CTF Forwarding Timing in Industrial Automation

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

Introduction

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 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.

Focus of this slide set!

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

"A MAC for IEEE Std 802.3 physical media"

v.s.

"An IEEE Std 802.3 compatible MAC implementation …"

Support for real implementations of IEEE Std 802.3-2018

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"]

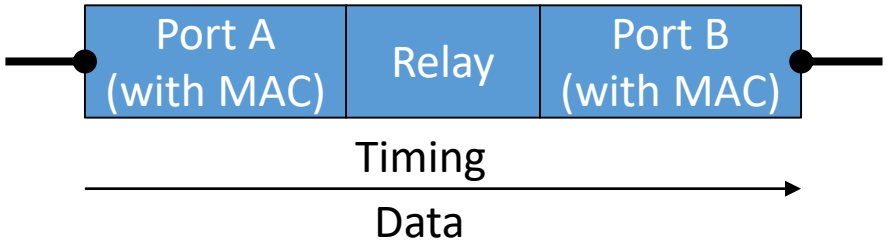
Implementations

… 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 during the IEEE 802 Plenary Tutorial on CTF
  (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 , section "IEEE 802.1 Considerations")
- It is the behavior that matters. The behavior of a Bridge is visible via (a) management variables and **(b) by frame transmission, which can be a result of frame reception**.

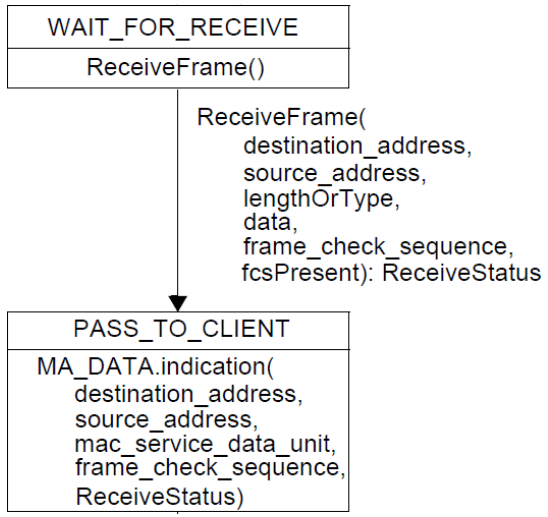
Aspects of externally visible Behavior
(on the relevant path from frame reception to frame transmission)



There are two aspects

- **Data**
  - Reception
    - Which data is passed, potentially serially, from a Port A to the relay during reception?
    - Are contents of frames with invalid FCS available to the relay?
  - Transmission
    - Which data is to be passed, potentially serially, from the Relay to a Port B during transmission?
    - Can frames with invalid FCS be transmitted (e.g., avoid unintended “correction”)?
- **Timing**
  - When are frames transmitted at a Port B as a result of frame reception at a Port A?

The Issue



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

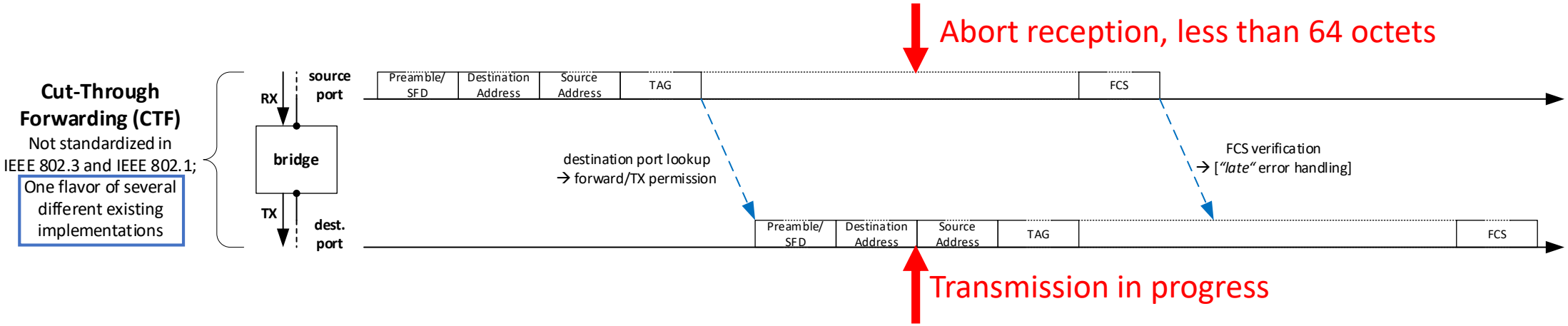
```
function ReceiveFrame (
 ...
  function ReceiveDataDecap: ReceiveStatus; {Nested function; see body below}
...
function ReceiveDataDecap: ReceiveStatus;
…
  receiveSucceeding := LayerMgmtRecognizeAddress(destinationField);
  if receiveSucceeding then
  begin {Disassemble MAC frame}
    destinationParam := 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}
```

→ ***Regardless of timing, serialization, etc. of real implementations – undersized frames are not passed to the MAC client (relay)!***

→ ***If IEEE 802.1 decides to standardize CTF, and the option to claim compatibility with real MAC implementations of IEEE Std 802.3 is desired, it appears worth to talk about minimum frame sizes.***

Where does it matter?

From the Tutorial



Affected flavors of CTF: Forwarding prior to 64 octets (measured from the DA)

- Example:
  - Forwarding after the Tag (e.g., 18 octets)
  - Reception on RX aborted after 40 octets
  - Some octets already transmitted on TX

  → Irrespectively of the RX→TX timing in a Bridge, this should have never happened!
- What should a CTF Bridge to?
  - → Abort transmission: Potentially new issues, now on the TX path!
  - → Pad to 64 octets: Obviously not!

→Avoiding the case entirely: Wait for 64 octets prior to forwarding.

Where does it matter?

My Understanding

- At least to my knowledge, there are implementations that go below 64 octets.
- On the other hand, the relevance of forwarding prior to 64 octets highly depends on:
  1. remaining device timing properties
  2. network scheduling aspects

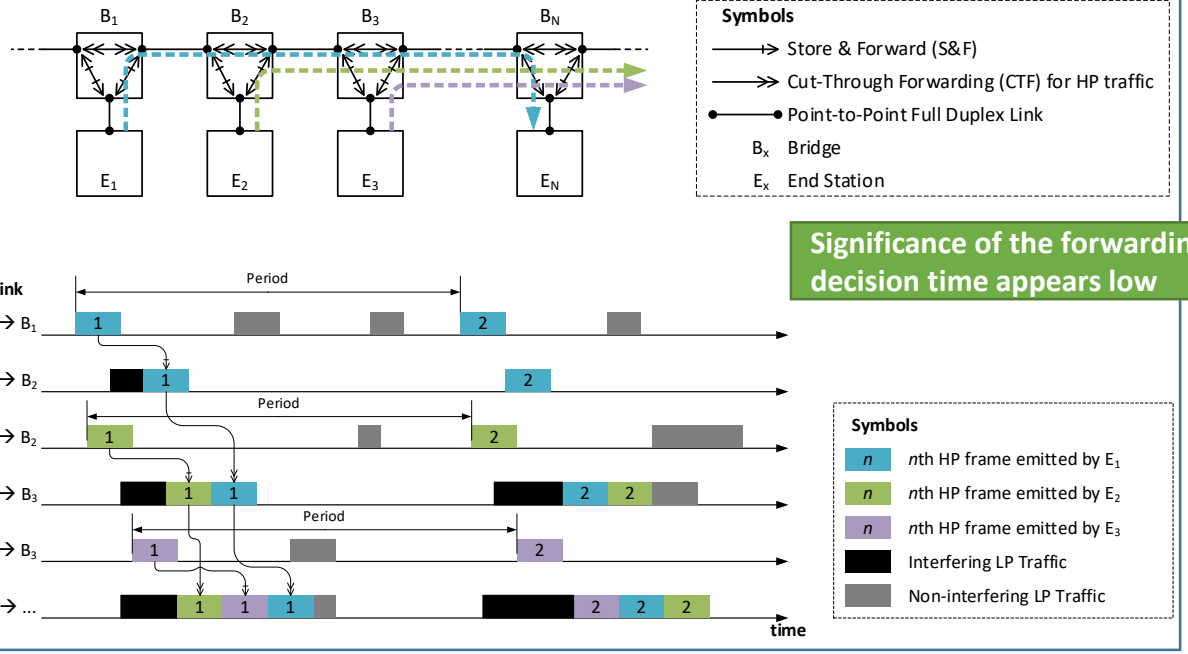
My Question

- Could *industrial automation* parties live with forwarding after 64 octets, in the case that this property decides on whether conformance with real implementations of IEEE Std 802.3 would be enabled by this?

→ Keep this question in mind, let's look at two more slides from the Tutorial before discussing!

CTF Speed-up in Max. End-to-End Delay: With Interference

Scenario: Uncoordinated transmission times



fwd. after VLAN-Tag

Significance of the forwarding decision time appears low

fwd. after 64 octets

		CTF-to-S&F max. delay ratio (end-to-end)									
		Preemption unsupported					Preemption supported				
H	Link \ l_{HP}	128	256	512	1024	1542	128	256	512	1024	1542
2	100 Mbit/s	96%	93%	87%	81%	78%	84%	77%	73%	70%	69%
4	100 Mbit/s	95%	91%	85%	78%	74%	80%	73%	67%	64%	63%
16	100 Mbit/s	95%	89%	82%	74%	69%	77%	68%	61%	57%	56%
64	100 Mbit/s	94%	89%	81%	73%	68%	76%	66%	60%	55%	54%
2	1 Gbit/s	97%	93%	88%	82%	79%	88%	80%	75%	71%	70%
4	1 Gbit/s	96%	92%	86%	78%	74%	85%	76%	69%	65%	64%
16	1 Gbit/s	96%	90%	83%	75%	70%	83%	72%	64%	59%	57%
64	1 Gbit/s	96%	90%	82%	73%	68%	82%	71%	62%	57%	55%
2	2,5 Gbit/s	98%	94%	89%	83%	79%	93%	85%	78%	73%	71%
4	2,5 Gbit/s	98%	93%	87%	80%	75%	92%	81%	73%	67%	65%
16	2,5 Gbit/s	97%	92%	85%	76%	71%	90%	78%	68%	61%	59%
64	2,5 Gbit/s	97%	92%	84%	75%	69%	90%	77%	66%	59%	57%

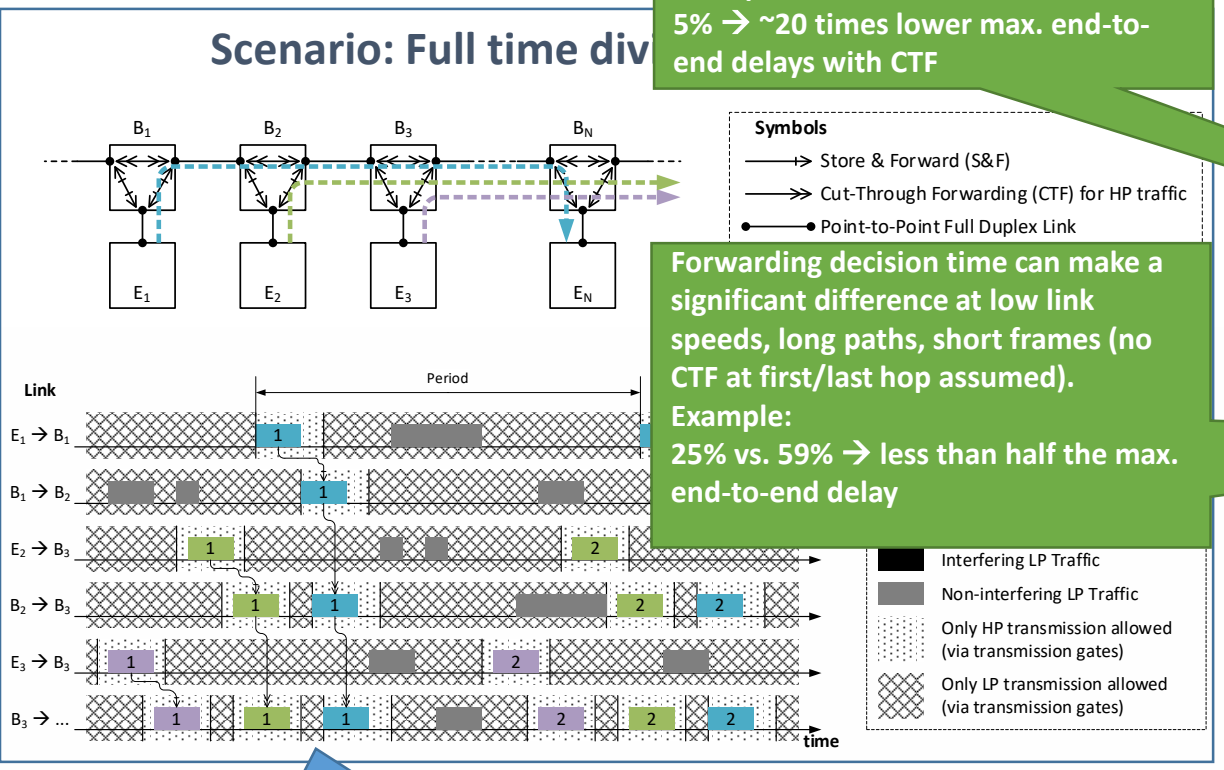
		CTF-to-S&F max. delay ratio (end-to-end)									
		Preemption unsupported					Preemption supported				
H	Link \ l_{HP}	128	256	512	1024	1542	128	256	512	1024	1542
2	100 Mbit/s	98%	94%	89%	82%	79%	91%	82%	75%	71%	70%
4	100 Mbit/s	98%	93%	86%	79%	75%	89%	78%	70%	66%	64%
16	100 Mbit/s	97%	92%	84%	75%	70%	87%	74%	65%	59%	57%
64	100 Mbit/s	97%	91%	83%	74%	69%	87%	73%	63%	58%	55%
2	1 Gbit/s	98%	94%	89%	83%	79%	92%	83%	76%	72%	70%
4	1 Gbit/s	98%	93%	87%	79%	75%	90%	79%	71%	66%	64%
16	1 Gbit/s	97%	92%	84%	75%	70%	88%	76%	66%	60%	58%
64	1 Gbit/s	97%	91%	83%	74%	69%	88%	74%	64%	58%	56%
2	2,5 Gbit/s	98%	94%	89%	83%	79%	93%	85%	78%	73%	71%
4	2,5 Gbit/s	98%	93%	87%	80%	75%	92%	81%	73%	67%	65%
16	2,5 Gbit/s	97%	92%	85%	76%	71%	90%	78%	68%	61%	59%
64	2,5 Gbit/s	97%	92%	84%	75%	69%	90%	77%	66%	59%	57%

See the annex of the tutorial slides for more information (device, network, and traffic assumptions, math, more results, etc.)

CTF Speed-up in Max. End-to-End Delay: Without Interference

In general, CTF becomes quite significant if interferences are avoided.
Example:
5% → ~20 times lower max. end-to-end delays with CTF

Forwarding decision time can make a significant difference at low link speeds, long paths, short frames (no CTF at first/last hop assumed).
Example:
25% vs. 59% → less than half the max. end-to-end delay



All interferences suppressed (via 802.1Qbv in this scenario)

fwd. after VLAN-

fwd. after 64 octets

		CTF-to-S&F max. delay ratio (end-to-end) Preemption supported or not				
H	Link \ l_{HP}	128	256	512	1024	1542
2	100 Mbit/s	61%	56%	53%	51%	51%
4	100 Mbit/s	48%	41%	37%	35%	35%
16	100 Mbit/s	31%	21%	16%	14%	13%
64	100 Mbit/s	25%	14%	9%	6%	5%
2	1 Gbit/s	75%	64%	58%	54%	53%
4	1 Gbit/s	67%	52%	43%	39%	37%
16	1 Gbit/s	56%	36%	25%	18%	16%
64	1 Gbit/s	52%	31%	18%	11%	8%
2	[illegible]bit/s	88%	74%	64%	58%	55%
4	2,5 Gbit/s	84%	66%	52%	44%	40%
16	2,5 Gbit/s	79%	55%	36%	25%	21%
64	2,5 Gbit/s	77%	50%	31%	18%	13%

		CTF-to-S&F max. delay ratio (end-to-end) Preemption supported or not				
H	Link \ l_{HP}	128	256	512	1024	1542
2	100 Mbit/s	79%	65%	57%	54%	52%
4	100 Mbit/s	72%	53%	43%	38%	37%
16	100 Mbit/s	62%	37%	24%	18%	15%
64	100 Mbit/s	59%	31%	17%	10%	8%
2	1 Gbit/s	83%	69%	60%	55%	54%
4	1 Gbit/s	78%	59%	47%	40%	38%
16	1 Gbit/s	70%	45%	29%	20%	17%
64	1 Gbit/s	68%	40%	23%	13%	10%
2	2,5 Gbit/s	88%	74%	64%	58%	55%
4	2,5 Gbit/s	84%	66%	52%	44%	40%
16	2,5 Gbit/s	79%	55%	36%	25%	21%
64	2,5 Gbit/s	77%	50%	31%	18%	13%

See the annex of the tutorial slides for more information (device, network, and traffic assumptions, math, more results, etc.)

Thank you for your Attention!

Time for Discussion
- now, and subsequently -

Difference may be small and **no** pre-condition/requirement for the suggested 802.1 Stds development project, but worth to talk about (figuratively):

"A MAC for IEEE Std 802.3 physical media"

v.s.

"An IEEE Std 802.3 compatible MAC implementation …"

Johannes Specht
Dipl.-Inform. (FH)
GERMANY
johannes.specht.standards@gmail.com