synchronizationUncertain feature in IEEE P1588-rev (for IEEE P802.1AS-rev comment i-368)

Rodney Cummings

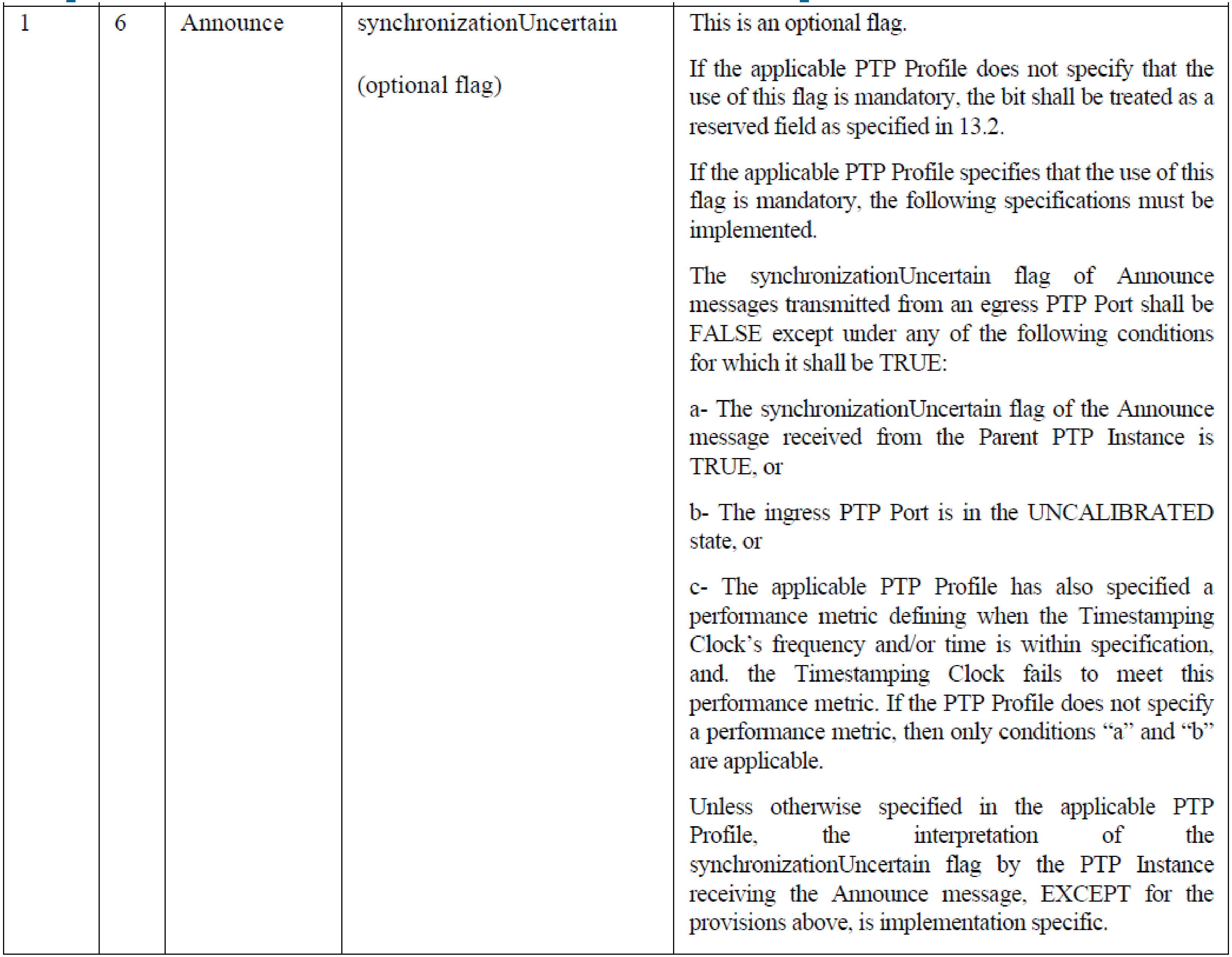
National Instruments

1588-rev syncUncertain History

Application Profile Requirements

- Applications have timing requirements
  - e.g. telecom, power grid, industrial, automotive
- Applications specify profiles of 1588
  - Transitive for app profile of 802.1AS (e.g. IEEE/IEC 60802)
- How does a 1588 device know that app profile requirements are met along the entire path from the Grandmaster?
  - Solution brought into 1588-rev from ITU-T: **synchronizationUncertain**

Spec in 1588-rev Initial Sponsor



1	6	Announce	synchronizationUncertain (optional flag)	This is an optional flag. If the applicable PTP Profile does not specify that the use of this flag is mandatory, the bit shall be treated as a reserved field as specified in 13.2. If the applicable PTP Profile specifies that the use of this flag is mandatory, the following specifications must be implemented. The synchronizationUncertain flag of Announce messages transmitted from an egress PTP Port shall be FALSE except under any of the following conditions for which it shall be TRUE: a- The synchronizationUncertain flag of the Announce message received from the Parent PTP Instance is TRUE, or b- The ingress PTP Port is in the UNCALIBRATED state, or c- The applicable PTP Profile has also specified a performance metric defining when the Timestamping Clock's frequency and/or time is within specification, and. the Timestamping Clock fails to meet this performance metric. If the PTP Profile does not specify a performance metric, then only conditions "a" and "b" are applicable. Unless otherwise specified in the applicable PTP Profile, the interpretation of the synchronizationUncertain flag by the PTP Instance receiving the Announce message, EXCEPT for the provisions above, is implementation specific.

Benefit of Flag in Announce

- Application in 1588 device knows when it has stable synchronization along the complete path
  - Stable according to the application's requirements

- It is impossible to instantaneously know
  if "all" devices in the network have stable sync
- For a given device, this tells me what matters:
  - Is my sync stable?

Remote Management

- What if we have a remote management client?
  - Client might not be running 1588
  - Client wants to read syncUncertain in node(s)
- This is not related to operation of the application
  - Non-essential, but still important
  - Presenter submitted comment to 1588 initial sponsor ballot
    - Must-be-satisfied=No
    - So far, comment's response is Accept In Principle

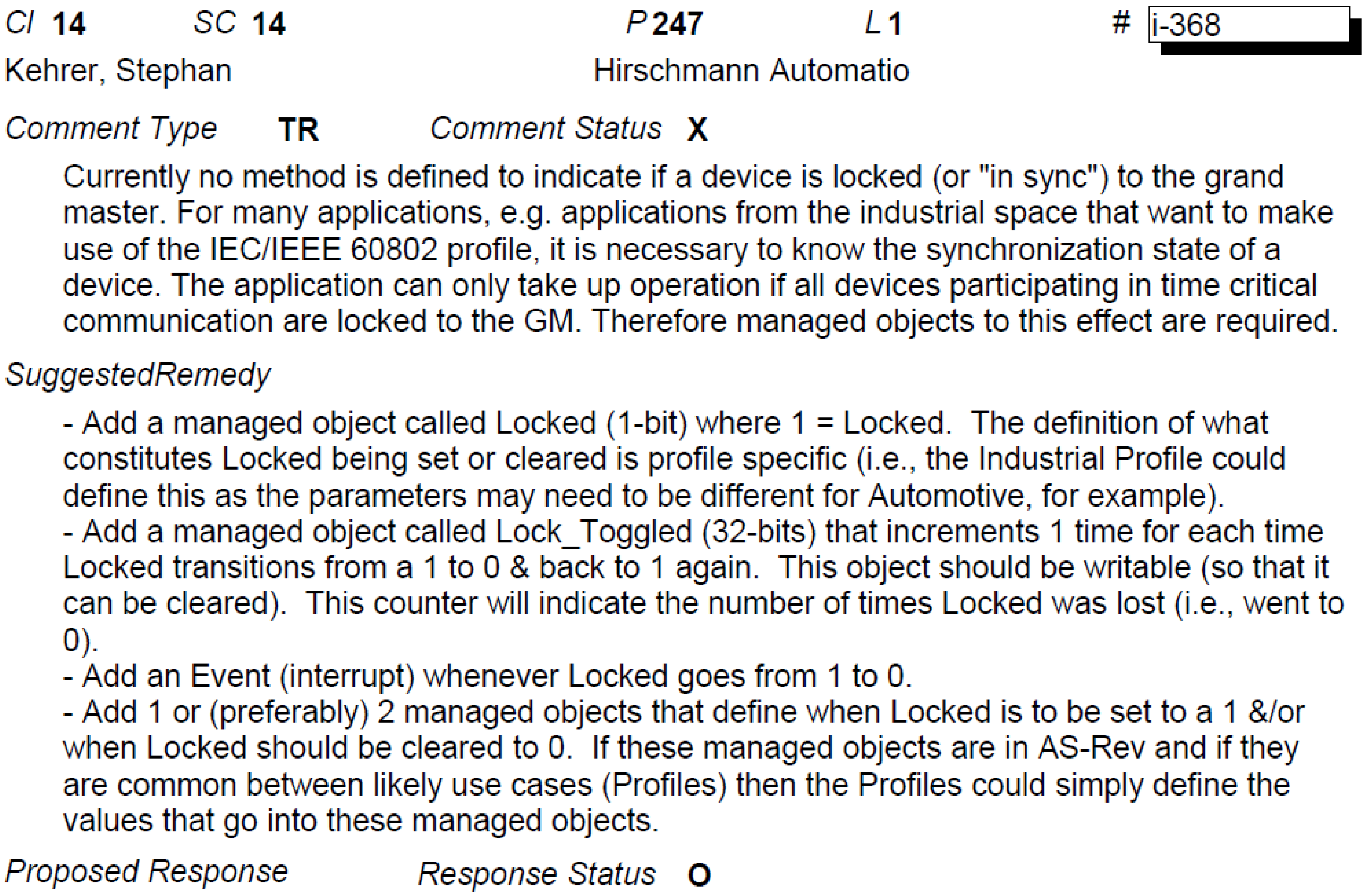
Coming in 1588-rev Sponsor Recirc 1

- currentDS.synchronizationUncertain
  - Read-only value of this node's flag (boolean)
  - Transmitted in Announce (when applicable)
- parentDS.synchronizationUncertain
  - Read-only value received from parent
  - Received in Announce (when applicable)

- Recirc 1 has not started as of this date

Comment i-368 of 802.1AS-rev Initial Sponsor

Comment i-368



Cl **14** *SC* **14** *P* **247** *L* **1** # i-368

Kehrer, Stephan Hirschmann Automatio

Comment Type **TR** *Comment Status* **X**

Currently no method is defined to indicate if a device is locked (or "in sync") to the grand master. For many applications, e.g. applications from the industrial space that want to make use of the IEC/IEEE 60802 profile, it is necessary to know the synchronization state of a device. The application can only take up operation if all devices participating in time critical communication are locked to the GM. Therefore managed objects to this effect are required.

SuggestedRemedy

- Add a managed object called Locked (1-bit) where 1 = Locked. The definition of what constitutes Locked being set or cleared is profile specific (i.e., the Industrial Profile could define this as the parameters may need to be different for Automotive, for example).
- Add a managed object called Lock_Toggled (32-bits) that increments 1 time for each time Locked transitions from a 1 to 0 & back to 1 again. This object should be writable (so that it can be cleared). This counter will indicate the number of times Locked was lost (i.e., went to 0).
- Add an Event (interrupt) whenever Locked goes from 1 to 0.
- Add 1 or (preferably) 2 managed objects that define when Locked is to be set to a 1 &/or when Locked should be cleared to 0. If these managed objects are in AS-Rev and if they are common between likely use cases (Profiles) then the Profiles could simply define the values that go into these managed objects.

Proposed Response *Response Status* **O**

Proposal

- 802.1AS-rev Response for i-368
  - REJECT: This comment is addressed by the synchronizationUncertain feature of 1588-rev, and therefore it is not needed in 802.1AS-rev
- IEC/IEEE 60802
  - "shall support the synchronizationUncertain feature of 1588-2019"
  - Application-specific performance metrics for syncUncertain
    - As expected by the 1588-rev specs

Benefits of Proposal

- Short and simple
  - Avoids delay of 802.1AS-rev
- Avoids major new feature, coordinating with app profiles
  - Should be done in task or working group, not sponsor ballot
- Meets essential need for 60802
  - Operation of the application
  - Industrial controllers/sensors/actuators already have mechanisms to detect and react to faults in a device
    - This is a fault like any other

Remote Management

- In presenter's opinion, this is non-essential for 60802
- YANG: syncUncertain will be in 1588's YANG
  - If 802.1AS YANG is an augment, inherited
- If 60802 requires for MIB, changes needed in 802.1AS
  - Two data set members: Managed objects and MIB
  - Extra features
    - Statistics counter: Managed object and MIB
    - Notification: MIB only
    - Application-specific managed objects in 802.1AS

Duplication of 1588 Text

Presenter's Opinion on Duplication

- If 802.1AS integrates syncUncertain, do so by reference
  - Do not duplicate text from 1588 in 802.1AS document
- In past this duplication was targeted at recommendations to read 802.1AS and ignore 1588
  - In other words, read a profile and ignore its base standard
  - Thankfully, most implementers ignore this advice
  - Following this advice can lead to bugs
- Duplication is harmful to standards development
  - Conformance contradictions, re-invention, more editor work
- Leave existing duplication, but avoid in future

Thank You