```
+------------------------------------------------------------------+
| IEEE 802.1 REVISION REQUEST 0138                                 |
+----------------==================================================+
```

DATE: May 7, 2014
NAME: Bob Noseworthy, Jeff Laird
COMPANY/AFFILIATION: University of New Hamsphire's InterOperability Lab
                     with support from the AVnu Alliance
E-MAIL: ren@iol.unh.edu, laird@iol.unh.edu

REQUESTED REVISION:
     STANDARD: 802.1AS-2011
     CLAUSE NUMBER: 10.2.11, Figure 10-8
     CLAUSE TITLE: PortSyncSyncSend state machine

RATIONALE FOR REVISION:

The behaviour specified by "Figure 10-8 - PortSyncSyncSend state
machine" allows a Sync message to be sent at 1/2 the specified Sync
interval. This is unintended behaviour and may result in the more
frequent transmission of Sync messages than desired (or supported) by
the link partner.

This issue arises as the PortSyncSyncSend state machine will issue
its own Sync message if no Sync has been received (rcvdPSSync is false)
by the time the system's local syncinterval has expired.

The condition in question is the left-hand exit from the
SET_SYNC_RECEIPT_TIMEOUT_TIME state, specifically:

(((rcvdPSSync &&
(currentTime – lastSyncSentTime >= 0.5*syncInterval) &&
rcvdPSSyncPtr->localPortNumber != thisPort))
|| ((currentTime – lastSyncSentTime >= syncInterval) &&
(lastRcvdPortNum != thisPort)))
&& portEnabled && pttPortEnabled && asCapable &&
selectedRole[thisPort] == MasterPort

Specifically the first significant expression (below):
rcvdPSSync &&
(currentTime – lastSyncSentTime >= 0.5*syncInterval)

Thus,
if the system's syncinterval (denoted here as 'syncinterval(slave)')
is slightly smaller than that of the upstream (toward the grandmaster)
system's syncinterval (denoted here as 'syncinterval(master)'), then
consider the case where the system's syncinterval(slave) has just
expired, resulting in the system transmitting a Sync message, and
quickly returning to the SET_SYNC_RECEIPT_TIMEOUT_TIME state. Moments
later, the upstream system's syncinterval(master) expires, and sends a
Sync message to the slave port of the system. This then causes another
rcvdPSSync to be true, and thus when 0.5*syncinterval(slave) expires,
the system will issue another Sync message. This is just one example
of how this shortened syncinterval may occur.

PROPOSED REVISION TEXT:

As one (of two) possible solutions, consider to:

Remove the 0.5* from the exit condition expression. This may result in
a slight delay in a system adjusting to send Sync messages faster when
the local syncInterval has been decreased via management (or Signalling
message) action, however this change should prevent the spurious
transmission of Sync messages at 0.5*syncTnterval.

Such a change would also allow for the condition to be logically
reduced to:

(currentTime – lastSyncSentTime >= syncInterval) &&
((rcvdPSSync &&
rcvdPSSyncPtr->localPortNumber != thisPort)
||
((lastRcvdPortNum != thisPort)))
&& portEnabled && pttPortEnabled && asCapable &&
selectedRole[thisPort] == MasterPort

If this change were adopted, the loop out and in to the
SET_SYNC_RECEIPT_TIMEOUT_TIME state should also remove the text "0.5*"
from the statement.

Further consideration may be warranted for the possibilities of dynamic
changes to syncInterval (via management or Signalling), or any
allowance for links to operate with differing syncIntervals on the same
network; however no immediate significant impact is seen with the
proposed change.

ALTERNATIVE REMEDY:
 A change where the state machine would favour waiting for incoming
Sync messages (rcvdPSSync) before generating its own Sync message
could be favoured. This approach necessitates a bounding of the
allowable range for the syncInterval (if nominally 1second, it will
vary by +/- ??)
Currently AS does not define an allowed range. IEEE 1588v2 specifies
a +/-30% range on such an interval. This may be considered too broad,
or perhaps allow messages to be sent 'too fast'.
If +/-30% were used, then the expression could be modified to:

(((rcvdPSSync &&
(currentTime – lastSyncSentTime >= 0.7*syncInterval) &&
rcvdPSSyncPtr->localPortNumber != thisPort))
|| ((currentTime – lastSyncSentTime >= 1.3*syncInterval) &&
(lastRcvdPortNum != thisPort)))
&& portEnabled && pttPortEnabled && asCapable &&
selectedRole[thisPort] == MasterPort

with a similar change on the looping transition into the
SET_SYNC_RECEIPT_TIMEOUT_TIME state (change the 0.5* there to
0.7*)

IMPACT ON EXISTING NETWORKS:

No known negative impact.
The proposed maintenance item will reduce the chance of erroneous or
spurious transmissions of Sync messages at 1/2 the desired interval,
thus increasing the likelihood that any conformant device will accept
and process the sync under the expected/requested load.
Existing devices that follow the current standard will interoperate
seamlessly with the proposed modification, but such legacy devices may
still generate the shortened Sync interval from Master ports.

```
+----------------------------------------------------------------------+
| Please attach supporting material, if any                            |
| Submit to:- Tony Jeffree, Chair IEEE 802.1                           |
|  and copy:- Glenn Parsons, Vice-Chair IEEE 802.1                     |
|     E-Mail: stds-802-1-maint-req@ieee.org                            |
|                                                                      |
|             +------- For official 802.1 use -----------+            |
|             | REV REQ NUMBER: 138                       |            |
|             | DATE RECEIVED: 13 May 2014                |            |
|             | EDITORIAL/TECHNICAL                       |            |
|             | ACCEPTED/DENIED                           |            |
|             | BALLOT REQ'D YES/NO                       |            |
|             | Status: R                                 |            |
|             +-------------------------------------------+            |
+----------------------------------------------------------------------+
```