IEEE 1588
Default PTP Profiles
Conformance Test Plan

for IEEE 1588-2008: IEEE Standard Profile for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems

Version 0.0.2
Technical Document

NOTICE: This is a living document. All contents are subject to change.
Individual tests and/or test groups may be added/deleted/renumbered in forthcoming revisions.
General feedback and comments are welcome through the IEEE 1588 Consortium at UNH-IOL.

Last Updated January 10, 2014

UNH InterOperability Laboratory
121 Technology Drive, Suite 2
Durham NH 03824

Tel: +1 603-862-0090
Fax: +1 603-862-8141
Email: ptplab@iol.unh.edu

TABLE OF CONTENTS

MODIFICATION RECORD

Version	Date	Editor(s)	Comments
0.0.1	2013-06-12	Carol Perkins	Initial Pre-release Draft Test Plan

ACKNOWLEDGMENTS

We would like to acknowledge the efforts of the following individuals in the development of this test plan:

Carol Perkins	UNH InterOperability Laboratory
Jeff Laird	UNH InterOperability Laboratory
Ryan McEachern	UNH InterOperability Laboratory
Bob Noseworthy	UNH InterOperability Laboratory

INTRODUCTION

The University of New Hampshire InterOperability Laboratory (UNH-IOL) is a non-profit institution designed to promote the industry adoption of standards through conformance and interoperability testing. This particular test plan has been developed to help implementers evaluate the **1588 Default PTP Profile** functionality of their products.

These tests are designed to determine if a product conforms to specifications defined in *IEEE Standard Profile for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems.* Successful completion of all tests contained in this suite does not guarantee that the tested device will successfully operate with other 1588 Default PTP Profile products. However, when combined with a satisfactory level of interoperability testing, these tests provide a reasonable level of confidence that the Device Under Test (DUT) will function properly in many 1588 Default PTP Profile environments.

The tests contained in this document are organized in order to simplify the identification of information related to a test, and to facilitate the actual testing process. Tests are separated into groups, primarily in order to reduce setup time in the lab environment, however the different groups typically also tend to focus on specific aspects of device functionality.

This test plan format is borrowed, with explicit permission, from the University of New Hampshire InterOperability Laboratory (UNH-IOL).

The test definitions themselves are intended to provide a high-level description of the motivation, resources, procedures, and methodologies specific to each test. Formally, each test description contains the following sections:

Test Label: The test label and title constitute the first line of the test block. The test label is the concatenation of the short test suite name, group number, and the test number within the group, separated by periods.

Purpose: The Purpose is a brief statement outlining what the test attempts to achieve. It is usually phrased as a simple assertion of the feature or capability to be tested.

Device Type & Prerequisites: The Device Type & Prerequisites section notes for each part of the test what the prerequisite conditions are for the given Device Type.

Device Types	Prerequisite Condition
To be determined (TBD)	To be determined (TBD)
All	All
Boundary Clock (BC)	None
Ordinary Clock (OC)	Management message mechanism (Mgmt. Msg.)
Transparent Clock (TC)	Grandmaster-Capable (GMC)
	Slave-Only (SO)
	One-step Clock
	Two-step Clock
	Syntonized
	Non-Syntonized
	Peer-to-Peer
	End-to-End

References: The References section specifies all reference material external to the test plan, including the specific references for the test in question, and any other references that might be helpful in understanding the test methodology or test results. External sources are always referenced by a bracketed number (e.g. [1]) when mentioned in the test description. Any other references in the test description that are not indicated in this manner refer to elements within the test suite document itself (e.g. "Appendix 5.A" or "Table 5.1.1-1").

Resource Requirements: The Resource Requirements section specifies the test hardware and software needed to perform the test. This is generally expressed in terms of minimum requirements for abstract test gear. In some cases precise equipment requirements may be provided with examples of specific manufacturer/model information provided.

Modification History: The Modification History logs the changes for this test since its introduction.

Discussion: The Discussion is a general discussion of the test and relevant section of the specification, including any assumptions made in the design or implementation of the test as well as known limitations.

Test Setup: The setup section describes the initial configuration of the test environment. Elements of the test procedure may change the test environment as the test progresses.

Procedure: The procedure section contains the systematic instructions for carrying out the test. It provides a cookbook approach to testing, and may be interspersed with requirements to record observable results. These procedures should be the ideal test methodology, independent of specific tool limitations or restrictions.

Observable Results: This section lists the specific observable items that can be examined by the tester in order to verify that the DUT is operating properly. When multiple values for an observable are possible, this section provides a short discussion on how to interpret them. The determination of a pass or fail outcome for a particular test is based on the successful (or unsuccessful) detection of a specific observable. All test-part outcomes are presumed to initially be FAIL, and remain so if any single failure condition is met. Only if no fail conditions are met, and the explicitly stated pass conditions observed, will the test part outcome be deemed a PASS.

With the exception of N/A, WARN, and INFO, if a test part results in neither a PASS nor a FAIL outcome then that test part outcome is deemed a FAIL.

A strong preference is to have any part of a test err on the side of falsely failing a device rather than falsely passing the device. Whether through automation or manual execution, tests can have only one of five outcomes:

Outcome	Meaning
PASS	Test part meets all PASS criteria, with no FAIL or WARN conditions met.
FAIL	Test part meets at least one FAIL criterion, or fails to meet any criteria.
N/A	Test part is Not Applicable to the device.
WARN	Test part does not meet a failing criterion, but behavior is not recommended and warned against.
INFO	Test part has no pass/fail criteria, but the observation may have value to the device manufacturer or industry at large.

Possible Problems: This section contains a description of known issues with the test procedure, which may affect test results in certain situations.

Summary of Tests Prerequisites and Certification Classifier

Test PWR.c.1.1 – logAnnounceInterval

Part	Applies To Device Type	Prerequisite Conditions
A	BC, OC	
B	BC, OC	Mgmt. Msg.

Test PWR.c.1.2 – logSyncInterval

Part	Applies To Device Type	Prerequisite Conditions
A	BC, OC	
B	BC, OC	Mgmt. Msg.

Test PWR.c.1.3 – announceReceiptTimeout

Part	Applies To Device Type	Prerequisite Conditions
A	BC, OC	
B	BC, OC	Mgmt. Msg.

Test PWR.c.1.4 – logMinPdelayReqInterval

Part	Applies To Device Type	Prerequisite Conditions
A	All	
B	All	Mgmt. Msg.

Test PWR.c.1.5 – priority1 and priority2

Part	Applies To Device Type	Prerequisite Conditions
A	BC, OC	
B	BC, OC	Mgmt. Msg.

Test PWR.c.1.6 – slaveOnly

Part	Applies To Device Type	Prerequisite Conditions
A	BC, OC	Mgmt. Msg.

Test PWR.c.1.7 – domainNumber

Part	Applies To Device Type	Prerequisite Conditions
A	BC, OC	GMC
B	BC, OC	
C	BC, OC	
D	BC, OC	Mgmt. Msg.

Test PWR.c.1.8 – primaryDomain

Part	Applies To Device Type	Prerequisite Conditions
A	TC	Mgmt. Msg.

Test PWR.c.2.1 – Disqualified Announce Messages, by clockIdentity

Part	Applies To Device Type	Prerequisite Conditions
A	BC	One-step Clock
B	BC	Two-step Clock

Test PWR.c.2.2 – Disqualified Announce Messages, by Most Recent

Part	Applies To Device Type	Prerequisite Conditions
A	TC	SNMP, One-step Clock
B	TC	SNMP, Two-step Clock
C	TC	One-step Clock
D	TC	Two-step Clock

Test PWR.c.2.3 – Disqualified Announce Messages, by Foreign Master Window

Part	Applies To Device Type	Prerequisite Conditions
C	OC, BC	GMC

Test PWR.c.2.4 – Disqualified Announce Messages, by stepsRemoved

Part	Applies To Device Type	Prerequisite Conditions
A	All	One-step Clock

Test PWR.c.2.5 – Disqualified Announce Messages, by alternateMasterFlag

Part	Applies To Device Type	Prerequisite Conditions
A	All	Two-step Clock

Test PWR.c.2.6 – Data Set Comparison on a Single Port

Part	Applies To Device Type	Prerequisite Conditions
A	OC, BC	None
B	TC	Syntonized
C	TC	Non-Syntonized

Test PWR.c.2.7 – Data Set Comparison on Multiple Ports

Part	Applies To Device Type	Prerequisite Conditions
A	All	One-step Clock

Test PWR.c.2.8 – State Decision Algorithm

Part	Applies To Device Type	Prerequisite Conditions
A	All	Two-step Clock

Test PWR.c.2.9 – Steps Removed

Part	Applies To Device Type	Prerequisite Conditions
A	BC, OC	One-step Clock
B	BC, OC	Two-step Clock
C	TC	One-step Clock
D	TC	Two-step Clock

Test PWR.c.2.10 – Source Port Identity

Part	Applies To Device Type	Prerequisite Conditions
A	All	None
B	OC, BC	None
C	TC	None

Test PWR.c.5.1 – Frequency Accuracy

Part	Applies To Device Type	Prerequisite Conditions
A	BC, OC	GM, One-step
B	BC, OC	GM, Two-step

Test PWR.c.5.2 – Frequency Adjustment Range

Part	Applies To Device Type	Prerequisite Conditions
A	BC, OC	One-step, Peer-to-peer
B	BC, OC	Two-step, Peer-to-peer
C	BC, OC	End-to-end

Test PWR.c.7.1 – Mean Path Delay for Delay_Req

Part	Applies To Device Type	Prerequisite Conditions
A	OC, BC	GMC, One-step Clock
B	OC, BC	GMC, Two-step Clock
C	OC, BC	Slave-only, One-step Clock
D	OC, BC	Slave-only, Two-step Clock

Test PWR.c.8.1 – Independent Ports for Boundary Clocks

Part	Applies To Device Type	Prerequisite Conditions
A	BC	One-step Clock
B	BC	Two-step Clock

Test PWR.c.8.2 – Independent Ports for Transparent Clocks

Part	Applies To Device Type	Prerequisite Conditions
A	TC	SNMP, One-step Clock
B	TC	SNMP, Two-step Clock
C	TC	One-step Clock
D	TC	Two-step Clock

Test PWR.c.8.3 – Peer Delay Mechanism

Part	Applies To Device Type	Prerequisite Conditions
C	OC, BC	GMC

Test PWR.c.8.4 – Peer Delay Turnaround Timestamps, One-Step Clock

Part	Applies To Device Type	Prerequisite Conditions
A	All	One-step Clock

Test PWR.c.8.5 – Peer Delay Turnaround Timestamps, Two-Step Clock

Part	Applies To Device Type	Prerequisite Conditions
A	All	Two-step Clock

Test PWR.c.8.6 – Pdelay_Req Message Field Values

Part	Applies To Device Type	Prerequisite Conditions
A	OC, BC	None
B	TC	Syntonized
C	TC	Non-Syntonized

Test PWR.c.8.7 – Pdelay_Resp Message Field Values, One-Step Clock

Part	Applies To Device Type	Prerequisite Conditions
A	All	One-step Clock

Test PWR.c.8.8 – Peer Delay Message Field Values, Two-Step Clock

Part	Applies To Device Type	Prerequisite Conditions
A	All	Two-step Clock

Test PWR.c.8.9 – Mean Path Delay

Part	Applies To Device Type	Prerequisite Conditions
A	BC, OC	One-step Clock
B	BC, OC	Two-step Clock
C	TC	One-step Clock
D	TC	Two-step Clock

Test PWR.c.8.10 – Restriction on Peer Delay Mechanism

Part	Applies To Device Type	Prerequisite Conditions
A	All	None
B	OC, BC	None
C	TC	None

SECTION PWR.c: 1588 Default PTP Profiles

Overview:

This selection of tests verifies the various requirements for 1588 Default PTP Profile products defined in the IEEE 1588 standard. Separate sections will follow with tests on requirements specific to the Delay Request-Response Default PTP Profile and for the Peer-to-Peer Default PTP Profile.

Comments and questions regarding the documentation or implementation of these tests are welcome and may be sent to ptplab@iol.unh.edu.

Notes:

Successful completion of all tests contained in this suite does not guarantee that the tested device will successfully operate with other 1588 Default PTP Profile products. However, when combined with a satisfactory level of interoperability testing, these tests provide a reasonable level of confidence that the Device Under Test (DUT) will function properly in many 1588 Default PTP Profile environments.

GROUP 1: PTP Attribute Values

Overview:

Ordinary and boundary clocks have two types of data sets, referred to as clock data sets and port data sets. The clock data sets include defaultDS, currentDS, and parentDS. The defaultDS attributes describe the clock. The currentDS attributes relate to synchronization. The parentDS attributes describe the parent (the clock to which this clock synchronizes) and the grandmaster (the clock at the root of the master-slave hierarchy). There is only one port data set, namely portDS. The default attribute values are the configuration of a Precision Time Protocol (PTP) device as it is delivered from the manufacturer. In IEEE Std 1588-2008 all nodes support specific values for attributes.

The tests defined in this group validate specific attribute values from the clock data sets and port data sets. The values verified from the port data sets include the logAnnounceInterval, logSyncInterval, announceReceiptTimeout and logMinPdelayReqInterval. The values verified from the clock data sets include the priority1, priority2, slaveOnly, and domainNumber.

Notes:

Test PWR.c.1.1 – logAnnounceInterval

Purpose: To validate the DUT's logAnnounceInterval

Device Type Prerequisites and Certification Classifier:

Part	Applies To Device Type	Prerequisite Conditions
A	BC, OC	
B	BC, OC	Mgmt. Msg.

References: [1] IEEE Std 1588-2008: sub-clause J.4.2
[2] IEEE Std 1588-2008: sub-clause 9.5.8

Resource Requirements: One test station capable of transmitting and receiving arbitrary MAC frames

Modification History: 2014-01-06 Preview release

Discussion: This test will validate the DUT's logAnnounceInterval value by observing the logMessageInterval field and frequency of Announce messages emitted from the DUT. The time between successive Announce messages is represented as 2portDS.logAnnounceInterval seconds. Reference [1] states that the default value for portDS.logAnnounceInterval must be 1, hence Announce messages must be transmitted every 10 seconds. Reference [2] states that a node shall space Announce messages at ±30% of 2portDS.logAnnounceInterval seconds, with 90% confidence. This translates to a minimum value of 7 s and a maximum value of 13 s. Refer to Appendix D: Equations
to calculate the mean, variance, standard deviation and with 90% confidence. Common message headers are 34 octets long with the last octet being the logMessageInterval.

Test Setup: Refer to Appendix A: DEFAULT TEST SETUP with the following modifications: Connect all DUT ports other than DUT.TS1 (if any) to a device supporting 1588 (a copy of the DUT is acceptable). This has the effect of maximizing the processor load on the DUT's CPU for PTP packet processing. If a Bridge is used, ensure the added Bridge has RSTP enabled to break network loops.

Test Procedure:

Part A: Default Initialization Value

A:1. Ensure all DUT ports are linked as described in the Test Setup.
A:2. Capture traffic received by Test Station 1 (TS1) for the duration of this test.
A:3. Observe 60 consecutive Announce intervals.
A:4. Calculate the 90% confidence interval of the mean Announce interval from the observed samples using the calculations provided to Appendix D: Equations

a. If the interval calculated is not fully within the 7 and 13 seconds allowed range, repeat step A:3 and A:4, this time observe 600 Announce intervals.

A:5. Note any outliers. An outlier is considered to be any Announce intervals found to be greater than 3.5 s.
A:6. Observe the logMessageInterval value in 60 consecutive Announce messages.
A:7. If the device has more than one port, repeat steps A:1-2 for one other port on the device.

Observable Results:

Part:Step	Status	Description
A:3	FAIL	No Announce messages received.
A:4	FAIL	The 90% confidence interval for the mean of the observed Announce intervals is not fully within the 7 and 13 seconds allowed range.
A:5	FAIL	Any outliers are found to be greater than 3.5.
A:6	FAIL	The logMessageInterval value (1 octet at offset 33) in any Announce message is anything other than 1.
A:7	PASS	The 90% confidence interval for the mean of the observed Announce intervals is fully within 7 and 13 s; there are not outliers greater than 3.5; and, the value of the logMessageInterval observed in each Announce messages was 1.

Part B: LogAnnounceInt

B:1. Capture traffic received by TS1 for the duration of this test.
B:2. Observe the DUT's logAnnounceInterval by using the GET management action of the managementId value 2009 hex.
B:3. Wait 5 seconds, observe 60 consecutive Announce intervals.
B:4. Change the DUT's logAnnounceInterval by setting managementId value 2009 hex to 0.3.
B:5. Wait 5 seconds, observe 60 consecutive Announce intervals.
B:6. Observe the DUT's logAnnounceInterval by using the GET management action of the managementId value 2009 hex.

Observable Results:

Part:Step	Status	Description
B:2	FAIL	The value of the logAnnounceInterval requested was not 1.
B:3	FAIL	The average time between Announce messages is not between 7 and 13 s.
B:5	FAIL	The average time between Announce messages is not between 1.7 and 2.3 s.
B:5	FAIL	The logMessageInterval value (1 octet at offset 33) in any Announce message is anything other than 0.3.
B:6	PASS	The value of the logAnnounceInterval requested and observed in Announce messages was 0.3.

Possible Problems: None

Test PWR.c.1.2 – logSyncInterval

Purpose: To validate the DUT's logSyncInterval

Device Type Prerequisites and Certification Classifier:

Part	Applies To Device Type	Prerequisite Conditions
A	BC, OC	
B	BC, OC	Mgmt. Msg.

References: [1] IEEE Std 1588-2008: annex J.3.2
[2] IEEE Std 1588-2008: sub-clause 9.5.9

Resource Requirements: One test station capable of transmitting and receiving arbitrary MAC frames

Modification History: 2014-01-06 Preview release

Discussion: This test will validate the DUT's portDS.logSyncInterval value by observing the logSyncInterval field and frequency of Sync messages emitted from the DUT. The time between successive Sync messages is represented as $2^{portDS.logSyncInterval}$ seconds. Reference [1] states that the default value for portDS.logSyncInterval must be 0; hence Sync messages must be transmitted every 1 second. Reference [2] states that a node shall space Sync messages at ±30% of $2^{portDS.logSyncInterval}$ seconds, with 90% confidence. This translates to a minimum value of 0.7 s and a maximum value of 1.3 s. Refer to Appendix D: Equations
to calculate the mean, variance, standard deviation and with 90% confidence. Common message headers are 34 octets long with the last octet being the logMessageInterval.

Test Setup: Refer to Appendix A: DEFAULT TEST SETUP with the following modifications: Connect all DUT ports other than DUT.TS1 (if any) to a device supporting 1588 (a copy of the DUT is acceptable). This has the effect of maximizing the processor load on the DUT's CPU for PTP packet processing. If a Bridge is used, ensure the added Bridge has RSTP enabled to break network loops.

Test Procedure:

Part A: Default Initialization Value

A:1. Ensure all DUT ports are linked as described in the Test Setup.
A:2. Capture traffic received by TS1 for the duration of this test.
A:3. Observe 60 consecutive Sync intervals.
A:4. Calculate the 90% confidence interval of the mean Sync interval from the observed samples using the calculations provided in Appendix D: Equations.
    a. If the interval calculated is not fully within the 0.7 and 1.3 s allowed range, repeat step A:3 and A:4, this time observe 600 Sync intervals.
A:5. Observe the logMessageInterval value in 60 Sync messages.
A:6. If the device has more than one port, repeat steps A:1-2 for one other port on the device.

Observable Results:

Part:Step	Status	Description
A:4	FAIL	The 90% confidence interval for the mean of the observed Sync intervals is not fully within the 0.7 and 1.3 s allowed range.
A:6	FAIL	The logMessageInterval value (1 octet at offset 33) in any Sync message is anything other than 0.
A:7	PASS	The 90% confidence interval for the mean of the observed Sync intervals is fully within 0.7 and 1.3 s; and, the value of the logMessageInterval observed in each Sync messages was 0.

Part B: LogSyncInt

B:1. Capture traffic received by TS1 for the duration of this test.
B:2. Observe the DUT's logSyncInterval by using the GET management action of the managementId value 200B hex.
B:3. Wait 5 seconds, observe 60 consecutive Sync intervals.
B:4. Change the DUT's logSyncInterval by setting managementId value 200B hex to 0.3.
B:5. Wait 5 seconds, observe 60 consecutive Sync intervals.
B:6. Observe the DUT's logSyncInterval by using the GET management action of the managementId value 200B hex.

Observable Results:

Part:Step	Status	Description
B:2	FAIL	The value of the logSyncInterval requested and observed in Sync messages was not 0.
B:3	FAIL	The average time between Sync messages is not between 0.7 and 1.3 s.
B:5	FAIL	The average time between Sync messages is not between 1.7 and 2.3 s.
B:5	FAIL	The logMessageInterval value (1 octet at offset 33) in any Sync message is anything other than 0.3.
B:6	PASS	The value of the logSyncInterval requested and observed in Sync messages was 0.3.

Possible Problems: None

Test PWR.c.1.3 – announceReceiptTimeout

Purpose: To validate the DUT's announceReceiptTimeout.

Device Type Prerequisites and Certification Classifier:

Part	Applies To Device Type	Prerequisite Conditions
A	BC, OC	
B	BC, OC	Mgmt. Msg.

References: [1] IEEE Std 1588-2008: annex J.3.2
[2] IEEE Std 1588-2008: sub-clause 8.2.5.4.2

Resource Requirements: One test station capable of transmitting and receiving arbitrary MAC frames

Modification History: 2013-06-12 Preview release

Discussion: This test will validate the DUT's portDS.announceReceiptTimeout value by requesting the AnnounceReceiptTimeout MIB object. The announceReceiptTimeout specifies the number of announceIntervals that have to pass without receipt of an Announce message before the occurrence of the event ANNOUNCE_RECEIPT_TIMEOUT_EXPIRES. The value of portDS.announceReceiptTimeout shall be an integral multiple of announceInterval [1]. By default it must be 3 with a configurable range from 2 to 10 [2].

This test will also detect the DUT's portDS.announceReceiptTimeoutInterval by sending a constant stream of "better" Announce messages to the DUT then increasing the gap between the Announce messages until the DUT is seen to resume sending Announce messages. The Default Profile Announce interval is 10 s, thus for this test the announceReceiptTimeoutInterval must be 30 s.

Test Setup: Refer to Appendix A: DEFAULT TEST SETUP.

Test Procedure:

Part A: Slow Rate of Announce Messages

A:1. Capture traffic received by TS1 for the duration of this test.
A:2. Wait up to 40 s or for 3 Announce messages to be received from the DUT, and observe its priority1 value.
A:3. Send an Announce message every N seconds with a lower (better) priority1 value so that the DUT becomes a slave to TS1. N is initially 10.
A:4. Wait 30 s.
A:5. Observe whether any Announce messages come from DUT.TS1.
A:6. Repeat steps A:3 through A:5, increasing N by 5 until step A:5 observes Announce messages coming from the DUT, or when N is 50 (5).

Observable Results:

Part:Step	Status	Description
A:2	FAIL	No Announce messages are received.
A:5	FAIL	When N is 10-25 the DUT sends Announce messages.
A:5	FAIL	N reaches 35 and the DUT still does not send Announce messages.
A:5	PASS	When N is 30 or 35 the DUT sends Announce messages.

Part B: AnnounceReceiptTimeout

B:1. Observe the DUT's AnnounceReceiptTimeout by using the GET management action of the managementId value 200A hex.
B:2. Change the DUT's AnnounceReceiptTimeout by setting managementId value 200A hex to 4.
B:3. Wait 20 s, observe the DUT's AnnounceReceiptTimeout by using the GET management action of the managementId value 200A hex.

Observable Results:

Part:Step	Status	Description
B:1	FAIL	The AnnounceReceiptTimeout value is not 3.
B:3	PASS	The AnnounceReceiptTimeout value is 4.

Possible Problems: None

Test PWR.c.1.4 – logMinPdelayReqInterval

Purpose: To the DUT's logMinPdelayReqInterval

Device Type Prerequisites and Certification Classifier:

Part	Applies To Device Type	Prerequisite Conditions
A	All	
B	All	Mgmt. Msg.

References: [1] IEEE Std 1588-2008: sub-clause 9.5.13
[2] IEEE Std 1588-2008: annex J.3.2

Resource Requirements: One test station capable of transmitting and receiving arbitrary MAC frames

Modification History: 2014-01-07 Preview release

Discussion: This test will validate the DUT's logMinPdelayReqInterval value by observing the frequency of Pdelay_Req messages and the logMessageInterval field of Pdelay_Req messages from the DUT. The time between successive Pdelay_Req messages must be no less than $2^{\text{portDS.logMinPdelayReqInterval}}$ seconds. Reference [1] states that the default value for portDS.logMinPdelayReqInterval must be 0. Reference [2] states that a node shall space Pdelay_Req messages at no less than $2^{\text{portDS.logMinPdelayReqInterval}}$ seconds on average. Common message headers are 34 octets long with the last octet being the logMessageInterval.

Test Setup: Refer to Appendix A: DEFAULT TEST SETUP with the following modifications: Connect all DUT ports other than DUT.TS1 (if any) to a device supporting Power Profile (a copy of the DUT is acceptable). This has the effect of maximizing the processor load on the DUT's CPU for PTP packet processing. If a Bridge is used, ensure the added Bridge has RSTP enabled to break network loops.

Test Procedure:

Part A: Default Initialization Value

A:1. Ensure all DUT ports are linked as described in the Test Setup.
A:2. Capture traffic received by TS1 for the duration of this test.
A:3. Observe 10 consecutive Pdelay_Req intervals.
A:4. Calculate the 90% confidence interval of the mean Pdelay_Req interval from the observed samples using the calculations provided in Appendix D: Equations
   a. If the interval calculated is not fully greater than 0.7 s, repeat step A:3 and A:4, this time observe 600 Pdelay_Req intervals.
A:5. If the device has more than one port, repeat steps A:1-2 for one other port on the device.

Observable Results:

Part:Step	Status	Description
A:4	FAIL	The 90% confidence interval for the mean of the observed Pdelay_Req intervals is not fully greater than 0.7 s.
A:5	PASS	The 90% confidence interval for the mean of the observed Pdelay_Req intervals is fully greater than 0.7 s.

Part B: logMinPdelayReqInterval

B:1. Capture traffic received by TS1 for the duration of this test.
B:2. Observe 10 consecutive Pdelay_Req intervals.
B:3. If the device has more than one port, repeat steps B:1-2 for one other port on the device.
B:4. Observe the DUT's logMinPdelayReqInterval by using the GET management action of the managementId value 6001 hex.
B:5. Change the DUT's logMinPdelayReqInterval to 0.3 by using the SET management action of the managementId value 6001 hex.
B:6. Wait 5 seconds, observe 10 consecutive Pdelay_Req intervals.
B:7. Observe the DUT's logMinPdelayReqInterval by using the GET management action of the managementId value 6001 hex.

Observable Results:

Part:Step	Status	Description
B:2	FAIL	The average time between Pdelay_Req messages is less than 0.7 s.
B:4	FAIL	The value of the logMinPdelayReqInterval requested and observed was not 0.
B:6	FAIL	The average time between Pdelay_Req messages is less than 1.7 s.
B:7	PASS	The value of the logMinPdelayReqInterval requested and observed was not 0.3.

Possible Problems: None

Test PWR.c.1.5 – priority1 and priority2

Purpose: To validate the DUT's priority1 and priority2

Device Type Prerequisites and Certification Classifier:

Part	Applies To Device Type	Prerequisite Conditions
A	BC, OC	
B	BC, OC	Mgmt. Msg.

References: [1] IEEE Std 1588-2008: annex J.3.2

Resource Requirements: One test station capable of transmitting and receiving arbitrary MAC frames

Modification History: 2014-01-07 Preview release

Discussion: This test will validate the DUT's defaultDS.priority1 and defaultDS.priority2 values by observing the priority1 and priority2 fields of Announce messages emitted from the DUT and by management-message request. At offset 47 and 52 of Announce messages, the grandmasterPriority1 and grandmasterPriority2 fields are each one octet long. The default value of priority1 and priority2 shall be 128 [1].

Test Setup: Refer to Appendix A: DEFAULT TEST SETUP.

Test Procedure:

Part A: Default Initialization Value

A:1. Ensure that the DUT is in default setup.
A:2. Capture traffic received by TS1 for the duration of this test.
A:3. Wait up to 30 seconds or for 3 Announce messages to be received from the DUT.

Observable Results:

Part:Step	Status	Description
A:3	FAIL	Three Announce messages are not received within 10 seconds.
A:3	FAIL	The grandmasterPriority1 field is not 128.
A:3	FAIL	The grandmasterPriority2 field is not 128.
A:3	PASS	The grandmasterPriority1 and the grandmasterPriority2 fields in each Announce message were 128.

Part B: Priority1and Priority2

B:1. Ensure that the DUT is in default setup.
B:2. Observe the DUT's priority1 and priority2 by using the GET management action of the managementId value 2005 and 2006 hex.
B:3. Change the DUT's priority1 and priority2 values to 130 by using the SET management action of the managementId value 2005 and 2006 hex.
B:4. Wait 5 seconds; observe the DUT's priority1 and priority2 by using the GET management action of the managementId value 2005 and 2006 hex.

Observable Results:

Part:Step	Status	Description
B:2	FAIL	The value of the priority1 field is not 128.
B:2	FAIL	The value of the priority2 field is not 128.
B:4	FAIL	The value of the priority1 field is not 130.
B:4	FAIL	The value of the priority2 field is not 130.
B:4	PASS	The GMC DUT's priority1 and priority2 started as 128 and changed to 130.

Possible Problems: None

Test PWR.c.1.6 – slaveOnly

Purpose: To validate the DUT's slaveOnly

Device Type Prerequisites and Certification Classifier:

Part	Applies To Device Type	Prerequisite Conditions
A	BC, OC	Mgmt. Msg.

References: [1] IEEE Std 1588-2008: sub-clause 9.2.3
[2] IEEE Std 1588-2008: annex J.3.2

Resource Requirements: One test station capable of transmitting and receiving arbitrary MAC frames

Modification History: 2014-01-07 Preview release

Discussion: This test will validate the DUT's defaultDS.slaveOnly value by management-message request. Ordinary clocks may be designed to be slave-only or grandmaster-capable clocks while boundary clocks cannot be slave-only [1]. If the clock is slave-only the value of defaultDS.slaveOnly shall be TRUE. Otherwise the clock is grandmaster-capable, and the value of defaultDS.slaveOnly shall be FALSE [2].

Test Setup: Refer to Appendix A: DEFAULT TEST SETUP.

Test Procedure:

Part A: Default Initialization Value

A:1. Ensure that the DUT is in default setup.
A:2. Observe the DUT's slaveOnly by using the GET management action of the managementId value 2008 hex.
A:3. Change the DUT's slaveOnly to TRUE by using the SET management action of the managementId value 2008 hex.
A:4. Wait 5 seconds, and then observe the DUT's slaveOnly by using the GET management action of the managementId value 2008 hex.

Observable Results:

Part:Step	Status	Description
A:2	FAIL	The slaveOnly field is TRUE.
A:4	FAIL	The slaveOnly field is FALSE.
A:4	PASS	The slaveOnly field changed from FALSE to TRUE.

Possible Problems: None

Test PWR.c.1.7 – domainNumber

Purpose: To validate the DUT's domainNumber

Device Type Prerequisites and Certification Classifier:

Part	Applies To Device Type	Prerequisite Conditions
A	BC, OC	GMC
B	BC, OC	
C	BC, OC	
D	BC, OC	Mgmt. Msg.

References: [1] IEEE Std 1588-2008: sub-clause 8.2.1.4.3
[2] IEEE Std 1588-2008: annex J.3.2

Resource Requirements: One test station capable of transmitting and receiving arbitrary MAC frames

Modification History: 2014-01-07 Preview release

Discussion: This test will validate the DUT's defaultDS.domainNumber value by observing the domainNumber field of Announce messages and Pdelay_Resp messages emitted from the DUT, and also by management-message request. Common message headers are 34 octets long with the 5^{th} octet stating the domainNumber [1]. The domainNumber can be configured to an integer in the range of 0 to 127, but by default the value must be initialized to 0 [2].

Test Setup: Refer to Appendix A: DEFAULT TEST SETUP.

Test Procedure:
Part A: Domain Number in Sync and Follow_Up messages
A:1. Ensure that the DUT is in default setup.
A:2. Capture all PTP messages from the DUT.
A:3. From TS1, send one Pdelay_Req message to the DUT every second.
A:4. Wait until TS1 receives an Announce message.
A:5. Wait at least 20 seconds. Verify the domainNumber of all received PTP messages from the DUT.

Observable Results:

Part:Step	Status	Description
A:5	FAIL	No Sync or Follow_Up (if two-step) messages are received.
A:5	FAIL	Any of the received PTP messages have a domainNumber other than 0.
A:5	PASS	All observed PTP messages have a domainNumber of 0.

Part B: Domain Number in Pdelay_Req, Announce, Pdelay_Resp, and Pdelay_Resp_Follow_Up messages
B:1. Ensure that the DUT is in default setup.
B:2. Capture all PTP messages from the DUT.
B:3. From TS1, send one Pdelay_Req message to the DUT every second.
B:4. Wait for the arrival of at least one of each of the following message types: Pdelay_Req, Announce, Pdelay_Resp, and if the DUT is two-step, Pdelay_Resp_Follow_Up. Observe the domainNumber of each.

Observable Results:

Part:Step	Status	Description
B:4	FAIL	TS1 does not receive each of the message types.
B:4	FAIL	Any of the received PTP messages has a domainNumber other than 0.
B:4	PASS	All observed PTP messages have a domainNumber of 0.

Part C: Not Accepting messages from other Domains

C:1. Ensure that the DUT is in default setup.
C:2. Capture traffic received and sent by TS1 for the duration of this test.
C:3. Have TS1 send Pdelay_Req messages with the value of the Domain Number 0.
C:4. Wait up to 10 seconds or for 3 Pdelay_Resp Messages to be received from the DUT.
C:5. Have TS1 send Pdelay_Req messages with the value of the Domain Number 2.
C:6. Wait up to 10 seconds or for 3 Pdelay_Resp Messages to be received from the DUT.

Observable Results:

Part:Step	Status	Description
C:4	FAIL	A Pdelay_Resp message is not received from the DUT.
C:6	FAIL	A Pdelay_Resp message is received from the DUT.
C:6	PASS	The DUT only accepts and responds to messages with a Domain Number 0.

Part D: Default Initialization Value

A:1. Ensure that the DUT is in default setup.
D:1. Capture traffic received by TS1 for the duration of this test.
D:2. Wait up to 30 seconds or for 3 Announce messages to be received from the DUT.
D:3. Observe the DUT's domain number by using the GET management action of the managementId value 2007 hex.
D:4. Change the DUT's domain number to 100 by using the SET management action of the managementId value 2007 hex.
D:5. Wait up to 10 seconds or for 3 Announce messages to be received from the DUT.
D:6. Observe the DUT's domain number by using the GET management action of the managementId value 2007 hex.

Observable Results:

Part:Step	Status	Description
D:3	FAIL	Three Announce messages are not received.
D:3	FAIL	The domainNumber is not 0.
D:4	FAIL	The domainNumber is not 0.
D:6	FAIL	The domainNumber is not 100.
D:7	FAIL	The domainNumber is not 100.
D:7	PASS	The value of the domain number requested and observed in Announce messages changed from 0 to 100.

Possible Problems: None

Test PWR.c.1.8 – primaryDomain

Purpose: To validate the DUT's primaryDomain

Device Type Prerequisites and Certification Classifier:

Part	Applies To Device Type	Prerequisite Conditions
A	TC	Mgmt. Msg.

References: [1] IEEE Std 1588-2008: sub-clause 8.3.2.3.2
[2] IEEE Std 1588-2008: annex J.3.2

Resource Requirements: One test station capable of transmitting and receiving arbitrary MAC frames

Modification History: 2014-01-07 Preview release

Discussion: This test will validate the DUT's transparentClockdefaultDS.primaryDomain value by management-message request. The domainNumber of the primary syntonization domain is stored as the transparentClockdefaultDS.primaryDomain value [1]. When initialized, the value shall be zero [2].

Test Setup: Refer to Appendix A: DEFAULT TEST SETUP.

Test Procedure:

A:1. Ensure that the DUT is in default setup.
A:2. Observe the DUT's primary domain number by using the GET management action of the managementId value 4002 hex.
A:3. Change the DUT's domain number to 100 by using the SET management action of the managementId value 4002 hex.
A:4. Observe the DUT's domain number by using the GET management action of the managementId value 4002 hex.

Observable Results:

Part:Step	Status	Description
A:2	FAIL	The primary domain number is not 0.
A:4	FAIL	The primary domain number is not 100.
A:4	PASS	The primary domain number did not change from 0 to 100.

Possible Problems: None

GROUP 2: Best Master Clock Algorithm

Overview:

This group covers requirements defined in IEEE 1588-2008 sub-clause 9.3 “Best master clock algorithm”, especially sections 9.3.2 (BMC algorithm), 9.3.3 (State decision algorithm) and 9.3.4 (Data set comparison algorithm). The best master clock algorithm comprises two parts: a data set comparison algorithm followed by a state decision algorithm.

The tests defined in this group validate the data set comparison algorithm, defined in sub-clause 9.3.4. These tests change various fields in the Announce messages originating from the test station (TS) to verify that the device under test (DUT) selects the proper grandmaster clock. Verification of best master clock selection is determined through observation of the DUT’s Announce message behavior and, if accessible, the DUT’s parentDS data set.

Notes:

Test PWR.c.2.1 – Disqualified Announce Messages, by clockIdentity

Purpose: To verify that the DUT selects the correct grandmaster clock based on the validity of the clockIdentity field in incoming Announce messages.

Device Type Prerequisites and Certification Classifier:

Part	Applies To Device Type	Prerequisite Conditions
A	OC, BC	Means of observing the DUT's grandmaster

References: [1] IEEE Std 1588-2008: sub-clause 9.3.2.5

Resource Requirements: One test station capable of transmitting and receiving arbitrary MAC frames

Modification History: 2013-02-26 Preview release

Discussion: The test will verify that the DUT disqualifies Announce messages sent and received by the same port [1]. To observe whether the DUT has disqualified an incoming Announce message we establish a connection, and then we send it both valid and invalid Announce messages. The valid Announce messages will use sourcePortIdentity.clockIdentity values that differ from the DUT's sourcePortIdentity while the unqualified Announce messages will use sourcePortIdentity.clockIdentity values that are the same as the DUT's sourcePortIdentity.clockIdentity values. If the DUT selects to the grandmaster that sent the invalid Announce messages then it did not disqualify the appropriate messages.

Test Setup: Refer to Appendix A: DEFAULT TEST SETUP.

Test Procedure:

Part A: Disqualified Announce messages, by clockIdentity

A:1. Capture traffic received by TS1 for the duration of this test.

A:2. For the duration of this test, have TS1 send Announce messages with all fields that influence the BMCA, other than the grandmasterIdentity, grandmasterPriority1 and sourcePortIdentity.clockIdentity fields, identical to the DUT's data sets. For the grandmasterIdentity field use the value 0x102233fffe44556**6**. For the grandmasterPriority1 field use a value one less than the DUT's grandmasterPriority1. For the sourcePortIdentity.clockIdentity field, use a value differing from the DUT's clockIdentity.

A:3. For the duration of this test, have TS1 also send, intermingled with the above Announce messages, a distinct stream of different Announce messages with all fields that influence the BMCA, other than the grandmasterIdentity, grandmasterPriority1 and sourcePortIdentity.clockIdentity fields, identical to the corresponding fields in DUT's data sets. For the grandmasterIdentity field use the value 0x102233fffe44556**7**. For the grandmasterPriority1 field use a value two less than the DUT's grandmasterPriority1. For the sourcePortIdentity.clockIdentity field, use the DUT's clockIdentity.

A:4. After 5 seconds, observe the DUT's grandmaster,
   a. by using the GET management action of the managementId value 2002 hex, or
   b. by other means, if provided.

A:5. If the device has more than one port, repeat steps A:1-4 for one other port on the device.

Observable Results:

Part:Step	Status	Description
A:4	FAIL	The DUT's grandmasterIdentity is 0x102233fffe44556**7**.
A:4	FAIL	The DUT's grandmasterIdentity is not 0x102233fffe44556**6**.
A:4	PASS	The DUT disqualified (and ignored) Announce messages that used its clockIdentity, and the DUT qualified (and reacted to) Announce messages that did not use its clockIdentity.

Possible Problems: None

Test PWR.c.2.2 – Disqualified Announce Messages, by Most Recent

Purpose: To verify that the DUT disqualifies Announce messages that are not the most recently received from a given clock.

Device Type Prerequisites and Certification Classifier:

Part	Applies To Device Type	Prerequisite Conditions
A	OC, BC	Mgmt. Msg. or means of observing the DUT's grandmaster

References: [1] IEEE Std 1588-2008: sub-clause 9.3.2.5

Resource Requirements: One test station capable of transmitting and receiving arbitrary MAC frames

Modification History: 2013-02-26 Preview release

Discussion: The test will verify that the DUT disqualifies Announce messages that are not the most recently received from a given clock [1]. The DUT will receive a stream of Announce messages with higher grandmasterPriority1 values than that of the DUT's, indicating that TS1 should not be grandmaster. However, the last Announce message in the stream will receive will have a grandmasterPriority1 lower than the DUT's, indicating that TS1 should be made grandmaster. If the DUT only considers the most recent Announce message received it will make TS1 its grandmaster.

Test Setup: Refer to Appendix A: DEFAULT TEST SETUP.

Test Procedure:

Part A: Disqualified Announce messages, by most recent

A:1. Capture traffic received by TS1 for the duration of this test.

A:2. For the duration of this test, after each message is sent observe the DUT's grandmaster,
- a. by using the GET management action of the managementId value 2002 hex, or
- b. by other means, if provided

A:3. Have TS1 send a stream of 10 Announce messages, one each second for 10 seconds, with all fields that influence the BMCA, other than the grandmasterIdentity and grandmasterPriority1 fields, identical to the corresponding fields in the DUT's data sets. For the grandmasterIdentity field use the value 0x102233fffe445566.

A:4. For the grandmasterPriority1 field in the first 9 messages use a value one greater than the DUT's grandmasterPriority1. Observe the DUT's grandmasterIdentity between one and two seconds after each Announce message is received.

A:5. For the grandmasterPriority1 field in the last Announce message use a value one less than the DUT's grandmasterPriority1. Observe the DUT's grandmasterIdentity between one and two seconds after the Announce message is received.

A:6. If the device has more than one port, repeat steps A:1-5 for one other port on the device.

Observable Results:

Part:Step	Status	Description
A:4	FAIL	The DUT's grandmasterIdentity is ever 0x102233fffe445566.
A:5	FAIL	The DUT's grandmasterIdentity is not 0x102233fffe445566.
A:5	PASS	The DUT disqualified Announce messages that were not the most recently received, and the DUT qualified the Announce message that was the most recently received.

Possible Problems: None

Test PWR.c.2.3 – Disqualified Announce Messages, by Foreign Master Window

Purpose: To verify that the DUT disqualifies Announce messages that were not preceded by at least one recent Announce message from the same clock.

Device Type Prerequisites and Certification Classifier:

Part	Applies To Device Type	Prerequisite Conditions
A, B	OC, BC	Mgmt. Msg. or means of observing the DUT's grandmaster

References: [1] IEEE Std 1588-2008: sub-clause 9.3.2.5
[2] IEEE Std 1588-2008: sub-clause 9.3.2.4.5

Resource Requirements: One test station capable of transmitting and receiving arbitrary MAC frames

Modification History: 2013-02-27 Preview release

Discussion: The test will verify that the DUT disqualifies an Announce message that is the only non-identical Announce message received in a single foreign master time window [1].

The central requirement is that if at least FOREIGN_MASTER_THRESHOLD (2) Announce messages have not been received within FOREIGN_MASTER_TIME_WINDOW (4 announceIntervals) then the Announce message is disqualified. Once qualification occurs the clock shall be considered in the BMC algorithm.

Reference [2] states that the size of foreignMasterDS shall be at least five records. If five new clocks arrive on the network and all begin sending Announce messages sufficiently frequently then the Announce messages from all five clocks must be considered in the BMC algorithm.

To observe whether the DUT has disqualified an incoming Announce message we vary how often Announce messages are emitted from TS1. This test ensures that when Announce messages arrive more frequently than every four seconds the DUT qualifies them and that when Announce messages arrive less frequently than every four seconds the DUT disqualifies them.

Test Setup: Refer to Appendix A: DEFAULT TEST SETUP.

Test Procedure:

Part A: Disqualified Announce messages, by Foreign Master Window

A:1. Capture traffic received by TS1 for the duration of this test.

A:2. For the duration of this test have TS1 send Announce messages with all fields that influence the BMCA, other than the grandmasterIdentity and grandmasterPriority1 fields, identical to the DUT's data sets. For the grandmasterIdentity field use the value 0x102233fffe445566. For the grandmasterPriority1 field use a value one less than the DUT's grandmasterPriority1.

A:3. Send the Announce messages once every four announceIntervals.

A:4. After 10 seconds, observe the DUT's grandmaster,
   a. by using the GET management action of the managementId value 2002 hex, or
   b. by other means, if provided.

A:5. Increase the rate at which the Announce messages are sent to once every two announceIntervals.

A:6. After 10 seconds, observe the DUT's grandmaster,
   a. by using the GET management action of the managementId value 2002 hex, or
   b. by other means, if provided.

A:7. Increase the rate at which the Announce messages are sent to once each announceInterval.

A:8. After 10 seconds, observe the DUT's grandmaster,
   a. by using the GET management action of the managementId value 2002 hex, or
   b. by other means, if provided.

A:9. If the device has more than one port, repeat steps A:1-8 for one other port on the device.

Observable Results:

Part:Step	Status	Description
A:4	FAIL	The DUT's grandmasterIdentity is 0x102233fffe445566.
A:6	FAIL	The DUT's grandmasterIdentity is not 0x102233fffe445566.
A:8	FAIL	The DUT's grandmasterIdentity is not 0x102233fffe445566.
A:8	PASS	The DUT disqualified Announce messages that are the only non-identical Announce messages received in a single foreign master time window, and the DUT qualified Announce messages that are not the only non-identical Announce messages received in a single foreign master time window.

Part B: Five Foreign Masters

B:1. For the duration of this test have TS1 send five simultaneous yet distinct streams of Announce messages with all fields that influence the BMCA other than the grandmasterIdentity and the grandmasterPriority1 fields, identical to the DUT's data sets.

B:2. For the first stream, send Announce messages **five** times every four announce intervals (i.e., once every 800 ms) with the following values:
   a. grandmasterIdentity = 0x102233fffe44556**5**
   b. grandmasterPriority1 = **one** less that the DUT's grandmasterPriority1 field

B:3. For the second stream, send Announce messages **four** times every four announce intervals (i.e., once every 1,000 ms) with the following values:
   a. grandmasterIdentity = 0x102233fffe44556**6**
   b. grandmasterPriority1 = **two** less that the DUT's grandmasterPriority1 field

B:4. For the third stream, send Announce messages **three** times every four announce intervals (i.e., once every 1,300 ms) with the following values:
   a. grandmasterIdentity = 0x102233fffe44556**7**
   b. grandmasterPriority1 = **three** less that the DUT's grandmasterPriority1 field

B:5. For the fourth stream, send Announce messages **twice** every four announce intervals (i.e., once every 2,000 ms) with the following values:
   a. grandmasterIdentity = 0x102233fffe44556**8**
   b. grandmasterPriority1 = **four** less that the DUT's grandmasterPriority1 field

B:6. For the fifth stream, send Announce messages **once** every four announce intervals (i.e., once every 4,000 ms) with the following values:
   a. grandmasterIdentity = 0x102233fffe44556**9**
   b. grandmasterPriority1 = **five** less that the DUT's grandmasterPriority1 field

B:7. After 10 seconds, observe the DUT's grandmaster,
   a. by using the GET management action of the managementId value 2002 hex, or
   b. by other means, if provided

B:8. If the device has more than one port, repeat steps B:1-7 for one other port on the device.

Observable Results:

Part:Step	Status	Description
B:7	FAIL	The DUT's grandmasterIdentity is 0x102233fffe445565, 0x102233fffe445566, 0x102233fffe445567 or 0x102233fffe445569.
B:7	FAIL	The DUT's grandmasterIdentity is not 0x102233fffe445568.
B:7	PASS	The DUT has a foreignMasterDS data set with a minimum capacity of five foreign master records.

Possible Problems: None

Test PWR.c.2.4 – Disqualified Announce Messages, by stepsRemoved

Purpose: To verify that the DUT selects the correct grandmaster clock based on the validity of the stepsRemoved in incoming Announce messages.

Device Type Prerequisites and Certification Classifier:

Part	Applies To Device Type	Prerequisite Conditions
A, B	OC, BC	No Mgmt. Msg., Means of observing the DUT's grandmaster
C, D	OC, BC	Mgmt. Msg.

References: [1] IEEE Std 1588-2008: sub-clause 9.3.2.5

Resource Requirements: Two test stations capable of transmitting and receiving arbitrary MAC frames

Modification History: 2013-02-26 Preview release

Discussion: The test will verify the DUT disqualifies Announce messages with the value 255 or greater in the stepsRemoved field for the BMC algorithm [1]. The stepsRemoved field is a 16-bit field, therefore the largest possible value is 65,535 ($2^{16} – 1$). To observe whether the DUT has disqualified an incoming Announce message we establish a connection in which the DUT is sending Announce messages, and then we send it Announce messages that could change its state and cause it to stop sending Announce messages.

Test Setup: Refer to Appendix A: DEFAULT TEST SETUP.

Test Procedure:

Part A: Disqualification by stepsRemoved equal to 255

A:1. Capture traffic received by TS1 for the duration of this test.

A:2. Send to the DUT Announce messages that use a stepsRemoved value of **255** (0x00FF), grandmasterIdentity value of 0x102233fffe44556**6** and that use a lower (better) priority1 value than the corresponding value of the DUT's data set.

A:3. After 5 seconds, observe the DUT's grandmaster,
   a. by using the GET management action of the managementId value 2002 hex, or
   b. by other means, if provided

A:4. Send to the DUT Announce messages that use a stepsRemoved value of **254** (0x00FE), grandmasterIdentity value of 0x102233fffe44556**6** and that use a lower (better) priority1 value than the corresponding value of the DUT's data set.

A:5. After 5 seconds, observe the DUT's grandmaster,
   a. by using the GET management action of the managementId value 2002 hex, or
   b. by other means, if provided

A:6. If the device has more than one port, repeat steps A:1-6 for one other port on the device.

Observable Results:

Part:Step	Status	Description
A:3	FAIL	The DUT's grandmasterIdentity is 0x102233fffe44556**6**.
A:5	FAIL	The DUT's grandmasterIdentity is not 0x102233fffe44556**6**.
A:5	PASS	The DUT disqualified (and ignored) Announce messages whose stepsRemoved value was too high, and the DUT qualified (and reacted to) Announce messages whose stepsRemoved value was valid.

Part B: Disqualification by stepsRemoved greater than 255

B:1. Capture traffic received by TS1 for the duration of this test.

B:2. Send to the DUT Announce messages that use a stepsRemoved value of **65,535** (0xFFFF), grandmasterIdentity value of 0x102233fffe44556**6** and that use a lower (better) priority1 value than the corresponding value of the DUT's data set.

B:3. After 5 seconds, observe the DUT's grandmaster,
  a. by using the GET management action of the managementId value 2002 hex, or
  b. by other means, if provided

B:4. Send to the DUT Announce messages that use a stepsRemoved value of **10** (0x000A), grandmasterIdentity value of 0x102233fffe44556**6** and that use a lower (better) priority1 value than the corresponding value of the DUT's data set.

B:5. After 5 seconds, observe the DUT's grandmaster,
  a. by using the GET management action of the managementId value 2002 hex, or
  b. by other means, if provided

B:6. If the device has more than one port, repeat steps B:1-6 for one other port on the device.

Observable Results:

Part:Step	Status	Description
B:3	FAIL	The DUT's grandmasterIdentity is 0x102233fffe44556**6**.
B:5	FAIL	The DUT's grandmasterIdentity is not 0x102233fffe44556**6**.
B:5	PASS	The DUT disqualified (and ignored) Announce messages whose stepsRemoved value was too high, and the DUT qualified (and reacted to) Announce messages whose stepsRemoved value was valid.

Part C: Disqualification by stepsRemoved equal to 255, SNMP

C:1. Capture traffic received by TS1 for the duration of this test.

C:2. Send to the DUT Announce messages that use a stepsRemoved value of **255** (0x00FF), grandmasterIdentity value of 0x102233fffe44556**6** and that use a lower (better) priority1 value than the corresponding value of the DUT's data set.

C:3. After 5 seconds, observe the DUT's stepsRemoved by using the GET management action of the managementId value 2001 hex.

C:4. Observe the DUT's grandmaster by using the GET management action of the managementId value 2002 hex.

C:5. Send to the DUT Announce messages that use a stepsRemoved value of **254** (0x00FE), grandmasterIdentity value of 0x102233fffe44556**6** and that use a lower (better) priority1 value than the corresponding value of the DUT's data set.

C:6. After 5 seconds, observe the DUT's stepsRemoved by using the GET management action of the managementId value 2001 hex

C:7. Observe the DUT's grandmaster by using the GET management action of the managementId value 2002 hex.

C:8. If the device has more than one port, repeat steps C:1-6 for one other port on the device.

Observable Results:

Part:Step	Status	Description
C:3	FAIL	The DUT's stepsRemoved is not 0x00FF.
C:4	FAIL	The DUT's grandmasterIdentity is 0x102233fffe44556**6**.
C:6	FAIL	The DUT's stepsRemoved is not 0x00FE.
C:7	FAIL	The DUT's grandmasterIdentity is not 0x102233fffe44556**6**.
C:7	PASS	The DUT disqualified (and ignored) Announce messages whose stepsRemoved value was too high, and the DUT qualified (and reacted to) Announce messages whose stepsRemoved value was valid.

Part D: Disqualification by stepsRemoved greater than 255, SNMP

D:1. Capture traffic received by TS1 for the duration of this test.

D:2. Send to the DUT Announce messages that use a stepsRemoved value of **65,535** (0xFFFF), grandmasterIdentity value of 0x102233fffe44556**6** and that use a lower (better) priority1 value than the corresponding value of the DUT's data set.

D:3. After 5 seconds, observe the DUT's stepsRemoved by using the GET management action of the managementId value 2001 hex.

D:4. Observe the DUT's grandmaster by using the GET management action of the managementId value 2002 hex.

D:5. Send to the DUT Announce messages that use a stepsRemoved value of **10** (0x000A), grandmasterIdentity value of 0x102233fffe44556**6** and that use a lower (better) priority1 value than the corresponding value of the DUT's data set.

D:6. After 5 seconds, observe the DUT's stepsRemoved by using the GET management action of the managementId value 2001 hex.

D:7. Observe the DUT's grandmaster by using the GET management action of the managementId value 2002 hex.

D:8. If the device has more than one port, repeat steps D:1-6 for one other port on the device.

Observable Results:

Part:Step	Status	Description
D:3	FAIL	The DUT's stepsRemoved is not 0xFFFF.
D:4	FAIL	The DUT's grandmasterIdentity is 0x102233fffe44556**6**.
D:6	FAIL	The DUT's stepsRemoved is not 0x000A.
D:7	FAIL	The DUT's grandmasterIdentity is not 0x102233fffe44556**6**.
D:7	PASS	The DUT disqualified (and ignored) Announce messages whose stepsRemoved value was too high, and the DUT qualified (and reacted to) Announce messages whose stepsRemoved value was valid.

Possible Problems: None

Test PWR.c.2.5 – Disqualified Announce Messages, by alternateMasterFlag

Purpose: To verify that the DUT selects the correct grandmaster clock based on the alternateMasterFlag of incoming Announce messages.

Device Type Prerequisites and Certification Classifier:

Part	Applies To Device Type	Prerequisite Conditions
A	OC, BC	Mgmt. Msg. or means of observing the DUT's grandmaster

References: [1] IEEE Std 1588-2008: sub-clause 9.3.2.2

Resource Requirements: One test station capable of transmitting and receiving arbitrary MAC frames

Modification History: 2013-02-26 Preview release

Discussion: The test will verify the DUT discards Announce messages with alternateMasterFlag TRUE except for the provisions of the master cluster option for the BMC algorithm [1]. To observe whether the DUT has discarded an incoming Announce message we vary the alternateMasterFlag field and observe the DUT's grandmaster.

Test Setup: Refer to Appendix A: DEFAULT TEST SETUP.

Test Procedure:

Part A: Discarded Announce messages, by alternateMasterFlag

A:1. Capture traffic received by TS1 for the duration of this test.

A:2. For the duration of this test, have TS1 send Announce messages with all fields that influence the BMCA, other than the grandmasterIdentity and grandmasterPriority1 fields, identical to the DUT's data sets. For the grandmasterIdentity field use the value 0x102233fffe44556**6**. For the grandmasterPriority1 field use a value one less than the DUT's grandmasterPriority1. Use FALSE for the alternateMasterFlag.

A:3. For the duration of this test, have TS1 also send, intermingled with the above Announce messages, a distinct stream of different Announce messages with all fields that influence the BMCA, other than the grandmasterIdentity and grandmasterPriority1 fields, identical to the corresponding fields in the DUT's data sets. For the grandmasterIdentity field use the value 0x102233fffe44556**7**. For the grandmasterPriority1 field use a value two less than the DUT's grandmasterPriority1. Use TRUE for the alternateMasterFlag..

A:4. After 5 seconds, observe the DUT's grandmaster,
   a. by using the GET management action of the managementId value 2002 hex, or
   b. by other means, if provided

A:5. If the device has more than one port, repeat steps A:1-4 for one other port on the device.

Observable Results:

Part:Step	Status	Description
A:4	FAIL	The DUT's grandmasterIdentity is 0x102233fffe44556**7**.
A:4	FAIL	The DUT's grandmasterIdentity is not 0x102233fffe44556**6**.
A:4	PASS	The DUT discarded Announce messages whose alternateMasterFlag was TRUE, and the DUT accepted Announce messages whose alternateMasterFlag was FALSE.

Possible Problems: None

Test PWR.c.2.6 – Data Set Comparison on a Single Port

Purpose: To verify that the DUT selects the correct grandmaster clock based on the value of the grandmasterPriority1, grandmasterClockQuality, grandmasterPriority2, and grandmasterIdentity fields in Announce messages.

Device Type Prerequisites and Certification Classifier:

Part	Applies To Device Type	Prerequisite Conditions
A, B	OC, BC	Means of observing the DUT's grandmaster
C	OC, BC	Mgmt. Msg.

References: [1] IEEE Std 1588-2008: sub-clause 9.3.4
[2] IEEE Std 1588-2008: sub-clause 7.6.2
[1] IEEE Std 1588-2008: Figure 27
[2] IEEE Std 1588-2008: Table 12

Resource Requirements: One test station capable of transmitting and receiving arbitrary MAC frames

Modification History: 2013-02-20 Preview release

Discussion: This test will verify that the DUT selects the correct grandmaster when receiving Announce messages with various values in the grandmasterPriority1, grandmasterClockQuality, grandmasterPriority2 and grandmasterIdentity fields. Reference [1] describes the data set comparison algorithm that is employed by IEEE 1588 Power Profile devices during selection of a grandmaster clock. When a new or existing clock in a PTP domain transmits an Announce message the device under test must determine whether the announcing clock is better than the current grandmaster, whose information is stored in the DUT's parentDS data set. When two new or existing clocks in a PTP domain transmit Announce messages the device under test must determine which announcing clock is better. After receiving and qualifying an Announce message, the DUT generates a STATE_DECISION_EVENT, which initiates the best master clock algorithm.

The best master clock algorithm compares one clock to another by comparing data sets that represent those clocks. This test ensures that the DUT performs the proper action based on the value of the grandmasterPriority1, clockClass, clockAccuracy, offsetScaledLogVariance, grandmasterPriority2, and grandmasterIdentity fields. The grandmasterClockQuality field is of type ClockQuality. A ClockQuality structure comprises three fields: clockClass, clockAccuracy, and offsetScaledLogVariance. Reference [2] states that lower values take precedence over higher ones.

In reference [3] the first step in comparing data sets describing different grandmasters is to compare the grandmasterPriority1 values. The value of the priority1 field can be anywhere from 0 to 255. If the priority1 values are equal then the second step is to compare the clockClass values. The clockClass field is the first octet of the grandmasterClockQuality field. If the clockClass values are equal then the third step is to compare the clockAccuracy values. The clockAccuracy field is the second octet of the grandmasterClockQuality field in Announce messages. Similarly, the fourth step is to compare the offsetScaledLogVariance values which occupy the third and fourth octets of the grandmasterClockQuality. The fifth step is to compare the grandmasterPriority2 values. The value of the priority2 field can be anywhere from 0 to 255. Finally, if all of the previous values match the sixth and tie-breaking step is to compare the grandmasterIdentity values.

Refer to Appendix C **Error! Reference source not found.**
Test Setup: Refer to Appendix A: DEFAULT TEST SETUP.

Test Procedure:

Part A: Data Set Comparison between self and one foreign master

A:1. Have TS1 send Announce messages with all fields that influence the BMCA, other than the grandmasterIdentity and the **grandmasterPriority1** field, identical to the corresponding fields of the DUT data sets. For the grandmasterIdentity field use the value 0x102233fffe44556**6**. For the **grandmasterPriority1** field use a value *greater than* the DUT's **grandmasterPriority1** value.

A:2. After 5 seconds, observe the DUT's grandmaster,
   a. by using the GET management action of the managementId value 2002 hex, or
   b. by other means, if provided

A:3. Decrease the value of the **grandmasterPriority1** field of the Announce messages emitted to a value *less than* the DUT's **grandmasterPriority1** value.

A:4. After 5 seconds, observe the DUT's grandmaster,
   a. by using the GET management action of the managementId value 2002 hex, or
   b. by other means, if provided

A:5. Repeat steps A:1-4 four times. Each time instead of altering the **grandmasterPriority1** field, alter one of the following fields.
   a. clockClass
   b. clockAccuracy
   c. offsetScaledLogVariance
   d. priority2

A:6. If the device has more than one port, repeat steps A:1-5 for one other port on the device.

Observable Results:

Part:Step	Status	Description
A:2	FAIL	The DUT's grandmasterIdentity is 0x102233fffe44556**6**.
A:4	FAIL	The DUT's grandmasterIdentity is not 0x102233fffe44556**6**.
A:5	PASS	The DUT selected the correct grandmaster.

Part B: Data Set Comparison between two foreign masters

B:1. Capture traffic received by TS1 for the duration of this test.

B:2. For the duration of this test, have TS1 send Announce messages with all fields that influence the BMCA, other than the grandmasterIdentity and **grandmasterPriority1** fields, identical to the DUT's data sets. For the grandmasterIdentity field use the value 0x102233fffe44556**6**. For the **grandmasterPriority1** field use a value *one less* than the DUT's **grandmasterPriority1**.

B:3. For the duration of this test, have TS1 also send a distinct stream of different Announce messages with all fields that influence the BMCA, other than the grandmasterIdentity and **grandmasterPriority1** fields, identical to the corresponding fields in DUT's data sets. For the grandmasterIdentity field use the value 0x102233fffe44556**7**. For the **grandmasterPriority1** field use a value *two less* than the DUT's **grandmasterPriority1**.

B:4. After 5 seconds, observe the DUT's grandmaster,
   a. by using the GET management action of the managementId value 2002 hex, or
   b. by other means, if provided

B:5. Repeat steps B:1-4 four times. Each time instead of altering the **grandmasterPriority1** field, alter one of the following fields.
   a. clockClass
   b. clockAccuracy
   c. offsetScaledLogVariance
   d. priority2

B:6. If the device has more than one port, repeat steps B:1-5 for one other port on the device.

Observable Results:

Part:Step	Status	Description
B:4	FAIL	The DUT's grandmasterIdentity is 0x102233fffe44556**6**.
B:4	FAIL	The DUT's grandmasterIdentity is not 0x102233fffe44556**7**.
B:5	PASS	The DUT selected the correct grandmaster.

Part C: Data Set Comparison between self and one foreign master with SNMP

C:1. Have TS1 send Announce messages with all fields that influence the BMCA, other than the grandmasterIdentity and the **grandmasterPriority1** field, identical to the corresponding fields of the DUT data sets. For the grandmasterIdentity field use the value 0x102233fffe44556**6**. For the **grandmasterPriority1** field use a value *greater than* the DUT's **grandmasterPriority1** value.

C:2. After 5 seconds, observe the DUT's parent **grandmasterPriority1** by using the GET management action of the managementId value 2002 hex.

C:3. Observe the DUT's grandmaster by using the GET management action of the managementId value 2002 hex.

C:4. Decrease the value of the **grandmasterPriority1** field of the Announce messages emitted to a value *less than* the DUT's **grandmasterPriority1** value.

C:5. After 5 seconds, observe the DUT's parent **grandmasterPriority1** by using the GET management action of the managementId value 2002 hex.

C:6. Observe the DUT's grandmaster by using the GET management action of the managementId value 2002 hex.

C:7. Repeat steps C:1-6 four times. Each time instead of altering and observing the **grandmasterPriority1**, alter and observe one of the following fields in the managementId value 2002 hex.

   a. clockClass, ieeeC37238parentDS.GMClkClass
   b. clockAccuracy, ieeeC37238parentDS.GMClkAccuracy
   c. offsetScaledLogVariance, ieeeC37238parentDS.GMOfstScdLVar
   d. priority2, ieeeC37238parentDS.GMPriority2

C:8. If the device has more than one port, repeat steps C:1-7 for one other port on the device.

Observable Results:

Part:Step	Status	Description
C:2	FAIL	The DUT's parent **grandmasterPriority1** is *less than* the DUT's **grandmasterPriority1** value.
C:3	FAIL	The DUT's grandmasterIdentity is 0x102233fffe44556**6**.
C:5	FAIL	The DUT's parent **grandmasterPriority1** is *greater than* the DUT's **grandmasterPriority1** value.
C:6	FAIL	The DUT's grandmasterIdentity is not 0x102233fffe44556**6**.
C:7	PASS	The DUT selected the correct grandmaster.

Possible Problems: None

Test PWR.c.2.7 – Data Set Comparison on Multiple Ports

Purpose: To verify that the DUT selects the correct grandmaster clock based on the value of the grandmasterPriority1, grandmasterClockQuality, grandmasterPriority2, and grandmasterIdentity fields in Announce messages received on multiple ports.

Device Type Prerequisites and Certification Classifier:

Part	Applies To Device Type	Prerequisite Conditions
A	BC	Mgmt. Msg. or means of observing the DUT's grandmaster

References: [1] IEEE Std 1588-2008: sub-clause 9.3.4

Resource Requirements: Two test stations capable of transmitting and receiving arbitrary MAC frames

Modification History: 2013-03-04 Preview release

Discussion: This test will verify that the DUT selects the correct grandmaster when receiving Announce messages on multiple ports with various values in the grandmasterPriority1, grandmasterClockQuality, grandmasterPriority2 and grandmasterIdentity fields. The best of all Announce messages received on multiple ports is determined using the data set comparison algorithm [1].

Test Setup: Refer to Appendix A: DEFAULT TEST SETUP.

Test Procedure:

Part A: Data Set Comparison between four foreign masters

A:1. Capture traffic received by TS1 and TS2 for the duration of this test.

A:2. For the duration of this test, have TS1 send to the *first* port on the DUT Announce messages with all fields that influence the BMCA, other than the grandmasterIdentity and **grandmasterPriority1** fields, identical to the DUT's data sets. For the grandmasterIdentity field use the value 0x102233fffe44556**6**. For the **grandmasterPriority1** field use a value *one less* than the DUT's **grandmasterPriority1**.

A:3. For the duration of this test, have TS1 also send to the *first* port on the DUT, a distinct stream of different Announce messages with all fields that influence the BMCA, other than the grandmasterIdentity and **grandmasterPriority1** fields, identical to the corresponding fields in the DUT's data sets. For the grandmasterIdentity field use the value 0x102233fffe44556**7**. For the **grandmasterPriority1** field use a value *two less* than the DUT's **grandmasterPriority1**.

A:4. For the duration of this test, have TS2 send to a *second* port on the DUT Announce messages with all fields that influence the BMCA, other than the grandmasterIdentity and **grandmasterPriority1** fields, identical to the DUT's data sets. For the grandmasterIdentity field use the value 0x102233fffe44556**8**. For the **grandmasterPriority1** field use a value *three less* than the DUT's **grandmasterPriority1**.

A:5. For the duration of this test, have TS2 also send to the *second* port on the DUT, a distinct stream of different Announce messages with all fields that influence the BMCA, other than the grandmasterIdentity and **grandmasterPriority1** fields, identical to the corresponding fields in the DUT's data sets. For the grandmasterIdentity field use the value 0x102233fffe44556**9**. For the **grandmasterPriority1** field use a value *four less* than the DUT's **grandmasterPriority1**.

A:6. After 5 seconds, observe the DUT's grandmaster,
   a. by using the GET management action of the managementId value 2002 hex, or
   b. by other means, if provided

A:7. Repeat steps A:1-6 four times. Each time instead of altering the **grandmasterPriority1** field, alter one of the following fields.
   a. clockClass
   b. clockAccuracy
   c. offsetScaledLogVariance
   d. priority2

A:8. If the device has more than two ports, repeat steps A:1-7 for a different pair of ports on the device.

Observable Results:

Part:Step	Status	Description
A:6	FAIL	The DUT's grandmasterIdentity is 0x102233fffe44556**6**, 0x102233fffe44556**7** or 0x102233fffe44556**8**.
A:6	FAIL	The DUT's grandmasterIdentity is not 0x102233fffe44556**9**.
A:6	PASS	The DUT selected the correct grandmaster.

Possible Problems: None

Test PWR.c.2.8 – State Decision Algorithm

Purpose: To verify that the DUT properly uses the state decision algorithm to determine the state of each of its ports.

Device Type Prerequisites and Certification Classifier:

Part	Applies To Device Type	Prerequisite Conditions
A-C	BC, OC	Mgmt. Msg.

References:
[1] IEEE Std 1588-2008: sub-clause 9.3.4
[2] IEEE Std 1588-2008: sub-clause 9.3.2.2
[3] IEEE Std 1588-2008: Figure 26
[4] IEEE Std 1588-2008: sub-clause 9.2.5

Resource Requirements: Four test stations capable of transmitting and receiving arbitrary MAC frames

Modification History: 2013-03-05 Preview release

Discussion: This test will verify that the DUT uses the state decision algorithm to determine the BMC event applicable to the state machine of each port. An ordinary or boundary clock uses the data set comparison algorithm [1] to determine the best of all Announce messages received on each of its ports, E_{rbest}. Then it will use the data set comparison algorithm to determine the best of all of those, E_{best}; i.e. the best Announce message received by the clock. Then the clock will use E_{rbest} and E_{best} and its own defaultDS data set, D_0, with the state decision algorithm to determine the BMC event applicable to the state machine [2]. This test will verify that the DUT enters the BMC_PASSIVE and BMC_MASTER states as depicted in the state decision algorithm [3]. To validate the DUT enters these states, the behavior of the device is checked with the behavior of each state specified by [4].

Test Setup: Refer to Appendix A: DEFAULT TEST SETUP.

Test Procedure:
Part A: State Decision Algorithm Output P1

A:1. Capture traffic received by TS1 and TS2 for the duration of this test.
A:2. Make the defaultDS.clockQuality.clockClass between 1 through 127,
   a. by means provided, if provided, or
   b. by write, if SNMP is supported
A:3. Have the TS1 and TS2 send Announce messages to the DUT, with all fields that influence the BMCA, other than the sourcePortIdentity field, identical to the DUT message fields. Set the sourcePortIdentity field of the TS1 to a value one less than the DUT's sourcePortIdentity value. Set the sourcePortIdentity field of the TS2 to a value two less than the DUT's sourcePortIdentity value.
A:4. Observe the messages emitted from the DUT.
A:5. Observe the DUT's grandmaster,
   a. by using the GET management action of the managementId value 2002 hex, or
   b. by means provided, if provided

Observable Results:

Part:Step	Status	Description
A:4	FAIL	The DUT sends any messages other than Pdelay_Req, Pdelay_Resp, Pdelay_Resp_Follow_Up, or signaling messages, or management messages that are a required response to another management message.
A:4	FAIL	The DUT is not in the BMC_PASSIVE state.
A:5	PASS	The DUT's grandmaster is not TS2.

Part B: State Decision Algorithm Output M1

B:1. Capture traffic received by TS1 and TS2 for the duration of this test.

B:2. Make the defaultDS.clockQuality.clockClass between 1 through 127,

a. by means provided, if provided, or
b. by using the SET management action of the managementId value 2000 hex.

B:3. Have the TS1 and TS2 send Announce messages to the DUT, with all fields that influence the BMCA, other than the sourcePortIdentity field, identical to the DUT message fields. Set the sourcePortIdentity field of the TS1 to a value one more than the DUT's sourcePortIdentity value. Set the sourcePortIdentity field of the TS2 to a value two more than the DUT's sourcePortIdentity value.

B:4. Observe the messages emitted from the DUT.

B:5. Observe the DUT's grandmaster,

a. by using the GET management action of the managementId value 2002 hex, or
b. by means provided, if provided

Observable Results:

Part:Step	Status	Description
B:4	FAIL	The DUT is not behaving as a master port.
B:5	FAIL	The DUT is not grandmaster.
B:5	PASS	The DUT is in the BMC_MASTER state.

Part C: State Decision Algorithm Output P2

C:1. Capture traffic received by TS1, TS2, TS3 and TS4 for the duration of this test.

C:2. Make the defaultDS.clockQuality.clockClass greater than 127,

a. by means provided, if provided, or
b. by using the SET management action of the managementId value 2000 hex.

C:3. Have the TS1 and TS2 send Announce messages to one port of the DUT, with all fields that influence the BMCA, other than the sourcePortIdentity field, identical to the DUT message fields. Set the sourcePortIdentity field of the TS1 to a value four less than the DUT's sourcePortIdentity value. Set the sourcePortIdentity field of the TS2 to a value three less than the DUT's sourcePortIdentity value.

C:4. Have the TS3 and TS4 send Announce messages to a different port of the DUT, with all fields that influence the BMCA, other than the sourcePortIdentity field, identical to the DUT message fields. Set the sourcePortIdentity field of the TS3 to a value two less than the DUT's sourcePortIdentity value. Set the sourcePortIdentity field of the TS4 to a value one less than the DUT's sourcePortIdentity value.

C:5. Observe the messages emitted from the DUT.

C:6. Observe the DUT's grandmaster,

a. by using the GET management action of the managementId value 2002 hex, or
b. by means provided, if provided

Observable Results:

Part:Step	Status	Description
C:4	FAIL	The DUT sends any messages other than Pdelay_Req, Pdelay_Resp, Pdelay_Resp_Follow_Up, or signaling messages, or management messages that are a required response to another management message.
C:4	FAIL	The DUT is not in the BMC_PASSIVE state.
C:5	PASS	The DUT's grandmaster is not TS1.

Possible Problems: Unsure whether need to add parts to test other outputs of the algorithm i.e. M2, M3 and S1. Do I need to add a check to see if the data sets of the local clocks are updated?

Test PWR.c.2.9 – Steps Removed

Purpose: To verify that the DUT selects the correct grandmaster clock based on the value of the stepsRemoved field in Announce messages.

Device Type Prerequisites and Certification Classifier:

Part	Applies To Device Type	Prerequisite Conditions
A - D	OC, BC	GMC
E - H	OC	Not GMC, Mgmt. Msg. or means of observing the DUT's grandmaster

References: [1] IEEE Std 1588-2008: sub-clause 9.3.4
[2] IEEE Std 1588-2008: figure 28
[3] IEEE Std 1588-2008: table 12

Resource Requirements: One test station capable of transmitting and receiving arbitrary MAC frames

Modification History: 2012-12-17 Preview release

Discussion: This test will verify that the DUT selects the correct grandmaster when receiving Announce messages with various values in the stepsRemoved field. Reference [1] describes the data set comparison algorithm that is employed by IEEE 1588 Power Profile devices during selection of a grandmaster clock. Reference [2] describes how to choose between two data sets when their grandmaster fields are all the same, including even the grandmasterIdentity. When the grandmaster fields are all the same the grandmaster selection process is more specific than choosing a grandmaster clock. Rather, in this case the decision regards, first, which port *on a communication path* is on the shortest path to the common grandmaster (i.e., E_{rbest}), and, second, which port *on the DUT* has the shortest path to the common grandmaster (i.e., E_{best}).

This test ensures that the DUT performs the proper action based on the value of the stepsRemoved field. The stepsRemoved field is at offset 61 in Announce messages.

Test Setup: Refer to Appendix A: DEFAULT TEST SETUP.

Test Procedure:

Part A: stepsRemoved, TS < DUT - 1

A:1. Power on the DUT acting as grandmaster.

A:2. Have the TS send Announce messages with all fields, other than the stepsRemoved field, that influence the BMCA identical to the DUT message fields. Set the stepsRemoved field to a value *two or more less than* the DUT's stepsRemoved value.

A:3. Observe whether the DUT continues sending Announce messages.

Observable Results:

Part:Step	Status	Description
A:3	FAIL	The DUT stops sending Announce messages.
A:3	PASS	The DUT continues sending Announce messages.

Part B: stepsRemoved, TS > DUT + 1

B:1. Power on the DUT acting as grandmaster.

B:2. Have the TS send Announce messages with all fields, other than the stepsRemoved field, that influence the BMCA identical to the DUT message fields. Set the stepsRemoved field to a value *two or more greater than* the DUT's stepsRemoved value.

B:3. Observe whether the DUT continues sending Announce messages.

Observable Results:

Part:Step	Status	Description
B:3	FAIL	The DUT continues sending Announce messages.
B:3	PASS	The DUT stops sending Announce messages.

Part C: stepsRemoved, TS = DUT - 1

C:1. Power on the DUT acting as grandmaster.

C:2. Have the TS send Announce messages with all fields, other than the stepsRemoved field, that influence the BMCA identical to the DUT message fields. Set the stepsRemoved field to a value *one less than* the DUT's stepsRemoved value.

C:3. Observe whether the DUT continues sending Announce messages.

Observable Results:

Part:Step	Status	Description
C:3	FAIL	The DUT stops sending Announce messages.
C:3	PASS	The DUT continues sending Announce messages.

Part D: stepsRemoved, TS = DUT + 1

D:1. Power on the DUT acting as grandmaster.

D:2. Have the TS send Announce messages with all fields, other than the stepsRemoved field, that influence the BMCA identical to the DUT message fields. Set the stepsRemoved field to a value *one greater than* the DUT's stepsRemoved value.

D:3. Observe whether the DUT continues sending Announce messages.

Observable Results:

Part:Step	Status	Description
D:3	FAIL	The DUT continues sending Announce messages.
D:3	PASS	The DUT stops sending Announce messages.

Part E: stepsRemoved, TS1 < TS2 – 1

E:1. Have the TS1 and TS2 send Announce messages with all fields, other than the stepsRemoved field, that influence the BMCA identical to the DUT's values. Set the stepsRemoved field of TS1 to a value *two or more less than* the TS2's stepsRemoved value.

E:2. Observe the DUT's grandmaster,
   a. by using the GET management action of the managementId value 2002 hex, or
   b. by means provided, if provided

Observable Results:

Part:Step	Status	Description
E:2	FAIL	The DUT's grandmaster is not TS1.
E:2	PASS	The DUT's grandmaster is TS1.

Part F: stepsRemoved, TS1 > TS2 + 1

F:1. Have the TS1 and TS2 send Announce messages with all fields, other than the stepsRemoved field, that influence the BMCA identical to the DUT's values. Set the stepsRemoved field of TS1 to a value *two or more greater than* the TS2's stepsRemoved value.

F:2. Observe the DUT's grandmaster,
   a. by using the GET management action of the managementId value 2002 hex, or
   b. by means provided, if provided

Observable Results:

Part:Step	Status	Description
F:2	FAIL	The DUT's grandmaster is not TS2.
F:2	PASS	The DUT's grandmaster is TS2.

Part G: stepsRemoved, TS1 = TS2 – 1

G:1. Have the TS1 and TS2 send Announce messages with all fields, other than the stepsRemoved field, that influence the BMCA identical to the DUT's values. Set the stepsRemoved field of TS1 to a value *one less than* the TS2's stepsRemoved value.

G:2. Observe the DUT's grandmaster,
   a. by using the GET management action of the managementId value 2002 hex, or
   b. by means provided, if provided

Observable Results:

Part:Step	Status	Description
G:2	FAIL	The DUT's grandmaster is not TS1.
G:2	PASS	The DUT's grandmaster is TS1.

Part H: stepsRemoved, TS1 = TS2 + 1

H:1. Have the TS1 and TS2 send Announce messages with all fields, other than the stepsRemoved field, that influence the BMCA identical to the DUT's values. Set the stepsRemoved field of TS1 to a value *one greater than* the TS2's stepsRemoved value.

H:2. Observe the DUT's grandmaster,

a. by using the GET management action of the managementId value 2002 hex, or
b. by means provided, if provided

Observable Results:

Part:Step	Status	Description
H:2	FAIL	The DUT's grandmaster is not TS2.
H:2	PASS	The DUT's grandmaster is TS2.

Possible Problems: Parts C and D may lead to error-1 indicating that one of the messages was transmitted and received on the same port.

Test PWR.c.2.10 – Source Port Identity

Purpose: To verify that the DUT selects the correct grandmaster clock based on the value of the sourcePortIdentity field in Announce messages.

Device Type Prerequisites and Certification Classifier:

Part	Applies To Device Type	Prerequisite Conditions
A - D	OC, BC	GMC
E - H	OC	Not GMC, Mgmt. Msg. or means of observing the DUT's grandmaster

References: [1] IEEE Std 1588-2008: sub-clause 9.3.4
[2] IEEE Std 1588-2008: figure 28
[3] IEEE Std 1588-2008: table 12

Resource Requirements: One test station capable of transmitting and receiving arbitrary MAC frames

Modification History: 2012-12-17 Preview release

Discussion: This test will verify that the DUT selects the correct grandmaster when receiving Announce messages with various values in the sourcePortIdentity field. Reference [1] describes the data set comparison algorithm that is employed by IEEE 1588 Power Profile devices during selection of a grandmaster clock. Reference [2] describes how to choose between two data sets when their grandmaster fields are all the same, including even the grandmasterIdentity. When the grandmaster fields are all the same the grandmaster selection process is more specific than choosing a grandmaster clock. Rather, in this case the decision regards, first, which port *on a communication path* is on the shortest path to the common grandmaster (i.e., E_{rbest}), and, second, which port *on the DUT* has the shortest path to the common grandmaster (i.e., E_{best}). This test naturally follows the previous test (stepsRemoved). If two incoming Announce messages share the same grandmaster fields and also the same stepsRemoved fields then the final tie-breaking distinction is based on the clockIdentity of the two Announce message sources.

This test ensures that the DUT performs the proper action based on the value of the sourcePortIdentity field. The clockIdentity field is the first eight octets of the sourcePortIdentity at offset 20 in message headers. The portNumber field is the last two octets of the sourcePortIdentity at offset 28 in message headers.

Refer to Appendix C **Error! Reference source not found.**
Test Setup: Refer to Appendix A: DEFAULT TEST SETUP.

Test Procedure:

Part A: sourcePortIdentity.clockIdentity, TS < DUT

A:1. Configure the DUT to act as grandmaster.

A:2. Have the TS send Announce messages with all fields that influence the BMCA, other than the clockIdentity field, identical to the DUT message fields. Set the clockIdentity field to a value *less than* the DUT's clockIdentity value.

A:3. Observe whether the DUT continues sending Announce messages.

Observable Results:

Part:Step	Status	Description
A:3	FAIL	The DUT continues sending Announce messages.
A:3	PASS	The DUT stops sending Announce messages.

Part B: sourcePortIdentity.clockIdentity, TS > DUT

B:1. Power on the DUT acting as grandmaster.

B:2. Have the TS send Announce messages with all fields that influence the BMCA, other than the clockIdentity field, identical to the DUT message fields. Set the clockIdentity field to a value *greater than* the DUT's clockIdentity value.

B:3. Observe whether the DUT continues sending Announce messages.

Observable Results:

Part:Step	Status	Description
B:3	FAIL	The DUT stops sending Announce messages.
B:3	PASS	The DUT continues sending Announce messages.

Part C: sourcePortIdentity.portNumber, TS < DUT

C:1. Power on the DUT acting as grandmaster.

C:2. Have the TS send Announce messages with all fields that influence the BMCA, other than the portNumber field, identical to the DUT message fields. Set the portNumber field to a value *less than* the DUT's portNumber value.

C:3. Observe whether the DUT continues sending Announce messages.

Observable Results:

Part:Step	Status	Description
C:3	FAIL	The DUT continues sending Announce messages.
C:3	PASS	The DUT stops sending Announce messages.

Part D: sourcePortIdentity.portNumber, TS > DUT

D:1. Power on the DUT acting as grandmaster.

D:2. Have the TS send Announce messages with all fields that influence the BMCA, other than the portNumber field, identical to the DUT message fields. Set the portNumber field to a value *greater than* the DUT's portNumber value.

D:3. Observe whether the DUT continues sending Announce messages.

Observable Results:

Part:Step	Status	Description
D:3	FAIL	The DUT stops sending Announce messages.
D:3	PASS	The DUT continues sending Announce messages.

Part E: sourcePortIdentity.clockIdentity, TS1 < TS2 < DUT

E:1. Configure the DUT to act as grandmaster.

E:2. Have the TS1 and TS2 send Announce messages with all fields that influence the BMCA, other than the clockIdentity field, identical to the DUT's values. Set the clockIdentity field of TS2 to a value *less than* the DUT's clockIdentity value. Set the clockIdentity field of TS1 to a value *less than* the TS2's clockIdentity value.

E:3. Observe the DUT's grandmaster,
  a. by using the GET management action of the managementId value 2002 hex, or
  b. by means provided, if provided

Observable Results:

Part:Step	Status	Description
E:2	FAIL	The DUT's grandmaster is not TS1.
E:2	PASS	The DUT's grandmaster is TS1.

Part F: sourcePortIdentity.clockIdentity, TS2 < TS1 < DUT

F:1. Power on the DUT acting as grandmaster.

F:2. Have the TS1 and TS2 send Announce messages with all fields that influence the BMCA, other than the clockIdentity field, identical to the DUT's values. Set the clockIdentity field of TS1 to a value *less than* the DUT's clockIdentity value. Set the clockIdentity field of TS2 to a value *less than* the TS1's clockIdentity value.

F:3. Observe the DUT's grandmaster,
  a. by using the GET management action of the managementId value 2002 hex, or
  b. by means provided, if provided

Observable Results:

Part:Step	Status	Description
F:2	FAIL	The DUT's grandmaster is not TS2.
F:2	PASS	The DUT's grandmaster is TS2.

Part G: sourcePortIdentity.portNumber, TS1 < TS2 < DUT

G:1. Configure the DUT to act as grandmaster.

G:2. Have the TS1 and TS2 send Announce messages with all fields that influence the BMCA, other than the portNumber field, identical to the DUT's values. Set the portNumber field of TS2 to a value *less than* the DUT's portNumber value. Set the portNumber field of TS1 to a value *less than* the TS2's portNumber value.

G:3. Observe the DUT's grandmaster,
  a. by using the GET management action of the managementId value 2002 hex, or
  b. by means provided, if provided

Observable Results:

Part:Step	Status	Description
G:2	FAIL	The DUT's grandmaster is not TS1.
G:2	PASS	The DUT's grandmaster is TS1.

Part H: sourcePortIdentity.portNumber, TS2 < TS1 < DUT

H:1. Power on the DUT acting as grandmaster.

H:2. Have the TS1 and TS2 send Announce messages with all fields that influence the BMCA, other than the portNumber field, identical to the DUT's values. Set the portNumber field of TS1 to a value *less than* the DUT's portNumber value. Set the portNumber field of TS2 to a value *less than* the TS1's portNumber value.

H:3. Observe the DUT's grandmaster,
   a. by using the GET management action of the managementId value 2002 hex, or
   b. by means provided, if provided

Observable Results:

Part:Step	Status	Description
H:2	FAIL	The DUT's grandmaster is not TS2.
H:2	PASS	The DUT's grandmaster is TS2.

Possible Problems: None

GROUP 3: State Configuration Options

Overview:

This group covers requirements defined in the IEEE 1588-2008 Std. sub-clause J.3.3 and J.4.3, "PTP Options". The PTP options specified indicate Clause 17 "State Configuration Options" shall be inactive unless specifically activated by a management procedure. This permitted feature can be used to enhance the performance of the best master clock algorithm.

Notes:

GROUP 4: Management Mechanism

Overview:

This group covers requirements defined in the IEEE 1588-2008 Std. sub-clause 15, "Management". Management messages are used to access specific attributes. The management message mechanism of the IEEE 1588 Std. shall be implemented by the node management.

Notes:

Some of the attributes defined in sub-clause 15 are validated in tests located in other groups of this test plan. A table will be made to locate where each attribute is validated in this test suite.

GROUP 5: Clock Physical Requirements

Overview:

This group covers requirements defined in the IEEE 1588-2008 Std. sub-clause J.3.4 and J.4.4, “Clock physical requirements”. These requirements outline frequency accuracy and frequency adjustment range.

Notes:

None

Test PWR.c.5.1 – Frequency Accuracy

Purpose: To verify grandmaster clocks maintain a frequency deviating no more than 0.01% from the SI second.

Device Type Prerequisites and Certification Classifier:

Part	Applies To Device Type	Prerequisite Conditions
A	BC, OC	GM, One-step
B	BC, OC	GM, Two-step

References: [1] IEEE Std 1588-2008: annex J.3.4.1

Resource Requirements: One test station capable of transmitting and receiving arbitrary MAC frames

Modification History: 2013-07-01 Preview release

Discussion: This test will verify that the grandmaster clock maintains a frequency deviating no more than 0.01% from the SI second [1]. The procedure of this test involves observing the timestamps in Sync and Follow_Up messages.

Test Setup: Refer to Appendix A: DEFAULT TEST SETUP.

Test Procedure:

Part A: Sync Messages

A:1. Capture traffic received by TS1 for the duration of this test part.
A:2. Wait one minute for Sync messages to be received from the DUT.
A:3. Observe the originTimestamp in each Sync message.

Observable Results:

Part:Step	Status	Description
A:2	FAIL	No sync message is received.
A:3	FAIL	The frequency deviates more than 0.01% from the SI second.
A:3	PASS	The frequency does not deviate more than 0.01% from the SI second.

Part B: Sync and Follow_Up Messages

B:1. Capture traffic received by TS1 for the duration of this test part.
B:2. Wait one minute for Sync and Follow_Up messages to be received from the DUT.
B:3. Observe the originTimestamp in each Sync and the preciseOriginTimestamp in each Follow_Up message.

Observable Results:

Part:Step	Status	Description
B:2	FAIL	No sync message is received.
B:2	FAIL	No Follow_Up message is received.
B:3	FAIL	The frequency deviates more than 0.01% from the SI second.
B:3	PASS	The frequency does not deviate more than 0.01% from the SI second.

Possible Problems: More detail to frequency calculation may be needed.

Test PWR.c.5.2 – Frequency Adjustment Range

Purpose: To verify clocks in the slave state correct their frequency to match any master clock.

Device Type Prerequisites and Certification Classifier:

Part	Applies To Device Type	Prerequisite Conditions
A	BC, OC	One-step, Peer-to-peer
B	BC, OC	Two-step, Peer-to-peer
C	BC, OC	End-to-end

References: [1] IEEE Std 1588-2008: annex J.3.4.1

Resource Requirements: One test station capable of transmitting and receiving arbitrary MAC frames

Modification History: 2013-07-01 Preview release

Discussion: This test will verify clocks in the slave state correct their frequency to match any master clock that deviates no more than 0.01% from the SI second [1]. The procedure of this test involves observing the timestamps in Pdelay_Resp and Pdelay_Resp_Follow_Up messages.

Test Setup: Refer to Appendix A: DEFAULT TEST SETUP.

Test Procedure:

Part A: Pdelay_Resp Messages

A:1. Capture traffic received by TS1 for the duration of this test part.
A:2. Wait one minute for Pdelay_Resp messages to be received from the DUT.
A:3. Observe the requestReceiptTimestamp in each Pdelay_Resp message.

Observable Results:

Part:Step	Status	Description
A:2	FAIL	No Pdelay_Resp message is received.
A:3	FAIL	The frequency deviates more than 0.01% from the SI second.
A:3	PASS	The frequency does not deviate more than 0.01% from the SI second.

Part B: Pdelay_Resp and Pdelay_Resp_Follow_Up Messages

B:1. Capture traffic received by TS1 for the duration of this test part.
B:2. Wait one minute for Pdelay_Resp and Pdelay_Resp_Follow_Up messages to be received from the DUT.
B:3. Observe the requestReceiptTimestamp in each Pdelay_Resp and the responseOriginTimestamp in each Pdelay_Resp_Follow_Up message.

Observable Results:

Part:Step	Status	Description
B:2	FAIL	No Pdelay_Resp message is received.
B:2	FAIL	No Pdelay_Resp_Follow_Up message is received.
B:3	FAIL	The frequency deviates more than 0.01% from the SI second.
B:3	PASS	The frequency does not deviate more than 0.01% from the SI second.

Part C: Delay_Req Messages

C:1. Capture traffic received by TS1 for the duration of this test part.
C:2. Wait one minute or for Delay_Req messages to be received from the DUT.
C:3. Observe the originTimestamp in each Delay_Req message.

Observable Results:

Part:Step	Status	Description
C:2	FAIL	No Delay_Req message is received.
C:3	FAIL	The frequency deviates more than 0.01% from the SI second.
C:3	PASS	The frequency does not deviate more than 0.01% from the SI second.

Possible Problems: More detail to frequency calculation may be needed.

GROUP 6: Miscellaneous

Overview:

This group covers miscellaneous requirements defined in the IEEE 1588-2008 Std.

Notes:

None

SECTION PWR.c: 1588 Delay Request-Response Default PTP Profile

Overview:

This selection of tests verifies the various requirements for 1588 Delay Request-Response Default PTP Profile products defined in the IEEE 1588 standard.

Comments and questions regarding the documentation or implementation of these tests are welcome and may be sent to ptplab@iol.unh.edu.

Notes:

Successful completion of all tests contained in this suite does not guarantee that the tested device will successfully operate with other 1588 Default PTP Profile products. However, when combined with a satisfactory level of interoperability testing, these tests provide a reasonable level of confidence that the Device Under Test (DUT) will function properly in many 1588 Default PTP Profile environments.

GROUP 7: Delay Request-Response Mechanism

Overview:

In the IEEE Std 1588-2008 the delay request-response mechanism (i.e. end to end) is the default path delay measurement mechanism, however the peer delay mechanism may also be implemented. This group covers requirements defined in IEEE 1588-2008 sub-clause 11.3, "Delay request-response mechanism". The end-to-end delay mechanism involves Node-A sending Delay_Req messages and Node-B Sync and Follow_Up messages. This messaging process allows Node-A to calculate the meanPathDelay.

Notes:

None

Test PWR.c.7.1 – Mean Path Delay for Delay_Req

Purpose: To verify that the meanPathDelay is computed correctly.

Device Type Prerequisites and Certification Classifier:

Part	Applies To Device Type	Prerequisite Conditions
A	OC, BC	GMC, One-step Clock
B	OC, BC	GMC, Two-step Clock
C	OC, BC	Slave-only, One-step Clock
D	OC, BC	Slave-only, Two-step Clock

References: [1] IEEE Std 1588-2008: sub-clause 11.4.3
[2] IEEE Std 1588-2008: sub-clause 11.5.2.2

Resource Requirements: One test station capable of transmitting and receiving arbitrary MAC frames

Modification History: 2014-01-08 Preview release

Discussion: This test will validate the DUT's meanPathDelay value by comparing it with the known mean path delay. Since the 1588 standard did not establish any accuracy requirements for meanPathDelay, this test just gives a warning if the DUT's meanPathDelay is not within ±50% of the actual mean path delay. The known mean path delay is derived by measuring the cable length. The speed of light in Ethernet cable is 66% of c, or 5.0 ns/m, so every 1.0 m of cable contributes 5.0 ns to the one-way path delay. The test uses the mean of several path delay measurements. In either case, one-step or two-step, the mean path delay value should not vary by much for the requesting and responding nodes when the test setup remains the same. To validate this, the mean and variance of the meanPathDelay is calculated;
refer to Appendix D: Equations

To validate the value of the meanPathDelay in transparent clocks the correctionField of the forwarded Sync messages, for one-step clocks, and Follow_Up messages, for two-step clocks, will be observed [2].

Devices shall measure and calculate the meanPathDelay for each instance of a peer delay measurement. For one-step clocks the calculation for the meanPathDelay is shown below [1].

$$mPD = \frac{(t_2 - t_3) + (rT\ of\ Delay_{Resp} - oT\ of\ Sync) - cF\ of\ Sync - cF\ of\ Delay_{Resp}}{2}$$

For two-step clocks the calculation for the meanPathDelay is shown below.

$$mPD = \frac{(t_2 - t_3) + (rT\ of\ Delay_{Resp} - poT\ of\ Follow\_Up) - cF\ of\ Sync - cF\ of\ Follow\_Up - cF\ of\ Delay_{Resp}}{2}$$

where rT is receiveTimestamp, oT is originTimestamp, cF is correctionField and poT is preciseOriginTimestamp.

Test Setup: Refer to Appendix A: DEFAULT TEST SETUP. The cable between TS1 and DUT.TS1 can be any length, but its length must be known.

Test Procedure:

Part A: One-Step Master DUT MeanPathDelay

A:1. Capture traffic received by TS1 for the duration of this test.
A:2. Wait up to 10 seconds or for a Sync message to be received from the DUT.
A:3. Respond to a Sync message with a Delay_Req message.
A:4. Wait up to 10 seconds or for a Delay_Resp message to be received from the DUT.
A:5. Observe the DUT's meanPathDelay,
  a. by using the GET management action of the managementId value 2004 hex, or
  b. by means provided, if provided.
A:6. Repeat steps A:2-A:4 60 times.
A:7. Calculate the mean and variance of the observed meanPathDelay values.
A:8. Compare this observed meanPathDelay with the known mean path delay derived from the cable.

Observable Results:

Part:Step	Status	Description
A:2	FAIL	No Sync message is received.
A:4	FAIL	No Delay_Resp message is received.
A:5	N/A	meanPathDelay is not observable.
A:7	INFO	The variance of the meanPathDelay value is reported.
A:8	WARN	The average observed meanPathDelay is not within ±50% of the actual mean path delay.

Part B: Two-Step Master DUT MeanPathDelay

B:1. Capture traffic received by TS1 for the duration of this test.
B:2. Wait up to 10 seconds or for a Sync and Follow_Up message to be received from the DUT.
B:3. Respond to a Sync message with a Delay_Req message.
B:4. Wait up to 10 seconds or for a Delay_Resp message to be received from the DUT.
B:5. Observe the DUT's meanPathDelay,
  a. by using the GET management action of the managementId value 2004 hex, or
  b. by means provided, if provided.
B:6. Repeat steps B:2-B:4 60 times.
B:7. Calculate the mean and variance of the observed meanPathDelay values.
B:8. Compare this observed meanPathDelay with the known mean path delay derived from the cable.

Observable Results:

Part:Step	Status	Description
B:2	FAIL	No Sync message is received.
B:2	FAIL	No Follow_Up message is received.
B:4	FAIL	No Delay_Resp message is received.
B:5	N/A	meanPathDelay is not observable.
B:7	INFO	The variance of the meanPathDelay value is reported.
B:8	WARN	The average observed meanPathDelay is not within ±50% of the actual mean path delay.

Part C: One-Step Slave DUT MeanPathDelay

C:1. Capture traffic received by TS1 for the duration of this test.

C:2. For the duration of this test generate and send Announce and Sync messages from TS1 to the DUT.

C:3. Wait up to 10 seconds or for a Delay_Req message to be received from the DUT and note the time of receipt, t4.

C:4. Observe the correctionField of the Delay_Req message.

C:5. Respond to a Delay_Req message with a Delay_Resp message. Set the fields of the Delay_Resp as follows: sequenceId = sequenceId of Delay_Req, requestingPortIdentity = sourcePortIdentity of Delay_Req, domainNumber = domainNumber of Delay_Req, receiveTimestamp = seconds and nanoseconds portion of time t4, correctionField = 0 + correctionField of Delay_Resp - fractional nanosecond portion of t4.

C:6. Observe the DUT's meanPathDelay,
   a. by using the GET management action of the managementId value 2004 hex, or
   b. by means provided, if provided.

C:7. Repeat steps A:2-A:4 60 times.

C:8. Calculate the mean and variance of the observed meanPathDelay values.

C:9. Compare this observed meanPathDelay with the known mean path delay derived from the cable.

Observable Results:

Part:Step	Status	Description
C:3	FAIL	No Delay_Req message is received.
C:4	FAIL	The correctionField of the Delay_Req is not 0.
C:5	N/A	meanPathDelay is not observable.
C:8	INFO	The variance of the meanPathDelay value is reported.
C:9	WARN	The average observed meanPathDelay is not within ±50% of the actual mean path delay.

Part D: Two-Step Slave DUT MeanPathDelay

D:1. Capture traffic received by TS1 for the duration of this test.

D:2. For the duration of this test generate and send Announce, Sync and Follow_Up messages from TS1 to the DUT.

D:3. Wait up to 10 seconds or for a Delay_Req message to be received from the DUT and note the time of receipt, t4.

D:4. Observe the correctionField of the Delay_Req message.

D:5. Respond to a Delay_Req message with a Delay_Resp message. Set the fields of the Delay_Resp as follows: sequenceId = sequenceId of Delay_Req, requestingPortIdentity = sourcePortIdentity of Delay_Req, domainNumber = domainNumber of Delay_Req, receiveTimestamp = seconds and nanoseconds portion of time t4, correctionField = 0 + correctionField of Delay_Resp - fractional nanosecond portion of t4.

D:6. Observe the DUT's meanPathDelay,
   a. by using the GET management action of the managementId value 2004 hex, or
   b. by means provided, if provided.

D:7. Repeat steps A:2-A:4 60 times.

D:8. Calculate the mean and variance of the observed meanPathDelay values.

D:9. Compare this observed meanPathDelay with the known mean path delay derived from the cable.

Observable Results:

Part:Step	Status	Description
D:2	FAIL	No Sync message is received.
D:2	FAIL	No Follow_Up message is received.
D:4	FAIL	No Delay_Resp message is received.
D:5	N/A	meanPathDelay is not observable.
D:7	INFO	The variance of the meanPathDelay value is reported.
D:8	WARN	The average observed meanPathDelay is not within ±50% of the actual mean path delay.

Possible Problems: Means of observing the meanPathDelay for ordinary and boundary clocks may not be available.

SECTION PWR.c: 1588 Peer-to-Peer Default PTP Profile

Overview:

This selection of tests verifies the various requirements for 1588 Peer-to-Peer Default PTP Profile products defined in the IEEE 1588 standard.

Comments and questions regarding the documentation or implementation of these tests are welcome and may be sent to ptplab@iol.unh.edu.

Notes:

Successful completion of all tests contained in this suite does not guarantee that the tested device will successfully operate with other 1588 Default PTP Profile products. However, when combined with a satisfactory level of interoperability testing, these tests provide a reasonable level of confidence that the Device Under Test (DUT) will function properly in many 1588 Default PTP Profile environments.

GROUP 8: Peer Delay Mechanism

Overview:

In the IEEE Std 1588-2008 the delay request-response mechanism (i.e. end to end) is the default path delay measurement mechanism, however the peer delay mechanism may also be implemented. This group covers requirements defined in IEEE 1588-2008 sub-clause 11.4, "Peer delay mechanism". The peer delay mechanism involves Node-A sending Pdelay_Req messages and Node-B responding with Pdelay_Resp and Pdelay_Resp_Follow_Up messages. This messaging process allows Node-A to calculate the meanPathDelay.

Notes:

None

Test PWR.c.8.1 – Independent Ports for Boundary Clocks

Purpose: To verify that link delay measurement is made independently by each port on a boundary clock implementing the peer delay mechanism.

Device Type Prerequisites and Certification Classifier:

Part	Applies To Device Type	Prerequisite Conditions
A	BC	One-step Clock
B	BC	Two-step Clock

References: [1] IEEE Std 1588-2008: sub-clause 11.4.1
[2] IEEE Std 1588-2008: sub-clause 8.2.1.2.3

Resource Requirements: Two test stations capable of transmitting and receiving arbitrary MAC frames.

Modification History: 2013-06-10 Preview release

Discussion: This test will validate the DUT's portDS.MeanPathDelay value on two ports. This test will require two cables of different length with known delay. The ports will be connected to cables that differ in length and therefore meanPathDelay. If the link delay measurement is made independently by each port then the meanPathDelay should differ between ports [1].

The speed of light in Ethernet cables is approximately 2/3 c, or 5.0 ns/m, so every 10 m of cable length difference contributes a difference in meanPathDelay of approximately 50 ns. This test uses a difference of roughly 100 m, for a meanPathDelay difference of roughly 500 ns.

This test does not apply to ordinary clocks since they have only one port [2].

Test Setup: Refer to Appendix A: DEFAULT TEST SETUP. Use a <1 m cable between TS1 and DUT.TS1. Use a 100 m cable between TS2 and DUT.TS2.

Note that another way to achieve the same effect is to use equal-length cables (of any length) and simulate a 100-meter-longer cable between TS2 and DUT.TS2. To do this, for each Pdelay_Resp_Follow_Up message transmitted from TS2 have TS2 subtract 1,000 ns (i.e., one round-trip link delay) from the responseOriginTimestamp before transmitting the message. This makes the turnaround time appear 1,000 ns shorter than it is which makes the round-trip propagation delay appear 1,000 ns longer than it is. In this case also subtract 500 ns from the preciseOrigin-Timestamp field of any Follow_Up messages transmitted from TS2. This makes the one-way propagation delay of Sync messages appear 500 ns longer than it is, matching the illusion created by the Pdelay_Resp_Follow_Up messages.

Test Procedure:

Part A: One-Step DUT Port peerMeanPathDelay

A:1. Capture traffic received by TS1 and TS2 for the duration of this test.
A:2. For each of the two test stations, wait up to 10 seconds for a Pdelay_Req message to be received from the DUT.
A:3. On both links, respond to each Pdelay_Req message with a Pdelay_Resp message.
A:4. After 5 seconds, observe the DUT's meanPathDelay of DUT.TS1 and DUT.TS2, by using the GET management action of the managementId value 2004 hex.
Make both observations within five seconds of each other.

Observable Results:

Part:Step	Status	Description
A:2	FAIL	No Pdelay_Req message is received.
A:4	FAIL	meanPathDelay is not observable.
A:4	FAIL	The observed meanPathDelays are not at least 400 ns apart.
A:4	PASS	The two meanPathDelays observed are at least 400 ns apart.

Part B: Two-Step DUT Port peerMeanPathDelay

B:1. Capture traffic received by TS1 and TS2 for the duration of this test.
B:2. For each of the two test stations, wait up to 10 seconds for a Pdelay_Req message to be received from the DUT.
B:3. On both links, respond to each Pdelay_Req message with a Pdelay_Resp and a Pdelay_Resp_Follow_Up message.
B:4. After 5 seconds, observe the DUT's meanPathDelay of DUT.TS1 and DUT.TS2, by using the GET management action of the managementId value 2004 hex.
Make both observations within five seconds of each other.

Observable Results:

Part:Step	Status	Description
B:2	FAIL	No Pdelay_Req message is received.
B:4	FAIL	meanPathDelay is not observable.
B:4	FAIL	The observed meanPathDelays are not at least 400 ns apart.
B:4	PASS	The two meanPathDelays observed are at least 400 ns apart.

Possible Problems: None

Test PWR.c.8.2 – Independent Ports for Transparent Clocks

Purpose: To verify that link delay measurement is made independently by each port of a transparent clock implementing the peer delay mechanism.

Device Type Prerequisites and Certification Classifier:

Part	Applies To Device Type	Prerequisite Conditions
A	TC	SNMP, One-step Clock
B	TC	SNMP, Two-step Clock
C	TC	One-step Clock
D	TC	Two-step Clock

References: [1] IEEE Std 1588-2008: sub-clause 11.4.1

Resource Requirements: Three test stations capable of transmitting and receiving arbitrary MAC frames.

Modification History: 2013-06-10 Preview release

Discussion: This test will validate the DUT's portDS.peerMeanPathDelay value on two ports. This test will require two cables of different length with known delay. The ports will be connected to cables that differ in length and therefore meanPathDelay. If the link delay measurement is made independently by each port then the meanPathDelays should differ between ports [1].

The speed of light in Ethernet cables is approximately 2/3 c, or 5.0 ns/m, so every 10 m of cable length difference contributes a difference in meanPathDelay of approximately 50 ns. This test uses a difference of roughly 100 m, for a meanPathDelay difference of roughly 500 ns.

A transparent clock adds the value of the incoming meanPathDelay to the correctionField of the outgoing Sync message if it is a one-step clock, or the correctionField of the outgoing Follow_Up message if it is a two-step clock.

Test Setup: Refer to Appendix A: DEFAULT TEST SETUP. Use a <1 m cable between TS1 and DUT.TS1. Use a 100 m cable between TS2 and DUT.TS2.

Note that another way to achieve the same effect is to use equal-length cables (of any length) and simulate a 100-meter-longer cable between TS2 and DUT.TS2. To do this, for each Pdelay_Resp_Follow_Up message transmitted from TS2 have TS2 subtract 1,000 ns (i.e., one round-trip link delay) from the responseOriginTimestamp before transmitting the message. This makes the turnaround time appear 1,000 ns shorter than it is, which makes the round-trip propagation delay appear 1,000 ns longer than it is. In this case also subtract 500 ns from the preciseOrigin-Timestamp field of any Follow_Up messages transmitted from TS2. This makes the one-way propagation delay of Sync messages appear 500 ns longer than it is, matching the illusion created by the Pdelay_Resp_Follow_Up messages.

Test Procedure:

Part A: One-Step DUT Port peerMeanPathDelay

A:1. Capture traffic received by TS1 and TS2 for the duration of this test.

A:2. For each of the two test stations, wait up to 10 seconds for a Pdelay_Req message to be received from the DUT.

A:3. On both links, respond to each Pdelay_Req message with a Pdelay_Resp message.

A:4. After 5 seconds, observe the peerMeanPathDelay of DUT.TS1 and DUT.TS2,
  a. by means provided, if provided, or
  b. by using the GET management action of the managementId value 2004 hex.

Make both observations within five seconds of each other.

Observable Results:

Part:Step	Status	Description
A:2	FAIL	No Pdelay_Req message is received.
A:4	FAIL	meanPathDelay is not observable.
A:4	FAIL	The observed meanPathDelays are not at least 400 ns apart.
A:4	PASS	The two meanPathDelays observed are at least 400 ns apart.

Part B: Two-Step DUT Port peerMeanPathDelay

B:1. Capture traffic received by TS1 and TS2 for the duration of this test.

B:2. For each of the two test stations, wait up to 10 seconds for a Pdelay_Req message to be received from the DUT.

B:3. On both links, respond to each Pdelay_Req message with a Pdelay_Resp and a Pdelay_Resp_Follow_Up message.

B:4. After 5 seconds, observe the peerMeanPathDelay of DUT.TS1 and DUT.TS2,
  a. by means provided, if provided, or
  b. by using the GET management action of the managementId value 2004 hex.

Make both observations within five seconds of each other.

Observable Results:

Part:Step	Status	Description
B:2	FAIL	No Pdelay_Req message is received.
B:4	FAIL	meanPathDelay is not observable.
B:4	FAIL	The observed meanPathDelays are not at least 400 ns apart.
B:4	PASS	The two meanPathDelays observed are at least 400 ns apart.

Part C: One-Step correctionField of Sync Message

C:1. Capture traffic received by TS1, TS2 and TS3 for the duration of this test.
C:2. For TS1, wait up to 10 seconds or for a Pdelay_Req message to be received from the DUT.
C:3. Respond to each Pdelay_Req message with a Pdelay_Resp message along with a Sync message.
C:4. For TS3, wait up to 10 seconds or for a Sync message to be received from the DUT.
C:5. Observe the correctionField of the Sync message.
C:6. For TS2, wait up to 10 seconds or for a Pdelay_Req message to be received from the DUT.
C:7. Respond to each Pdelay_Req message with a Pdelay_Resp message along with a Sync message.
C:8. For TS3, wait up to 10 seconds or for a Sync message to be received from the DUT.
C:9. Observe the correctionField of the Sync message.

Observable Results:

Part:Step	Status	Description
C:2	FAIL	No Pdelay_Req message is received.
C:4	FAIL	No Sync message is received.
C:9	FAIL	The observed correctionFields are not at least 400 ns apart from the correctionField observed in step C:5.
C:9	PASS	The observed correctionFields are at least 400 ns apart from the correctionField observed in step C:5.

Part D: Two-Step correctionField of Follow Up Message

D:1. Capture traffic received by TS1, TS2 and TS3 for the duration of this test.
D:2. For TS1, wait up to 10 seconds or for a Pdelay_Req message to be received from the DUT.
D:3. Respond to each Pdelay_Req message with a Pdelay_Resp message along with Sync and Follow Up messages.
D:4. For TS3, wait up to 10 seconds or for a Follow Up message to be received from the DUT.
D:5. Observe the correctionField of the Follow Up message.
D:6. For TS2, wait up to 10 seconds or for a Pdelay_Req message to be received from the DUT.
D:7. Respond to each Pdelay_Req message with a Pdelay_Resp message along with Sync and Follow Up messages.
D:8. For TS3, wait up to 10 seconds or for a Follow Up message to be received from the DUT.
D:9. Observe the correctionField of the Follow Up message.

Observable Results:

Part:Step	Status	Description
D:2	FAIL	No Pdelay_Req message is received.
D:4	FAIL	No Sync and Follow Up message is received.
D:9	FAIL	The observed correctionFields are not at least 400 ns apart from the correctionField observed in step C:5.
D:9	PASS	The observed correctionFields are at least 400 ns apart from the correctionField observed in step C:5.

Possible Problems: None

Test PWR.c.8.3 – Peer Delay Mechanism

Purpose: To verify that the peer delay mechanism operates independent of port state.

Device Type Prerequisites and Certification Classifier:

Part	Applies To Device Type	Prerequisite Conditions
C	OC, BC	GMC

References: [3] IEEE Std 1588-2008: sub-clause 11.4

Resource Requirements: One test station capable of transmitting and receiving arbitrary MAC frames

Modification History: 2013-06-10 Preview release

Discussion: The peer delay mechanism measures the link delay between two communicating ports with Pdelay_Req, Pdelay_Resp and possibly Pdelay_Resp_Follow_Up messages. This test will verify that in ordinary and boundary clocks the peer delay mechanism operates independently of whether the port is in the master or slave state. This will be tested by observing the peer delay messages emitted from the DUT while it is master (lower priority1) and then while it is slave (higher priority1).

Test Setup: Refer to Appendix A: DEFAULT TEST SETUP.

Test Procedure:

Part A: Does the DUT use the peer delay mechanism regardless of port state?

A:1. Capture traffic received by TS1 for the duration of this test.
A:2. Set priority1 on the TS or on the DUT such that the DUT's priority1 is **greater than** TS1's priority1.
A:3. Send Announce messages from TS1, and confirm that TS1 enters the slave state.
A:4. Send Pdelay_Req messages from TS1.
A:5. Wait up to 10 seconds for messages to be received from the DUT.
A:6. Set priority1 on the TS or on the DUT such that the DUT's priority1 is **less than** TS1's priority1.
A:7. Send Announce messages from TS1, and confirm that DUT enters the master state.
A:8. Send Pdelay_Req messages from TS1.
A:9. Wait up to 10 seconds for messages to be received from the DUT.

Observable Results:

Part:Step	Status	Description
A:5	FAIL	Pdelay_Req and Pdelay_Resp messages are not received.
A:9	FAIL	Pdelay_Req and Pdelay_Resp messages are not received.
A:9	PASS	The peer delay mechanism works in both master and slave state.

Possible Problems: None

Test PWR.c.8.4 – Peer Delay Turnaround Timestamps, One-Step Clock

Purpose: To verify that the correctionField of the Pdelay_Resp message is reasonable in one-step clocks.

Device Type Prerequisites and Certification Classifier:

Part	Applies To Device Type	Prerequisite Conditions
A	All	One-step Clock

References: [1] IEEE Std 1588-2008: sub-clause 11.4.3

Resource Requirements: One test station capable of transmitting and receiving arbitrary MAC frames

Modification History: 2013-06-10 Preview release

Discussion: This test will validate the correctionField of the DUT's Pdelay_Resp messages by comparing it to a known upper bound. For informational purposes the test also calculates the mean and variance of the correctionField values; refer to Appendix D: Equations

For one-step clocks, the second step of the peer delay mechanism is for the delay responder, Node-B, to prepare and send a Pdelay_Resp message according to [1]. The correctionField must be copied from the correctionField of the Pdelay_Req message and then increased by the turnaround time. If there is no asymmetry correction then the correctionField of a Pdelay_Req message shall be 0 [1]. In this test the correctionField in Pdelay_Req messages sent by the test station will be 0, so the correctionField observed in the DUT's Pdelay_Resp messages will be the DUT's indication of its Pdelay turnaround time. This turnaround time must not be greater than the time between the test station's sending of the Pdelay_Req and the test station's receiving of the corresponding Pdelay_Resp, commonly designated as $t_4 - t_1$.

Test Setup: Refer to Appendix A: DEFAULT TEST SETUP.

Test Procedure:

Part A: correctionField

A:1. Capture traffic received by TS1 for the duration of this test.
A:2. Send a Pdelay_Req message every second from TS1.
A:3. Wait up to 10 seconds for Pdelay_Resp messages to be received from the DUT.
A:4. For one minute record the test station's send-to-receive time difference $t_4 - t_1$ and the correctionField of the corresponding Pdelay_Resp message received from DUT.TS1.
A:5. Calculate the mean and the variance of the correctionField values.

Observable Results:

Part:Step	Status	Description
A:3	FAIL	Fewer than 55 Pdelay_Resp messages are received.
A:4	FAIL	The correctionField value in each of the Pdelay_Resp messages is not greater than the correctionField value from its corresponding Pdelay_Req message, i.e., 0.
A:4	FAIL	The correctionField value in any Pdelay_Resp message is greater than $t_4 - t_1$ where t_1 is the departure time of the Pdelay_Req from TS1 and t_4 is the arrival time of the DUT's Pdelay_Resp at TS1.
A:5	INFO	The mean of the correctionField values is reported.
A:5	INFO	The variance of the correctionField values is reported.
A:5	PASS	The correctionFields of the Pdelay_Resp messages are all greater than 0 and less than $t_4 - t_1$.

Possible Problems: The values of the correctionField may vary if asymmetry corrections are required.

Test PWR.c.8.5 – Peer Delay Turnaround Timestamps, Two-Step Clock

Purpose: To verify that the timestamp fields and correctionField values in Pdelay_Resp and Pdelay_Resp_Follow_Up messages are reasonable in two-step clocks.

Device Type Prerequisites and Certification Classifier:

Part	Applies To Device Type	Prerequisite Conditions
A	All	Two-step Clock

References: [1] IEEE Std 1588-2008: sub-clause 11.4.3

Resource Requirements: One test station capable of transmitting and receiving arbitrary MAC frames

Modification History: 2013-02-19 Preview release

Discussion: This test will validate the fields in the DUT's Pdelay_Resp and Pdelay_Resp_Follow_Up messages that are used to indicate the DUT's peer delay turnaround time, typically represented as $t_3 - t_2$. The fields being checked are correctionField and requestReceiptTimestamp in Pdelay_Resp messages and correctionField and response-OriginTimestamp in Pdelay_Resp_Follow_Up messages. The turnaround time is validated by comparing it to a known upper bound. For informational purposes the test also calculates the mean and variance of the turnaround time; refer to Appendix D: Equations
.
For two-step clocks, the second step of the peer delay mechanism is for the delay responder, Node-B, to prepare and send Pdelay_Resp and Pdelay_Resp_Follow_Up messages according to [1]. The requestReceiptTimestamp field should be set to the seconds and nanoseconds portion of t_2.

Two-step devices *should* use 11.4.3 option (c) 8 of IEEE Std 1588-2008 to populate the timestamp and correction fields of Pdelay_Resp and Pdelay_Resp_Follow_Up messages [2]. After the test station has received both response messages from the DUT, the test station can deduce the DUT's Pdelay turnaround time $t_3 - t_2$ from the following.

$$t_3 = responseOriginTimestamp + cF \ (both\ in\ Pdelay\_Resp\_Follow\_Up)$$
$$t_2 = requestReceiptTimestamp - cF \ (both\ in\ Pdelay\_Resp)$$

The test station will not add any asymmetry correction to its Pdelay_Req correctionField, so this will not need to be considered in the Pdelay_Resp_Follow_Up correctionField (shown as 0 in the table above). The DUT's Pdelay turnaround time, $t_3 - t_2$, must not be greater than the time between the test station's sending of the Pdelay_Req and the test station's receiving of the corresponding Pdelay_Resp, commonly designated as $t_4 - t_1$.

Test Setup: Refer to Appendix A: DEFAULT TEST SETUP.

Test Procedure:
Part A: Turnaround Time
- A:1. Capture traffic received by TS1 for the duration of this test.
- A:2. Send a Pdelay_Req message every second from TS1. Alternate correctionField values between 0 and 0x0000 4000 0000 0000 (approximately 1 s).
- A:3. Wait up to 10 seconds for Pdelay_Resp messages to be received from the DUT.
- A:4. For one minute record the test station's send-to-receive time difference $t_4 - t_1$ and the turnaround time of the corresponding pair of Pdelay_Resp and Pdelay_Resp_Follow_Up messages received from DUT.TS1.
- A:5. Calculate the mean and the variance of the correctionField values.

Observable Results:

Part:Step	Status	Description
A:3	FAIL	Fewer than 55 Pdelay_Resp/Pdelay_Resp_Follow_Up message pairs are received.
A:3	WARN	The correctionFields of alternate Pdelay_Resp message do not oscillate in value as the correctionFields of the corresponding Pdelay_Req messages do.
A:4	FAIL	The turnaround time for each of the Pdelay_Resp and Pdelay_Resp_Follow_Up message pairs is not greater than 0.
A:4	FAIL	The turnaround time for any Pdelay_Resp/Pdelay_Resp_Follow_Up message pair is greater than $t_4 - t_1$ where t_1 is the departure time of the Pdelay_Req from TS1 and t_4 is the arrival time of the DUT's Pdelay_Resp at TS1.
A:5	INFO	The mean of the turnaround times is reported.
A:5	INFO	The variance of the turnaround times is reported.
A:5	PASS	The turnaround times of the DUT's Pdelay_Resp/Pdelay_Resp_Follow_Up messages are all greater than 0 and less than $t_4 - t_1$.

Possible Problems: The values of the correctionField may vary if asymmetry corrections are required.

Test PWR.c.8.6 – Pdelay_Req Message Field Values

Purpose: To validate the DUT's Pdelay_Req message field values.

Device Type Prerequisites and Certification Classifier:

Part	Applies To Device Type	Prerequisite Conditions
A	OC, BC	None
B	TC	Syntonized
C	TC	Non-Syntonized

References: [1] IEEE Std 1588-2008: sub-clause 11.4.3

Resource Requirements: One test station capable of transmitting and receiving arbitrary MAC frames

Modification History: 2012-02-19 Preview release

Discussion: This test will verify that Pdelay_Req messages are prepared and sent correctly by observing the domainNumber, correctionField and originTimestamp of Pdelay_Req messages emitted from the DUT. The first step of the peer delay mechanism is for the delay requester, Node-A, to prepare and send Pdelay_Req messages [1]. The correctionField shall be set to zero. The originTimestamp shall be set to zero or an estimate no worse than ± 1 s of the egress timestamp, t_1, of the Pdelay_Req message.

The domainNumber field default initialization value was tested in Test Pwr.c.7 – defaultDS.domainNumber. This test generalizes to other domainNumber values. If Node-A is an ordinary or boundary clock, the domainNumber field shall be set to the domain of Node-A. If Node-A is a syntonized peer-to-peer transparent clock, the domainNumber field shall be set to the domain being measured. The domain being measured is either the primary syntonization domain or one of the alternate domains if syntonization to multiple domains is implemented. If Node-A is not a syntonized peer-to-peer transparent clock, the domainNumber field shall be set to zero.

Refer to Appendix C: **Error! Reference source not found.**
Test Setup: Refer to Appendix A: DEFAULT TEST SETUP.

Test Procedure:
Part A: BC, OC
A:1. Capture traffic received by TS1 for the duration of this test.
A:2. Wait up to 10 seconds for 3 Pdelay_Req messages to be received from the DUT.

Observable Results:

Part:Step	Status	Description
A:2	FAIL	Three Pdelay_Req messages are not received.
A:2	FAIL	The domainNumber is not the domain of the DUT node.
A:2	FAIL	The correctionField is not zero.
A:2	FAIL	The originTimestamp is not zero.
A:2	PASS	All fields of the Pdelay_Req messages are correct.

Part B: Syntonized TC

B:1. Capture traffic received by TS1 for the duration of this test.
B:2. Wait up to 10 seconds for 3 Pdelay_Req messages to be received from the DUT.

Observable Results:

Part:Step	Status	Description
B:2	FAIL	Three Pdelay_Req messages are not received.
B:2	FAIL	The domainNumber is not the domain being measured.
B:2	FAIL	The correctionField is not zero.
B:2	FAIL	The originTimestamp is not zero.
B:2	PASS	All fields of the Pdelay_Req messages are correct.

Part C: Non-Syntonized TC

C:1. Capture traffic received by TS1 for the duration of this test.
C:2. Wait up to 10 seconds for 3 Pdelay_Req messages to be received from the DUT.

Observable Results:

Part:Step	Status	Description
C:2	FAIL	Three Pdelay_Req messages are not received.
C:2	FAIL	The domainNumber is not zero
C:2	FAIL	The correctionField is not zero.
C:2	FAIL	The originTimestamp is not zero.
C:2	PASS	All fields of the Pdelay_Req messages are correct.

Possible Problems: The values of the correctionField may vary if asymmetry corrections are required.

Test PWR.c.8.7 – Pdelay_Resp Message Field Values, One-Step Clock

Purpose: To validate Pdelay_Resp message field values in one-step clocks.

Device Type Prerequisites and Certification Classifier:

Part	Applies To Device Type	Prerequisite Conditions
A	All	One-step Clock

References: [1] IEEE Std 1588-2008: sub-clause 11.4.3

Resource Requirements: One test station capable of transmitting and receiving arbitrary MAC frames

Modification History: 2012-02-19 Preview release

Discussion: This test will verify that Pdelay_Resp messages are prepared and sent correctly by observing the domainNumber, correctionField, sequenceId, requestReceiptTimestamp and requestingPortIdentity of Pdelay_Resp messages emitted from the DUT. For one-step clocks, the second step of the peer delay mechanism is for the delay responder, Node-B, to prepare and send a Pdelay_Resp message according to [1]. Four fields of the Pdelay_Resp message are copied from corresponding fields in the received Pdelay_Req message, as indicated in Table 2: Pdelay_Resp Message Fields. The correctionField should be first copied from the correctionField of the Pdelay_Req message and then increased by the turnaround time. The requestReceiptTimestamp field of the Pdelay_Resp message shall be set to 0.
Refer to Appendix C **Error! Reference source not found.**
Test Setup: Refer to Appendix A: DEFAULT TEST SETUP.

Test Procedure:
Part A: Pdelay_Resp Field Values

- A:1. Capture traffic received by TS1 for the duration of this test.
- A:2. Send a Pdelay_Req message every second from TS1. Alternate correctionField values between 0 and 0x0000 4000 0000 0000 (approximately 1 s).
- A:3. Wait up to 10 seconds for Pdelay_Resp messages to be received from the DUT.

Observable Results:

Part:Step	Status	Description
A:3	FAIL	No Pdelay_Resp message is received.
A:3	FAIL	The domainNumber field of the Pdelay_Resp message is not the same as the domainNumber of the Pdelay_Req message.
A:3	FAIL	The correctionField of each Pdelay_Resp message is not greater than the correctionField of the corresponding Pdelay_Req message.
A:3	WARN	The correctionFields of alternate Pdelay_Resp message do not oscillate in value as the correctionFields of the corresponding Pdelay_Req messages do.
A:3	FAIL	The sequenceId field of the Pdelay_Resp message is not the same as the sequenceId field of the immediately preceding Pdelay_Req message.
A:3	FAIL	The requestReceiptTimestamp field of the Pdelay_Resp message is not zero.
A:3	FAIL	The requestingPortIdentity field of the Pdelay_Resp message is not the same as the sourcePortIdentity field of the Pdelay_Req message.
A:3	PASS	All fields of the Pdelay_Resp messages are correct.

Possible Problems: The values of the correctionField may vary if asymmetry corrections are required.

Test PWR.c.8.8 – Peer Delay Message Field Values, Two-Step Clock

Purpose: To validate Pdelay_Resp and Pdelay_Resp_Follow_Up message field values in two-step clocks.

Device Type Prerequisites and Certification Classifier:

Part	Applies To Device Type	Prerequisite Conditions
A	All	Two-step Clock

References: [1] IEEE Std 1588-2008: sub-clause 11.4.3

Resource Requirements: One test station capable of transmitting and receiving arbitrary MAC frames

Modification History: 2012-02-19 Preview release

Discussion: This test will verify that Pdelay_Resp and Pdelay_Resp_Follow_Up messages are prepared and sent correctly by observing fields in such messages emitted from the DUT. For two-step clocks, the second step of the peer delay mechanism is for the delay responder, Node-B, to prepare and send Pdelay_Resp and Pdelay_Resp_-Follow_Up messages according to [1]. The domainNumber, sequenceId, and requestingPortIdentity fields of the Pdelay_Resp messages should be copied from corresponding Pdelay_Req message fields as specified in Table 1 below. The correctionField of the Pdelay_Resp should be set to zero and then reduced by any fractional nanosecond portion of Pdelay_Req arrival time t_2. The requestReceiptTimestamp field should be set to the seconds and nanoseconds portion of t_2.

The domainNumber, correctionField, sequenceId, and requestingPortIdentity fields of the Pdelay_Resp_Follow_Up messages should be copied from corresponding Pdelay_Req message fields as specified in Table 2 below. The responseOriginTimestamp field should be set to the seconds and nanoseconds portion of Pdelay_Resp departure time t_3. Then any fractional nanosecond portion of t_3 should be added to correctionField.

Refer to Appendix C **Error! Reference source not found.**
Test Setup: Refer to Appendix A: DEFAULT TEST SETUP.

Test Procedure:

Part A: Field Values

A:1. Capture traffic received by TS1 for the duration of this test.

A:2. Send a Pdelay_Req message every second from TS1.

A:3. Wait up to 10 seconds for Pdelay_Resp and Pdelay_Resp_Follow_Up messages to be received from the DUT.

A:4. For one minute record the correctionField and requestReceiptTimestamp of each received Pdelay_Resp message and the correctionField and responseOriginTimestamp of each received Pdelay_Resp_Follow_Up message.

Observable Results:

Part:Step	Status	Description
A:2	FAIL	No Pdelay_Req message is sent.
A:3	FAIL	No Pdelay_Resp message is received.
A:3	FAIL	The domainNumber field of the Pdelay_Resp message is not the same as the domainNumber from the Pdelay_Req message.
A:3	FAIL	The correctionField of the Pdelay_Resp message is larger than a nanosecond.
A:3	FAIL	The sequenceId field of the Pdelay_Resp message is not the same as the sequenceId field from the Pdelay_Req message.
A:3	FAIL	The requestReceiptTimestamp field is not correct.
A:3	FAIL	The requestingPortIdentity field of the Pdelay_Resp message is not the same as the sourcePortIdentity field from the Pdelay_Req message.
A:3	FAIL	No Pdelay_Resp_Follow_Up message is received.
A:3	FAIL	The domainNumber field of the Pdelay_Resp_Follow_Up message is not the same as the domainNumber from the Pdelay_Req message.
A:3	FAIL	The correctionField of the Pdelay_Resp_Follow_Up message is greater than a nanosecond.
A:3	FAIL	The sequenceId field of the Pdelay_Resp_Follow_Up message is not the same as the sequenceId field from the Pdelay_Req message.
A:3	FAIL	The responseOriginTimestamp field is not correct.
A:3	FAIL	The requestingPortIdentity field of the Pdelay_Resp_Follow_Up message is not the same as the sourcePortIdentity field from the Pdelay_Req message.
A:3	INFO	The Pdelay_Resp message should be transmitted as soon as possible after the receipt of the associated Pdelay_Req message.
A:3	INFO	The Pdelay_Resp_Follow_Up message should be transmitted as soon as possible after the transmission of the associated Pdelay_Resp message.
A:3	PASS	All fields of the Pdelay_Resp and Pdelay_Resp_Follow_Up messages are correct.

Possible Problems: The values of the correctionField may vary if asymmetry corrections are required.

Test PWR.c.8.9 – Mean Path Delay

Purpose: To verify that the meanPathDelay is computed correctly.

Device Type Prerequisites and Certification Classifier:

Part	Applies To Device Type	Prerequisite Conditions
A	BC, OC	One-step Clock
B	BC, OC	Two-step Clock
C	TC	One-step Clock
D	TC	Two-step Clock

References: [3] IEEE Std 1588-2008: sub-clause 11.4.3
[4] IEEE Std 1588-2008: sub-clause 11.5.2.2

Resource Requirements: One test station capable of transmitting and receiving arbitrary MAC frames

Modification History: 2013-03-06 Preview release

Discussion: This test will validate the DUT's meanPathDelay value by comparing it with the known mean path delay. Since neither the 1588 standard nor the Power Profile establish any accuracy requirements for meanPathDelay, this test just gives a warning if the DUT's meanPathDelay is not within ±50% of the actual mean path delay. The known mean path delay is derived by measuring the cable length. The speed of light in Ethernet cable is 66% of c, or 5.0 ns/m, so every 1.0 m of cable contributes 5.0 ns to the one-way path delay. The test uses the mean of several path delay measurements. In either case, one-step or two-step, the mean path delay value should not vary by much for the requesting and responding nodes when the test setup remains the same. To validate this, the mean and variance of the meanPathDelay is calculated;
refer to Appendix D: Equations

To validate the value of the meanPathDelay in transparent clocks the correctionField of the forwarded Sync messages, for one-step clocks, and Follow_Up messages, for two-step clocks, will be observed [2].

Devices shall measure and calculate the meanPathDelay for each instance of a peer delay measurement. For one-step clocks the calculation for the meanPathDelay is shown below [1].

$$mPD = \frac{(t_4 - t_1) - correctionField\ of\ Pdelay\_Resp}{2}$$

For two-step clocks the calculation for the meanPathDelay is shown below.

$$mPD = \frac{(t_4 - t_1) - (responseOriginTimestamp - requestReceiptTimestamp) - correctionFields}{2}$$

Test Setup: Refer to Appendix A: DEFAULT TEST SETUP. The cable between TS1 and DUT.TS1 can be any length, but its length must be known.

Test Procedure:

Part A: One-Step DUT MeanPathDelay

A:1. Capture traffic received by TS1 for the duration of this test.
A:2. Wait up to 10 seconds or for a Pdelay_Req message to be received from the DUT.
A:3. Respond to a Pdelay_Req message with a Pdelay_Resp message.
A:4. Observe the DUT's meanPathDelay,
   a. by using the GET management action of the managementId value 2004 hex, or
   b. by means provided, if provided.
A:5. Repeat steps A:2-A:4 60 times.
A:6. Calculate the mean and variance of the observed meanPathDelay values.
A:7. Compare this observed meanPathDelay with the known mean path delay derived from the cable.

Observable Results:

Part:Step	Status	Description
A:2	FAIL	No Pdelay_Req message is received.
A:4	N/A	meanPathDelay is not observable.
A:6	INFO	The variance of the meanPathDelay value is reported.
A:7	WARN	The average observed meanPathDelay is not within ±50% of the actual mean path delay.

Part B: Two-Step DUT MeanPathDelay

B:1. Capture traffic received by TS1 for the duration of this test.
B:2. Wait up to 10 seconds or for a Pdelay_Req message to be received from the DUT.
B:3. Respond to a Pdelay_Req message with a Pdelay_Resp and a Pdelay_Resp_Follow_Up message.
B:4. Observe the DUT's meanPathDelay,
   a. by using the GET management action of the managementId value 2004 hex, or
   b. by means provided, if provided.
B:5. Repeat steps B:2-B:4 60 times.
B:6. Calculate the mean and variance of the observed meanPathDelay values.
B:7. Compare this observed meanPathDelay with the known mean path delay derived from the cable.

Observable Results:

Part:Step	Status	Description
B:2	FAIL	No Pdelay_Req message is received.
B:4	N/A	meanPathDelay is not observable.
B:6	INFO	The variance of the meanPathDelay value is reported.
B:7	WARN	The average observed meanPathDelay is not within ±50% of the actual mean path delay.

Part C: One-Step Transparent Clock DUT MeanPathDelay

C:1. Capture traffic received by TS1 for the duration of this test.
C:2. For the duration of this test generate and send Sync messages from TS1 to the DUT with the correctionField empty.
C:3. Wait up to 10 seconds or for a Pdelay_Req message to be received by TS1 from the DUT.
C:4. Respond to a Pdelay_Req message with a Pdelay_Resp message.
C:5. Wait up to 10 seconds or for a Sync message to be received by TS2 from the DUT.
C:6. Observe the correctionField of the Sync message.
C:7. Observe the DUT's meanPathDelay, by using the GET management action of the managementId value 2004 hex.
C:8. Repeat steps C:2-C:7 60 times.
C:9. Calculate the mean and variance of the observed correctionField values.
C:10. Compare this observed correctionField with the known mean path delay derived from the cable.

Observable Results:

Part:Step	Status	Description
C:3	FAIL	No Pdelay_Req message is received by TS1.
C:5	FAIL	No Sync message is received by TS2.
C:6	N/A	correctionField is not observable.
C:9	INFO	The variance of the meanPathDelay (contents of the correctionField) value is reported.
C:10	WARN	The average observed and requested meanPathDelay is not within ±50% of the actual mean path delay.

Part D: Two-Step Transparent Clock DUT MeanPathDelay

D:1. Capture traffic received by TS1 and TS2 for the duration of this test.
D:2. For the duration of this test generate and send Sync and Follow_Up messages from TS1 to the DUT with the correctionField empty.
D:3. Wait up to 10 seconds or for a Pdelay_Req message to be received by TS1 from the DUT.
D:4. Respond to a Pdelay_Req message with a Pdelay_Resp and a Pdelay_Resp_Follow_Up message.
D:5. Wait up to 10 seconds or for a Sync and Follow_Up message to be received by TS2 from the DUT.
D:6. Observe the correctionField of the Follow_Up message.
D:7. Observe the DUT's meanPathDelay, by using the GET management action of the managementId value 2004 hex.
D:8. Repeat steps D:2-D:7 60 times.
D:9. Calculate the mean and variance of the observed correctionField values.
D:10. Compare this observed correctionField with the known mean path delay derived from the cable.

Observable Results:

Part:Step	Status	Description
D:3	FAIL	No Pdelay_Req message is received by TS1.
D:5	FAIL	Neither a Sync nor Follow_Up message is received by TS2.
D:6	N/A	correctionField is not observable.
D:9	INFO	The variance of the meanPathDelay (contents of the correctionField) value is reported.
D:10	WARN	The average observed and requested meanPathDelay is not within ±50% of the actual mean path delay.

Possible Problems: Means of observing the meanPathDelay for ordinary and boundary clocks may not be available.

Test PWR.c.8.10 – Restriction on Peer Delay Mechanism

Purpose: To verify that the proper action is taken after the receipt of zero or multiple Pdelay_Resp messages.

Device Type Prerequisites and Certification Classifier:

Part	Applies To Device Type	Prerequisite Conditions
A	All	None
B	OC, BC	None
C	TC	None

References: [1] IEEE Std 1588-2008: sub-clause 11.4.4

Resource Requirements: Two test stations capable of transmitting and receiving arbitrary MAC frames

Modification History: 2013-06-10 Preview release

Discussion: This test will verify that the proper action is taken after the DUT receives a varying number of Pdelay_Resp messages by observing whether the DUT port retransmits a Pdelay_Req message or enters the FAULTY state. Specific actions should be taken after the delay requester, Node-A, receives 0, 1 or multiple Pdelay_Resp messages for a transmitted Pdelay_Req [1]. The receipt of multiple responses can be detected by observing that the sourcePortIdentity fields of the Pdelay_Resp messages differ. When no Pdelay_Resp message is received, Node-A should periodically retransmit a Pdelay_Req message to check for the appearance of Node-B. The standard does not specify a retransmission rate, so this test produces a result of WARN if no retransmitted Pdelay_Req is received within 10 seconds. When a single Pdelay_Resp message is received, the protocol of 11.4 should be executed. When multiple Pdelay_Resp messages are received, ordinary and boundary clock ports should enter the FAULTY state, and peer-to-peer transparent clocks should enter a fault condition. For this case, the device may periodically retransmit a Pdelay_Req message and the port must discard received Sync and Follow_Up messages.

Refer to Appendix C **Error! Reference source not found.**
Test Setup: Refer to Appendix A: DEFAULT TEST SETUP.

Test Procedure:
Part A: 0 Pdelay_Resp Received
A:1. Capture traffic received by TS1 for the duration of this test.
A:2. Wait up to 10 seconds for 3 Pdelay_Req messages to be received from the DUT.
A:3. Respond to five consecutive Pdelay_Req messages from the DUT with Pdelay_Resp and Pdelay_Resp_-Follow_Up messages, observing whether the DUT continues to send Pdelay_Req messages.
A:4. Stop responding to the DUT's Pdelay_Resp messages.
A:5. Wait up to 10 seconds for a Pdelay_Req message to be received from the DUT.

Observable Results:

Part:Step	Status	Description
A:2	FAIL	No Pdelay_Req message is received.
A:3	FAIL	The DUT stops sending Pdelay_Req messages.
A:5	WARN	The DUT does not transmit another Pdelay_Req within 10 seconds.
A:5	PASS	The DUT continues to transmit Pdelay_Req messages.

Part B: Multiple Pdelay_Resp Received

B:1. Capture traffic received by TS1 for the duration of this test.
B:2. With TS2 as grandmaster, send valid Sync and Follow_Up messages from TS2.
B:3. Wait up to 10 seconds for the DUT to forward the Sync and Follow_Up messages received at DUT.TS2 out DUT.TS1.
B:4. Wait up to 10 seconds for 3 Pdelay_Req messages to be received from the DUT.
B:5. From TS1 and within a single Pdelay_Req interval respond to a Pdelay_Req message with two Pdelay_Resp messages, each with differing sourcePortIdentity fields.
B:6. Using SNMP or a vendor-provided method, observe whether DUT.TS1 enters the FAULTY state.
B:7. Observe whether DUT.TS1 discontinues forwarding Sync and Follow_Up messages received from TS2.

Observable Results:

Part:Step	Status	Description
B:3	FAIL	Sync and Follow_Up messages are not forwarded.
B:4	FAIL	No Pdelay_Req message is received.
B:6	FAIL	The device does not enter the FAULTY state.
B:6	FAIL	No FaultyState notification is received indicating the device has entered FAULTY state.
B:7	FAIL	The device continues forwarding Sync and Follow_Up messages.
B:7	PASS	The device enters the FAULTY state and stops forwarding Sync and Follow_Up messages.

Part C: Multiple Pdelay_Resp Received

C:1. Capture traffic received by TS1 for the duration of this test.
C:2. With TS2 as grandmaster, send valid Sync and Follow_Up messages from TS2.
C:3. Wait up to 10 seconds for the DUT to forward the Sync and Follow_Up messages received at DUT.TS2 out DUT.TS1.
C:4. Wait up to 10 seconds for 3 Pdelay_Req messages to be received from the DUT.
C:5. From TS1 and within a single Pdelay_Req interval respond to a Pdelay_Req message with two Pdelay_Resp messages, each with differing sourcePortIdentity fields.
C:6. Using SNMP or a vendor-provided method, observe whether DUT.TS1 enters the FAULTY state.
C:7. Observe whether DUT.TS1 discontinues forwarding Sync and Follow_Up messages received from TS2.

Observable Results:

Part:Step	Status	Description
C:2	FAIL	Sync and Follow_Up messages are not forwarded.
C:3	FAIL	No Pdelay_Req message is received.
C:3	FAIL	The device does not enter the fault condition.
C:3	FAIL	The device continues forwarding Sync and Follow_Up messages.
C:3	PASS	The device enters the fault condition and stops forwarding Sync and Follow_Up messages.

Possible Problems: None

Appendix A: DEFAULT TEST SETUP

Except where otherwise specified, all tests will require the DUT to have the following default configuration at the beginning of each test case:

VLAN Settings:
- The ports connected to the test stations should not be members of any VLAN
  - If this isn't possible, then only a member of VLAN 1 (and AAAA cannot be 1)
- The ports connected to the test stations should not be members of the untagged set for any VLAN
- Port Settings
  - Enable Ingress Filtering – Disabled or Enabled
  - Acceptable Frame Types – Accept All Frames

Spanning Tree Settings:
- RSTP or MSTP is enabled
- DUT priority should be default (0x8000)
- Hello Time must be 2 seconds (if configurable)
- Port Settings
  - AdminEdge – False
  - AutoEdge – False

FDB Settings:
- Ageing time – 300 seconds (default)

MVRP Settings:
- MVRP may be enabled
- Port Settings
  - Restricted_VLAN_Registration – Normal Registration

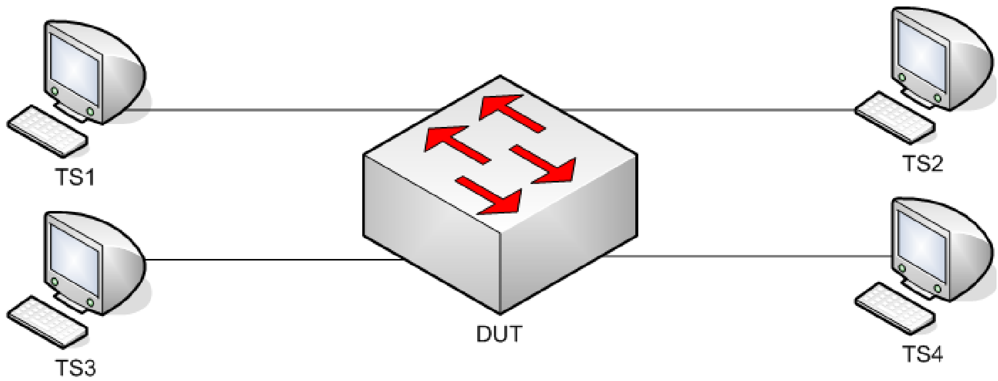
Choice of VLAN component type:
Except where otherwise noted, each test case may be configured and executed using either C-VLAN components, or S-VLAN components. The Destination MAC address in MVRPDUs transmitted to the DUT must be selected accordingly. For AVnu testing purposes, all Bridges must operate with C-VLAN components by default, per AVnu PICS item AVnu_MVRP-4.

VID Translation Tables:
Disabled (i.e. all VLANs map to themselves)

Dynamic Port Parameters: *(The conditions below should be as indicated at the beginning of each test)*
- DUT is root of the spanning tree
- Port Conditions
  - Port Role = Designated
  - Port State = Forwarding
  - No Dynamic VLAN Registration Entries exist
  - No Dynamic MAC Address Registration Entries exist
    - At a minimum, no Dynamic MAC Address Registration Entries may exist for any of the MAC addresses used in test traffic

Default Test Topology

Unless otherwise stated, all tests will use the following topology. Many tests require less than 4 Test Stations. In those cases, the additional test stations may or may not be actually included in the physical topology, at the discretion of the test technician.



Test Traffic:

Some tests specify the transmission of test traffic from a test station. Unless otherwise specified, this test traffic shall be well-formed 64-byte VLAN-tagged broadcast traffic sent at a rate of 10 frames per second. For each test, each test station will be configured for a unique source address amongst the set of test stations.

Appendix B: NOTES ON TEST PROCEDURES

There are scenarios where in test procedures it is desirable to leave certain aspects of the testing procedure as general as possible. In these cases, the steps in the described test procedure may use placeholder values, or may intentionally use non-specific terminology, and the final determination of interpretation or choice of values is left to the discretion of the test technician. The following is an attempt to capture and describe all such instances used throughout the procedures.

Ports on DUT

In general, any Bridge Port on the DUT may be used as an interface with a test station. There is assumed to be no difference in behavior, with respect to the protocols involved in this test suite, between any two Bridge Ports on the DUT. Hence, actual ports used may be chosen for convenience, on a basis such as media type or link speed. Specific Bridge Ports are designated by the Test Station (TS) connected to them via the notation DUT.TS (e.g. Test Station 1 (TS1) is connected to Bridge Port DUT.TS1)

VLAN IDs

Many of the procedures call for registration of certain VIDs, or for traffic to be transmitted tagged with a certain VID. Including a specific VID in the procedures of these cases limits the generality of the test case, and may preclude test execution on certain devices which treat the VID with a special significance. Given this, placeholders are used for VIDs when appropriate.

In general, any VID may be used in these cases, with the following stipulations: Each placeholder must use a different VID (e.g. VLAN AAAA and VLAN BBBB may not both be VID 204) . A given placeholder must remain consistent within a given test part (e.g. VLAN AAAA must refer to the same VID throughout a test part).

Where there exists a constraint on the VID chosen for a given placeholder, the test case will specify the constraint.

Use of "various"

To maintain generality, some steps will specify that "various other values" (or the like) should be used in place of a given parameter. Ideally, all possible values would be tested in this case. However, limits on available time may constrain the ability of the test technician to attempt this. Given this, a subset of the set of applicable values must generally be used.

When deciding how many values should be used, it should be noted that the more values that are tested, the greater the confidence of the results obtained (although there is a diminishing return on this).

When deciding which specific values to use, it is generally recommended to choose them at pseudo-randomly yet deterministically. However, if there exists subsets of the applicable values with special significance, values from each subset should be attempted.

Inter-test health check

Where possible and time permitting, the health and operation of the DUT can be evaluated between each test, and potentially even between each test part. The purpose of such 'health checking' is simply to establish the device as operational. This may take the form of monitoring for periodic transmissions (Pdelay_Req messages, BPDU Messages, etc) or actively probing the device depending on its capabilities (e.g.: ICMP Echo Requests). These optional inter-test health checking aids in detecting 'silent failures' caused by previous tests that may appear to pass but otherwise result in following test cases to fail.

Appendix C: Tables

Appendix D: Equations

In a sequence of messages from the DUT, the mean, variance and standard deviation of the intervals, I, are calculated as follows.

$$mean = \frac{1}{n}(I_1 + I_2 + I_3 + \cdots + I_n)$$

$$variance = \frac{1}{n}((I_1 - mean)^2 + (I_2 - mean)^2 + (I_3 - mean)^2 + \cdots + (I_n - mean)^2)$$

$$standard\ deviation = s = \sqrt{variance}$$

With the sample mean ($mean$) and the sample std.dev (s) computed, the next step is to compute, with 90% confidence, the range the true mean is within as follows.

$$mean - 1.645\left(\frac{s}{\sqrt{n}}\right) < \mu < mean + 1.645\left(\frac{s}{\sqrt{n}}\right)$$

In a sequence of Pdelay_Resp messages from the DUT the mean and variance of the correctionField *cF* are calculated as follows.

$$mean = \frac{1}{n}(cF_1 + cF_2 + cF_3 + \cdots + cF_n)$$

$$variance = \frac{1}{n}((cF_1 - mean)^2 + (cF_2 - mean)^2 + (cF_3 - mean)^2 + \cdots + (cF_n - mean)^2)$$

In a sequence of Pdelay_Resp and Pdelay_Resp_Follow_Up messages from the DUT, the mean and variance of the turnaround time *tT* are calculated as follows.

$$mean = \frac{1}{n}(tT_1 + tT_2 + tT_3 + \cdots + tT_n)$$

$$variance = \frac{1}{n}((tT_1 - mean)^2 + (tT_2 - mean)^2 + (tT_3 - mean)^2 + \cdots + (tT_n - mean)^2)$$

The mean and variance are calculated as follows.

$$mean = \frac{1}{n}(mPD_1 + mPD_2 + mPD_3 + \cdots + mPD_n)$$

$$variance = \frac{1}{n}((mPD_1 - mean)^2 + (mPD_2 - mean)^2 + (mPD_3 - mean)^2 + \cdots + (mPD_n - mean)^2)$$

Where *mPD* is the meanPathDelay computed for successive peer delay measurements.