Rune Haugom <rune@ontimenet.com>
To: Geoffrey M Garner <gmgarner@alum.mit.edu>
Reply-To: rune@ontimenet.com
Re: 802.1as - syncReceiptTimeout after losing one followup

Hi Goeff,

It is always messy when there is too much text. Let's see if the attached image will make it clearer.

Best regards,
Rune

Den 05.06.2013 04:06, skrev Geoffrey Garner:
Hi Rune,

Sorry for the slight delay in responding. I want to clarify one point. It appears that the upstream node (D2) sends the next Sync (with sequenceId 101) before it sends the Follow_Up for this Sync (with sequenceId 100). That is because the Sync
with Id 101 arrives before the Follow_Up. Did you intend this?

Thanks.

Best regards,

Geoff

On May 31, 2013, at 8:51 AM, Rune Haugom <rune@ontimenet.com> wrote:

Hi Goeff,

We are wondering if you could clearify a question we have related to Figure 11-6 and syncReceiptTimeout in the 802.1as standard.
To us it seems that it is possible to get a syncReceiptTimeout after losing one followup message only.

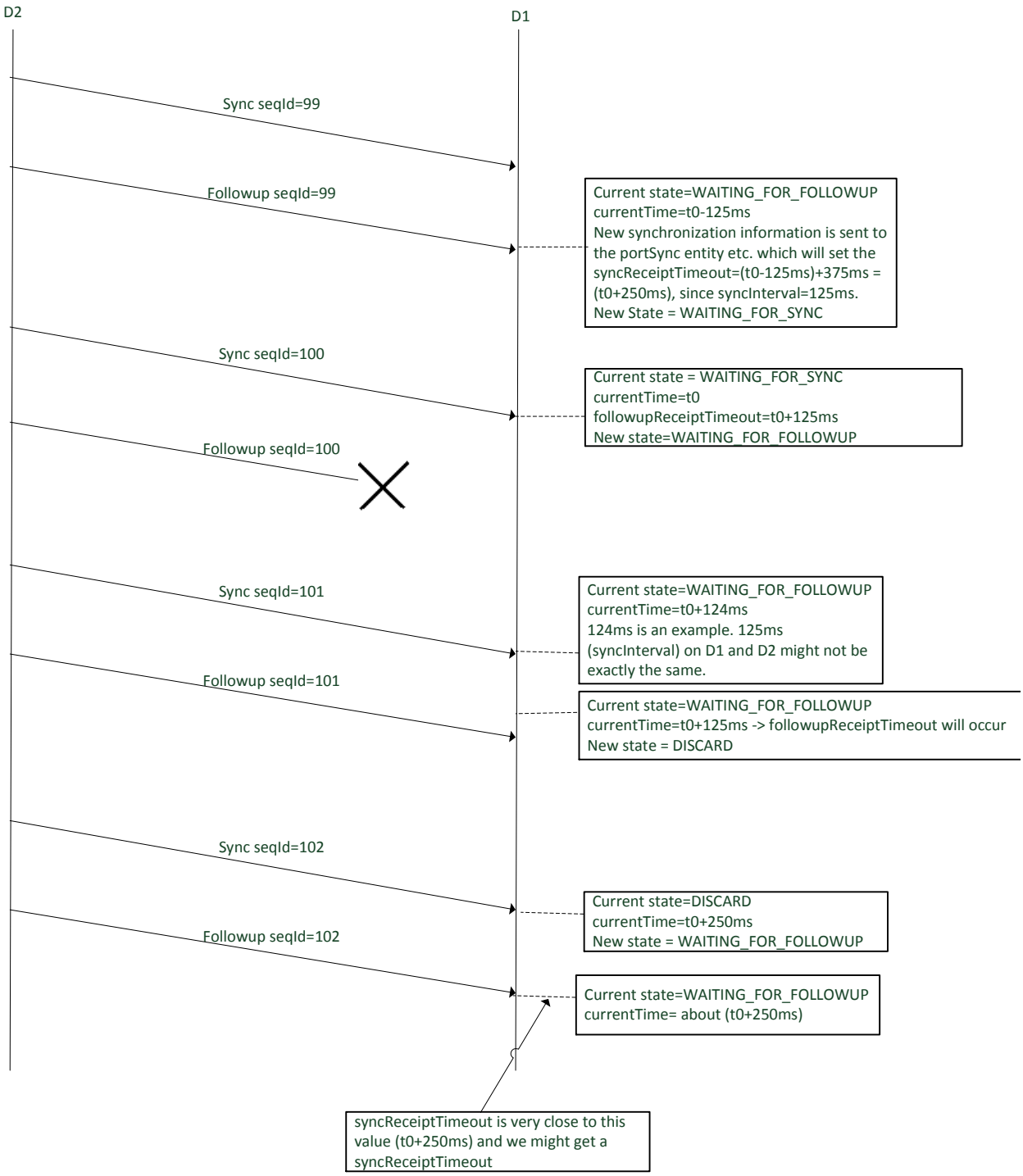
Assume syncInterval=125ms and that device 1 (D1) is a slave to device 2 (D2).
The below statements are referring to Figure 11-6 in the 802.1as standard.
- Assume state = WAITING_FOR_SYNC when D1 gets an incoming sync with sequence id=100.
- D1 will move to state = WAITING_FOR_FOLLOWUP and calculate the followUp

timeout to be 125ms from now (t0). (Below we will assume this followup is lost.)
- Due to different timings on the neighbor device (D2) we might get another incoming sync (sequence id=101) at t0+124ms. This Sync message will not be handled at this time.
- D1 will get the followUp timeout at (t0+125)ms and move to state=DISCARD. The already received sync with sequence id = 101 will be discarded in this state.
- At time (t0+250)ms another sync message (sequence id=102) will enter the state machine.
- It is now 250ms since D1 received the sync message with sequence id = 100. And it is 375ms since the last synchronization information that was sent to the portSync entity from the MD entity.
- Since the timers can vary a couple of milliseconds, we might actually get a syncReceiptTimeout since that timer was schedules to happen 375ms into the future at time (t0-125)ms. (t0-125ms is the time the last synchronization information was received by the PortSync entity.)

In this example one can get a syncReceiptTimeout after losing only one followup message. Do you agree/disagree? If you agree, is this the intention?

Best regards,

Rune
OnTime Networks AS


D2
D1
Sync seqId=99
Followup seqId=99
Current state=WAITING_FOR_FOLLOWUP
currentTime=t0-125ms
New synchronization information is sent to the portSync entity etc. which will set the syncReceiptTimeout=(t0-125ms)+375ms = (t0+250ms), since syncInterval=125ms.
New State = WAITING_FOR_SYNC
Sync seqId=100
Current state = WAITING_FOR_SYNC
currentTime=t0
followupReceiptTimeout=t0+125ms
New state=WAITING_FOR_FOLLOWUP
Followup seqId=100
Sync seqId=101
Current state=WAITING_FOR_FOLLOWUP
currentTime=t0+124ms
124ms is an example. 125ms (syncInterval) on D1 and D2 might not be exactly the same.
Followup seqId=101
Current state=WAITING_FOR_FOLLOWUP
currentTime=t0+125ms -> followupReceiptTimeout will occur
New state = DISCARD
Sync seqId=102
Current state=DISCARD
currentTime=t0+250ms
New state = WAITING_FOR_FOLLOWUP
Followup seqId=102
Current state=WAITING_FOR_FOLLOWUP
currentTime= about (t0+250ms)
syncReceiptTimeout is very close to this value (t0+250ms) and we might get a syncReceiptTimeout