.2.2.1 of IEEE Std 802.3-2018, "Ground rules for the procedural model"]

Observations & Considerations

- MAC *implementations* that handle frames serially appear conformant to IEEE Std 802.3-2018.
- Serial handling fits well to the concept around "incomplete frames" proposed on earlier slides.
- Aspects of the data behavioral that appear important to look at (only "fast path" in a bridge):
  1. Port A: Is it possible to access frame contents in the relay in presence FCS errors?
  2. Port B: Is it possible to send FCS errors forwarded by the relay (i.e., no unintended "repair")?

Note: Less than minFrameSize octets appears likewise important, but is not in focus of this slide set (see also https://mentor.ieee.org/802.1/dcn/21/1-21-0058-01-ICne-ctf-forwarding-timing-in-industrial-automation.pdf).

Data: Receive Path



WAIT_FOR_RECEIVE

ReceiveFrame()

ReceiveFrame(
destination_address,
source_address,
lengthOrType,
data,
frame_check_sequence,
fcsPresent): ReceiveStatus

PASS_TO_CLIENT

MA_DATA.indication(
destination_address,
source_address,
mac_service_data_unit,
frame_check_sequence,
ReceiveStatus)

Source: Figure 4A-4 of IEEE Std 802.1-2018

From 4A.2.9 of IEEE Std 802.3-2018

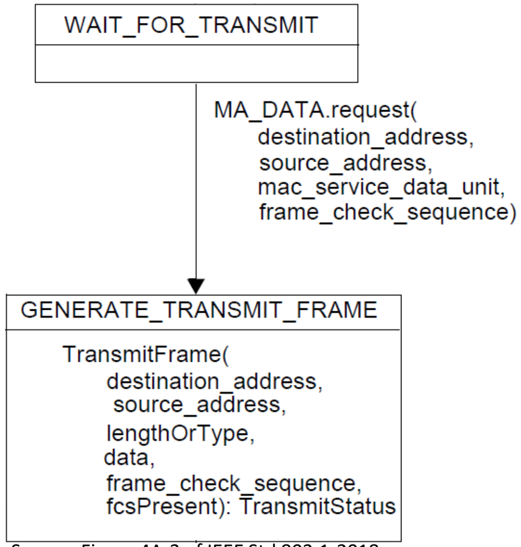
```
procedure ReceiveLinkMgmt;
begin
  repeat
    StartReceive;
    while receiving do nothing; {Wait for frame to finish arriving}
    excessBits := frameSize mod 8;
    frameSize := frameSize – excessBits; {Truncate to octet boundary}
    receiveSucceeding := (frameSize ≥ minFrameSize) {Reject frames too small}
  until receiveSucceeding
end; {ReceiveLinkMgmt}
```

From 4A.2.9 of IEEE Std 802.3-2018

```
function ReceiveFrame (
 …
  function ReceiveDataDecap: ReceiveStatus; {Nested function; see body below}
…
function ReceiveDataDecap: ReceiveStatus;
…
  receiveSucceeding := LayerMgmtRecognizeAddress(destinationField);
  if receiveSucceeding then
  begin {Disassemble MAC frame}
    destinationParam := destinationField;
    sourceParam := sourceField;
    lengthOrTypeParam := lengthOrTypeField;
    dataParam := RemovePad(lengthOrTypeField, dataField);
    fcsParamValue := fcsField;
    fcsParamPresent := passReceiveFCSMode;
    …
    if exceedsMaxLength then status := frameTooLong
    else if fcsField = CRC32(incomingFrame) then
    if validLength then status := receiveOK else status := lengthError
    else if excessBits = 0 then status := frameCheckError
    else status := alignmentError;
  …
  ReceiveDataDecap := status
end; {ReceiveDataDecap}
```

→ It seems that contents of valid and **several types of invalid frames including frames with invalid FCS,** are transferred to MAC clients (e.g., Relay) other than MAC control and LLC (see also 3.4 of IEEE Std 802.3-2018).

Data: Transmit Path



Source: Figure 4A-3 of IEEE Std 802.1-2018

From 4A.2.8 of IEEE Std 802.3-2018

```
process BitTransmitter;
begin {Inner loop}
…
 while transmitting do
 begin
  TransmitBit(outgoingFrame[currentTransmitBit]);
  currentTransmitBit := currentTransmitBit + 1;
  transmitting := (currentTransmitBit ≤ lastTransmitBit)
 end
…
```

From 4A.2.8 of IEEE Std 802.3-2018

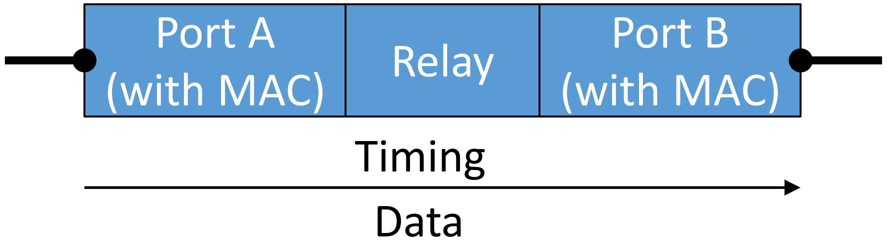
```
function TransmitFrame (
 destinationParam: AddressValue;
 sourceParam: AddressValue;
 lengthOrTypeParam: LengthOrTypeValue;
 dataParam: DataValue;
 fcsParamValue: CRCValue;
 fcsParamPresent: Bit): TransmitStatus;
 procedure TransmitDataEncap; {Nested procedure; see body below}
 begin
  if transmitEnabled then
  begin
   TransmitDataEncap;
   TransmitFrame := TransmitLinkMgmt
…
```

From 4A.2.8 of IEEE Std 802.3-2018

```
procedure TransmitDataEncap;
begin
 with outgoingFrame do
 …
 destinationField := destinationParam;
 sourceField := sourceParam;
 lengthOrTypeField := lengthOrTypeParam;
 if fcsParamPresent then
 begin
  dataField := dataParam; {No need to generate pad if the FCS is passed from MAC client}
  fcsField := fcsParamValue {Use the FCS passed from MAC client}
…
```

→ It seems that contents of **frames from a MAC client, including frames with invalid FCS,** are transmitted.

Intermediate Discussion



Discussion

- It seems both, the transmit and the receive path of IEEE 802.3 Std conformant MAC implementations allow exchange of frames with invalid FCS between a MAC client (relay) and a PHY entity.
- The simplified MAC (clause 4A of IEEE Std 802.3-2018) has been considered, but the same observation appears valid for the CSMA/CD MAC (clause 4 of IEEE Std 802.3-2018).
- Not mentioned in the slides before were error counters on the receive path (this is nothing new) …
- … but there may be other implications, for example, during transmission.
- Should IEEE WG 802.3 be asked for having a look at invalid FCS’?

Summary & Conclusions

Summary & Conclusion

Proposed to IEEE WG 802.1

- Motion to develop PAR&CSD for an IEEE 802.1 project to standardize CTF as standalone IEEE 802.1 standard (not amendments to existing 802.1 Standards).

Technical Contents in this Slide Set

- An approach for clear specification of bridge behavior with support for CTF appears feasible and in scope of IEEE WG 802.1.
- Considerations on the option to refer to implementations of IEEE Std 802.3-2018 in a potential IEEE 802.1 standard for CTF.

Contributions related to PAR&CSD

- If appropriate, the author is happy to prepare and present individual contributions, independent of whether IEEE 802.1 would consider such contributions during a potential future PAR&CSD development or not.

Thank you for your Attention!

Questions, Opinions, Ideas?

Johannes Specht

Dipl.-Inform. (FH)

GERMANY
johannes.specht.standards@gmail.com