

```

+-----+
| IEEE 802.1 REVISION REQUEST 0140 |
+-----+

```

DATE: July 15, 2014
NAME: Bob Noseworthy
COMPANY/AFFILIATION: University of New Hampshire's InterOperability Lab
E-MAIL: ren@iol.unh.edu

REQUESTED REVISION:

STANDARD: 802.1AS-2011
CLAUSE NUMBER: 11
CLAUSE TITLE: MDPdelayReq State Machine (Figure 11-8)

RATIONALE FOR REVISION:

There exists numerous scenarios wherein asCapable, once set TRUE, can immediately be set FALSE with little or no delay.

A time sensitive network utilizing 802.1AS's gPTP is likely to be extremely sensitive to any spurious indication of asCapable being FALSE in an otherwise operational environment. Consider an AVB network utilizing MSRP, per IEEE 802.1BA, the existing stream reservations would be torn down if a system allowed asCapable to be set FALSE momentarily (as port would be an SRPdomainBoundary).

Three prime causes of this "hair trigger":

Consider the WAITING_FOR_PDELAY_INTERVAL_TIMER of the MDPdelayReq State Machine (Figure 11-8). Specifically the IF-ELSE statement quoted below:

```

if ((neighborPropDelay <= neighborPropDelayThresh) &&
    (rcvdPdelayRespPtr->sourcePortIdentity.clockIdentity != thisClock) &&
    neighborRateRatioValid)
    asCapable = TRUE;
else
    asCapable = FALSE;

```

Thus, if at any time during normal path delay packet exchange, an implementation ever computes neighborPropDelay to exceed neighborPropDelayThresh, then asCapable will immediately be set false. The computation of neighborPropDelay is not set by the standard (10.2.4.7, end of 11.1.2 "any scheme...is acceptable"), and thus may (or may not) have suitable averaging and error handling to limit spurious values.

Similarly, the determination of neighborRateRatioValid is not set by the standard (11.2.15.1.11, 11.2.15.2.3 - "Any scheme ... is acceptable"), and thus may result in an implementation momentarily setting this value to FALSE.

Finally, a single received PdelayResp with an improper clockIdentity would cause asCapable to be FALSE.

Beyond these three immediate "hair triggers", there may be value in greater hysteresis to prevent asCapable from being "quickly" reset to FALSE once it has been set to TRUE.

Currently, when a Pdelay_Req is sent, but a response is not received (lost) or received but is invalid (still considered "lost"), then asCapable will eventually be set to FALSE.

Specifically in the RESET state of Figure 11-8, asCapable will be set to FALSE as soon as lostResponses > allowedLostResponses. allowedLostResponses is defined to be 3 (11.5.3). lostResponses is initialized to zero by the state machine, and incremented only after the value is checked, hence on the fourth lost/invalid response to a Pdelay_Req, asCapable will be set to FALSE.

PROPOSED REVISION TEXT:

These possible remedies are only this commenter's observations and require further study, ongoing within the IEEE 802.1ASbt working group.

No strong recommendation is given; however, two complementary possibilities are explored below.

Specific to the allowedLostResponses hysteresis issue:

The behaviour could be altered to increase the hysteresis by changing the allowedLostResponses value to a value substantially greater than 3, but only if asCapable has been set for "a time" - but this must be balanced with the need to rapidly detect a link-local issue and fail over to an alternate network path (if available).

A trivial approach may address this by defining a boolean "asCapableIsStable" that is set to TRUE in the WAITING_FOR_PDELAY_INTERVAL_TIMER if asCapable=TRUE for 60 consecutive times. Once asCapableIsStable is TRUE, then the value of allowedLostResponses could be increased from 3 to 10 (and restored to 3 when asCapableIsStable is FALSE. This approach has the problem of greatly increasing the delay before asCapable is FALSE in the face of a true problem with Pdelay exchanges.

=====

Specific to the hair-trigger behavior behaviour in WAITING_FOR_PDELAY_INTERVAL_TIMER. This state could be amended to preclude the possibility of asCapable being set FALSE immediately if one of the three "immediate" hair-triggers is encountered. This could be as simple as adding a counter "detectedFaults" similar to "allowedLostResponses"

For example, asCapable could be set to FALSE only if 3 faults are detected in a row, by changing the else statement in this state:

From just "asCapable=FALSE;" to an if/else statement as follows:

```

if ((neighborPropDelay <= neighborPropDelayThresh) &&
    (rcvdPdelayRespPtr->sourcePortIdentity.clockIdentity != thisClock) &&
    neighborRateRatioValid)
{
    asCapable = TRUE;
    detectedFaults = 0;
}
else
{
    if (detectedFaults <= allowedFaults)
        detectedFaults += 1;
    else
    {
        asCapable = FALSE;
        detectedFaults = 0;
    }
}

```

where detectedFaults is properly initialized to zero in INITIAL_SEND_PDELAY_REQ and allowedFaults is defined, perhaps to 3 to be consistent with the current allowedLostResponses. This solution could also be adjusted to not require the faults to be "in a row", however if this adjustment were pursued, a window would have to be defined (eg: "3 faults in 60" would cause asCapable to be FALSE)

=====

A better approach may be to still allow for a rapid fault detection mechanism to allow asCapable to be set FALSE when an issue is encountered, but provide stronger guidance on the neighborPropDelay and neighborRateRatioValid. Perhaps testable minimum requirements such as requiring neighborPropDelay outlier rejection and/or averaging mechanisms and requirements over certain windows.

IMPACT ON EXISTING NETWORKS:

Any changes that address the raised hair-trigger issues should only improve stability of the overall time sensitive network, balanced with application needs for rapid start up and rapid fault detection and fail-over.

```

+-----+
| Please attach supporting material, if any |
| Submit to:- Glenn Parsons, Chair IEEE 802.1 |
| and copy:- John Messenger, Vice-Chair IEEE 802.1 |
| E-Mail: stds-802-1-maint-req@ieee.org |
|
| +----- For official 802.1 use -----+ |
| | REV REQ NUMBER: 0140 | |
| | DATE RECEIVED: 15 July 2014 | |
| | EDITORIAL/TECHNICAL | |
| | ACCEPTED/DENIED | |
| | BALLOT REQ'D YES/NO | |
| | Status: R | |
| +-----+ |
+-----+

```