

**SCTE** | **STANDARDS**

---

**Data Standards Subcommittee**

---

**AMERICAN NATIONAL STANDARD**

**ANSI/SCTE 137-1 2017 (R2021)**

**Modular Headend Architecture Part 1: DOCSIS Timing  
Interface**

## NOTICE

The Society of Cable Telecommunications Engineers (SCTE) Standards and Operational Practices (hereafter called “documents”) are intended to serve the public interest by providing specifications, test methods and procedures that promote uniformity of product, interoperability, interchangeability, best practices, and the long term reliability of broadband communications facilities. These documents shall not in any way preclude any member or non-member of SCTE from manufacturing or selling products not conforming to such documents, nor shall the existence of such standards preclude their voluntary use by those other than SCTE members.

SCTE assumes no obligations or liability whatsoever to any party who may adopt the documents. Such adopting party assumes all risks associated with adoption of these documents and accepts full responsibility for any damage and/or claims arising from the adoption of such documents.

NOTE: The user’s attention is called to the possibility that compliance with this document may require the use of an invention covered by patent rights. By publication of this document, no position is taken with respect to the validity of any such claim(s) or of any patent rights in connection therewith. If a patent holder has filed a statement of willingness to grant a license under these rights on reasonable and nondiscriminatory terms and conditions to applicants desiring to obtain such a license, then details may be obtained from the standards developer. SCTE shall not be responsible for identifying patents for which a license may be required or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Patent holders who believe that they hold patents which are essential to the implementation of this document have been requested to provide information about those patents and any related licensing terms and conditions. Any such declarations made before or after publication of this document are available on the SCTE web site at <https://scte.org>.

All Rights Reserved  
©2021 Society of Cable Telecommunications Engineers, Inc.  
140 Philips Road  
Exton, PA 19341

DOCSIS® and M-CMTS™ are trademarks of Cable Television Laboratories, Inc. (CableLabs) and used in this document with permission.

## Document Types and Tags

Document Type: Specification

Document Tags:

- |   |                                    |  |
|---|------------------------------------|--|
| <input type="checkbox"/> Test or Measurement          | <input type="checkbox"/> Checklist | <input type="checkbox"/> Facility          |
| X Architecture or Framework                           | <input type="checkbox"/> Metric    | X Access Network                           |
| <input type="checkbox"/> Procedure, Process or Method | <input type="checkbox"/> Cloud     | <input type="checkbox"/> Customer Premises |

## Document Release History

Release	Date
SCTE 137-1 2007	9/18/2007
SCTE 137-1 2010	7/26/2010
SCTE 137-1 2017	6/5/2017

Note: This document is a reaffirmation of SCTE 137-1 2017. No substantive changes have been made to this document. Information components may have been updated such as the title page, NOTICE text, headers, and footers.

# Contents

<b>1</b>	<b>SCOPE</b>	<b>8</b>
1.1	INTRODUCTION AND OVERVIEW	8
1.1.1	<i>System Requirements</i>	8
1.1.2	<i>T-Services Consideration</i>	8
1.1.3	<i>Modular Implementation Requirements</i>	9
1.1.4	<i>Architecture</i>	10
1.1.5	<i>Synchronization Needed for T-Services Deployment</i>	11
1.2	REQUIREMENTS	11
<b>2</b>	<b>REFERENCES</b>	<b>12</b>
2.1	NORMATIVE REFERENCES	12
2.2	INFORMATIVE REFERENCES	12
2.3	REFERENCE ACQUISITION	12
<b>3</b>	<b>TERMS AND DEFINITIONS</b>	<b>13</b>
<b>4</b>	<b>ABBREVIATIONS</b>	<b>14</b>
<b>5</b>	<b>PHYSICAL LAYER REQUIREMENTS</b>	<b>16</b>
5.1	INTRODUCTION	16
5.2	PHYSICAL CONNECTOR DESCRIPTION	16
5.3	CABLE REQUIREMENTS	17
5.4	ELECTRICAL DESCRIPTION	17
5.4.1	<i>Impedance</i>	17
5.4.2	<i>Isolation</i>	17
5.4.3	<i>EMI Considerations</i>	17
5.4.4	<i>Signal Strength (voltage)</i>	17
5.4.5	<i>Common-mode rejection</i>	17
5.4.6	<i>Signal Description</i>	17
<b>6</b>	<b>DOCSIS TIMING PROTOCOL</b>	<b>18</b>
6.1	DTI TIMING ENTITIES	18
6.2	DTI TIMING STRUCTURE	19
6.3	TRACEABILITY OF DOCSIS TIMESTAMP	20
6.3.1	<i>GPS Frequency Mode</i>	21
6.4	DTI FRAME STRUCTURE REQUIREMENTS	21
6.4.1	<i>Conventions for this Specification</i>	22
6.4.2	<i>SERVER TO CLIENT</i>	22
6.4.3	<i>CLIENT TO SERVER</i>	28
6.5	DTI SERVER-CLIENT PROTOCOL INTERACTION	31
<b>7</b>	<b>DTI CLIENT AND SERVER OPERATION</b>	<b>33</b>
7.1	DTI SERVER MODES	33
7.1.1	<i>Free Running Master Mode</i>	33
7.1.2	<i>External Reference Modes</i>	33
7.1.3	<i>DTI Server Functional Requirements</i>	39
7.1.4	<i>DTI Server Test Signal Mode</i>	40
7.2	DTI CLIENT OPERATION	40
7.2.1	<i>DTI Normal operating conditions</i>	41
7.2.2	<i>DTI Client Operational Modes</i>	41
7.2.3	<i>DTI Client Mode Transition Diagram</i>	42
7.2.4	<i>Functional Requirements</i>	44
7.2.5	<i>DTI Client Port</i>	44

7.2.6	<i>DTI Client Test Port</i> .....	44
7.2.7	<i>DTI Client Test Port Clock</i> .....	44
7.2.8	<i>Alternative EQAM DTI Testing</i> .....	47
7.2.9	<i>DTI Status LEDs</i> .....	47
7.3	<b>DTI DISTRIBUTION FALLBACK STRATEGIES</b> .....	48
<b>ANNEX A RANGING WANDER QUALIFICATION FILTER</b> .....		<b>50</b>
A.1	<b>CHIP TIMING JITTER FOR SYNCHRONOUS OPERATION</b> .....	50
<b>ANNEX B DTI-MIB (NORMATIVE)</b> .....		<b>52</b>
<b>ANNEX C DTI-S CRC-16</b> .....		<b>73</b>
<b>APPENDIX I DTI SERVER FUNCTIONAL DESCRIPTION</b> .....		<b>74</b>
I.1	<b>SERVER DTI SIGNAL PROCESSING</b> .....	74
<b>APPENDIX II DTI CLIENT FUNCTIONAL DESCRIPTION</b> .....		<b>76</b>
II.1	<b>DTI CLIENT PHY</b> .....	77
II.2	<b>DTI CLIENT FRAME PROCESSOR</b> .....	78
II.3	<b>DTI CLIENT CLOCK PROCESSOR</b> .....	79
<b>APPENDIX III DTI JITTER BUDGET</b> .....		<b>81</b>
III.1	<b>MODEL DESCRIPTION</b> .....	81
III.2	<b>ANALYSIS</b> .....	82
III.2.1	<i>Transmit Jitter Specification</i> .....	82
III.2.2	<i>Receive Jitter Analysis</i> .....	82
III.2.3	<i>Receive Frame Jitter Analysis</i> .....	83
III.2.4	<i>10.24 MHz Output Jitter</i> .....	83
<b>APPENDIX IV SYMBOL CLOCK SYNCHRONIZATION</b> .....		<b>84</b>
<b>APPENDIX V DTI HIGH SPEED CLOCK CONSIDERATIONS</b> .....		<b>85</b>
<b>APPENDIX VI M-CMTS SYSTEM ISSUES REGARDING USE OF DTI CONVEYED TIMEBASE BY AN M-CMTS ENTITY INCORPORATING A DTI CLIENT</b> .....		<b>87</b>
VI.1	<b>EXAMPLE ISSUES</b> .....	87
VI.2	<b>SOLUTION</b> .....	88

## Figures

FIGURE 1-1 - TIMING ARCHITECTURE.....	10
FIGURE 6-1 - DTI PROTOCOL TIMING STRUCTURE .....	19
FIGURE 6-2 - DTI FRAME STRUCTURE.....	22
FIGURE 6-3 - EXAMPLE OF DTI SERVER-CLIENT PROTOCOL.....	32
FIGURE 7-1 - GPS TRACEABLE MODE MTIE REQUIREMENTS.....	36
FIGURE 7-2 - NETWORK INPUT MTIE REQUIREMENTS .....	38
FIGURE 7-3 - NETWORK TRACEABLE MODE MTIE REQUIREMENTS.....	39
FIGURE 7-4 - DTI CLIENT SYSTEM INTERFACE.....	40
FIGURE 7-5 - CLIENT MODE TRANSITION DIAGRAM.....	43
FIGURE 7-6 - TEST PORT TIMING DIAGRAM .....	46
FIGURE 7-7 - EDGE QAM TEST PORT OPTION .....	47
FIGURE 7-8 - DTI DISTRIBUTION FALLBACK STRATEGIES.....	48
FIGURE A-1 - CURRENT NON-MODULAR CMTS IMPLEMENTATION .....	50
FIGURE A-2 - NEW MODULAR CMTS IMPLEMENTATION.....	50
FIGURE A-3 - RANGING WANDER TEST FILTER.....	51
FIGURE C-1 - CRC INITIALIZATION .....	73
FIGURE C-2 - CRC ALGORITHMS .....	73
FIGURE I-1 - SERVER TOD/GPS/NETWORK SIGNAL PROCESSING REFERENCE BLOCK DIAGRAM.....	74
FIGURE I-2 - DTI SERVER REFERENCE SIGNAL PROCESSING BLOCK DIAGRAM .....	75
FIGURE II-1 - DTI CLIENT BLOCK DIAGRAM.....	76
FIGURE II-2 - EXAMPLE DTI PHY INTERFACE CIRCUIT .....	77
FIGURE II-3 - DTI CLIENT FRAME PROCESSOR.....	78
FIGURE II-4 - DTI CLOCK PROCESSOR .....	79
FIGURE III-1 - DTI JITTER BUDGET REFERENCE MODEL.....	81
FIGURE V-1 - FRACTIONAL CLOCK DITHER PATTERN .....	85
FIGURE V-2 - PHASE MEASUREMENT DITHER PATTERN 191 PS OFFSET.....	86
FIGURE V-3 - PHASE MEASUREMENT DITHER PATTERN 381 PS OFFSET.....	86

## Tables

TABLE 5-1 - PHYSICAL LAYER COMPARISONS.....	16
TABLE 5-2 - DTI RJ45 INTERCONNECT .....	16
TABLE 6-1 - DTI SERVER FRAME STRUCTURE .....	22
TABLE 6-2 - TOD SUBCOMMUTATED MESSAGE FORMAT .....	25
TABLE 6-3 - PATH TRACEABILITY MESSAGE FORMAT .....	27
TABLE 6-4 - PATH POSITION MESSAGE FORMAT .....	28
TABLE 6-5 - DTI CLIENT FRAME STRUCTURE.....	28
TABLE 6-6 - CLIENT DTI VERSION AND PATH MESSAGE FORMAT .....	31
TABLE 7-1 - DTI SERVER HOLDOVER REQUIREMENTS.....	34
TABLE 7-2 - CLIENT OPERATING STATES .....	41
TABLE 7-3 - CLIENT MODE TRANSITIONS.....	42
TABLE 7-4 - DTI CLIENT TEST PORT .....	44
TABLE 7-5 - CLIENT PHASE NOISE.....	45
TABLE 7-6 - DTI STATUS LEDs .....	47
TABLE III-1 - DTI 10.24 MHZ OUTPUT JITTER PERFORMANCE.....	83
TABLE IV-1 - SYMBOL CLOCK SYNCHRONIZATION.....	84

This page intentionally left blank.

# 1 SCOPE

## 1.1 Introduction and Overview

The requirements for timing and synchronization of the DOCSIS system come from the following areas:

- Existing DOCSIS Specification & ATP Requirements
- Remote PHY System Requirements
- Implementation Requirements
- Emerging Services like T-Services and wireless

These requirements place definitions and constraints on the use of the 10.24 MHz DOCSIS master clock and the DOCSIS timestamp, which is delivered in the SYNC message. The DOCSIS standard [RFI2] originally envisioned the M-CMTS Core, EQAMs, and upstream receive functions on one assembly, fed with a common clock. The timestamp counter resided in the M-CMTS Core function.

The M-CMTS™ Remote PHY architecture may result in three components: the M-CMTS CORE, the upstream receiver, and the EQAM being located in a different chassis, and potentially at different physical locations. As a system, the three components comply with the DOCSIS standard [RFI2] and any existing CMTS equipment.

The DOCSIS Timing Protocol (DTI) defined in this document supports the accurate and robust transport of the DTI server 10.24 MHz master clock, 32-bit DOCSIS timestamp, and Time of Day, to the DTI client within the DOCSIS M-CMTS cable network. The DTI protocol is structured to minimize the complexity and cost of the DTI client clocks, and the per port cost of the shared server function while supporting all S-CDMA and TDMA timing requirements.

### 1.1.1 System Requirements

The DTI system requirements refer to the DOCSIS timing requirements as outlined in the DOCSIS standard [RFI2]. These requirements are presented independent of the CMTS architecture.

The sections of the DOCSIS standard [RFI2] that are of interest are:

- 6.2.11.2 Mini-slot Numbering
- 6.2.21.8.2 Chip Timing Jitter for Synchronous Operation
- 6.3.7 CMTS Timestamp Jitter
- 6.3.8 CMTS Clock Generation
- 6.3.9 CMTS Downstream Symbol Clock Jitter for Synchronous Operation
- 6.3.10 CMTS Downstream Symbol Clock Drift for Synchronous Operation
- 9.3 Timing and Synchronization

### 1.1.2 T-Services Consideration

To maintain compatibility with the T-Service synchronization hierarchy, the DTI Server clock operates with the specifications detailed in section 7.1, which integrate both the DOCSIS timing system requirements and the existing legacy synchronization network clock consistent with [G.812] and [T1.101]. This is done to ensure that the CM supporting T-Services can derive its clocking and meet [G.823] or [G.824] jitter and wander requirements for both traffic bearing and synchronization bearing transport clock sources.

Support of T-Services will require that the master clock and the downstream symbol clock be locked and upstream and downstream clocks be coherent.



### 1.1.3 Modular Implementation Requirements

The M-CMTS CORE element:

- Uses the DTI server master clock for creating a timestamp
- Uses the timestamp for MAP generation

The Edge QAM element:

- Uses the DTI server master clock for symbol rate generation
- Uses the timestamp for inserting and/or correcting SYNC messages

The Upstream receive element:

- Uses the timestamp and/or S-CDMA frame and the MAP for determining when to look for the start of a receive burst
- Uses a clock locked to the master clock for reception of symbols in S-CDMA mode

1.1.4 Architecture

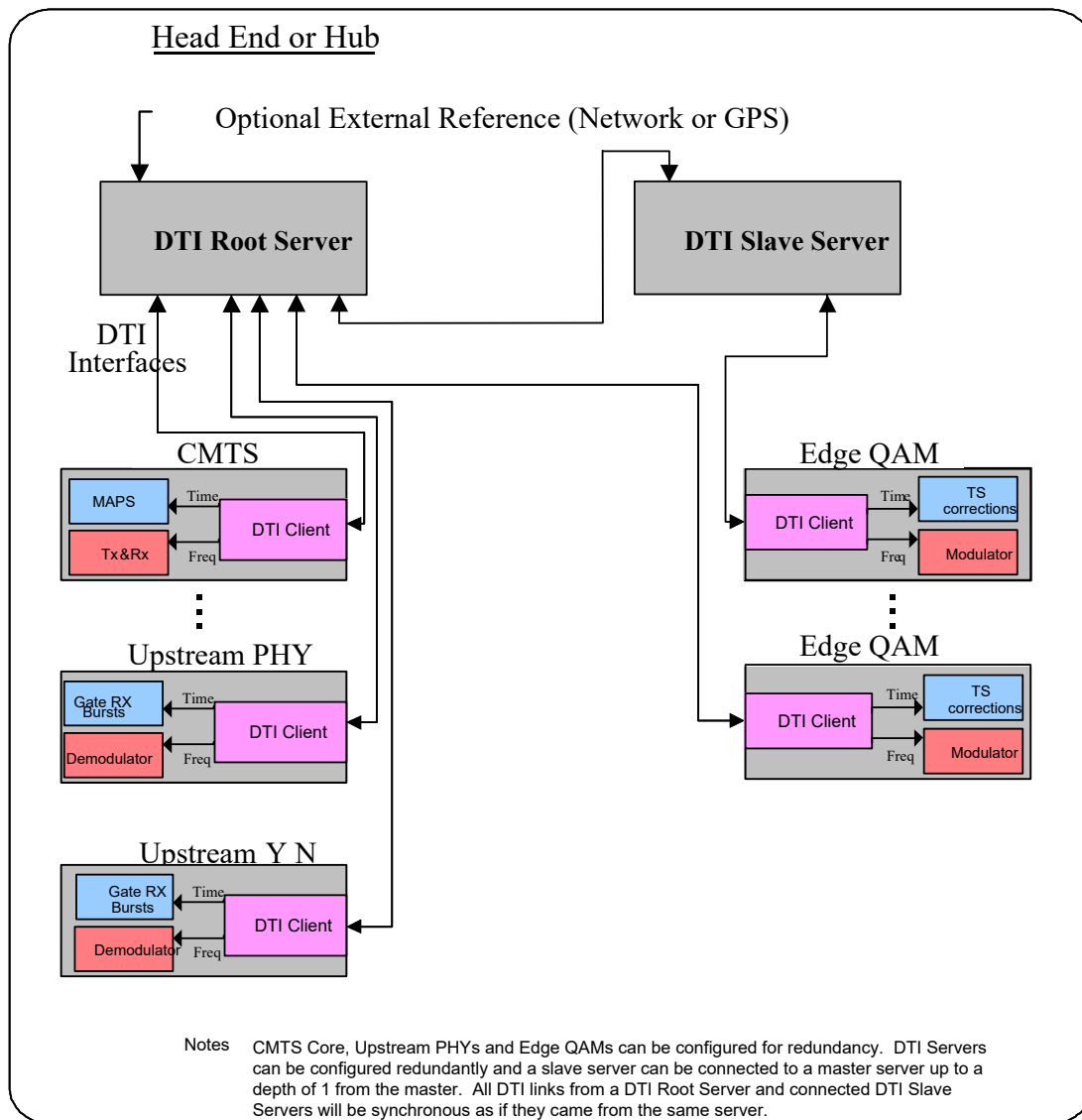


Figure 1–1 - Timing Architecture

Figure 1–1 shows frequency and timing distribution examples for both headend and Hub. The DTI Server establishes the reference for the timing distribution network and synchronizes all connected DTI Clients via point-to-point connections between the server and each client. A single protocol initiated by the DTI server permits the client to perform frequency and time synchronization. As shown, upstream receive, Edge QAMs, and the M-CMTS CORE may have different uses for the synchronized frequency and time, but utilize a common client function.

The DTI protocol and server-client interactions are described in detail in Sections 6 and 0. The essential characteristics are:

- The DTI server initiates the protocol, which the DTI client uses to establish its time and frequency synchronization.
- Using a ping-pong scheme, the client always immediately replies to the DTI server when it receives a transmission from the DTI server. The server uses this response to auto-compensate any delays with the effect that the client becomes precisely synchronized to the server.

- The server-to-client-to-server handshake continually repeats, assuring that a tight synchronization can be maintained.

The DTI protocol and components support accurate and robust transport of the server 10.24 MHz master clock and 32-bit DOCSIS timestamp to the client within a node or building. The protocol is structured to minimize the complexity and cost of the client clocks and the per port cost of the shared server function while supporting all the DOCSIS S-CDMA, TDMA, and future T-Services timing requirements in a modular system.

The high accuracy (<5 ns) and high stability (<1 ns timing jitter budget) is achieved by using a simple ping-pong layer 2 timing protocol over a single twisted pair connection using common passive PHY components in both directions. This structure provides delay reciprocity so that all cable delay processing can be performed in the server. The client's role in delay correction is to provide a fixed delay response to the server frame and to use the cable advance supplied by the server to advance the local 10 kHz DTI frame clock to correct for cable delay.

To ensure reliable transport and client clock operation, the client clock is required to report the current phase error of its local clock (frame clock) with respect to the delay-corrected server frame clock. This measurement is reported to the server at the 10 kHz frame rate. The server's role is to process this measurement data and verify the client's timing operation. This protocol supports real-time detection and mitigation of client clock faults.

The DTI client can be realized with a single digital component, a simple PHY and a low cost local oscillator, as holdover and filtering are supported in the shared server. A common definition of the DTI high-speed clock is necessary to ensure compatibility between all DOCSIS DTI client components.

### 1.1.5 Synchronization Needed for T-Services Deployment

The deployment of T-Services compliant with the existing Telco T1/E1 standards will require both synchronization and traceability to a common external clock source. In this case, if a cable modem supporting T-Services is connected to an M-CMTS EQAM, the cable modem will need to be synchronized with the DTI Server operating with an external T-Service reference.

## 1.2 Requirements

Throughout this document, the words that are used to define the significance of particular requirements are capitalized. These words are:

"MUST"	This word means that the item is an absolute requirement of this specification.
"MUST NOT"	This phrase means that the item is an absolute prohibition of this specification.
"SHOULD"	This word means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and the case carefully weighed before choosing a different course.
"SHOULD NOT"	This phrase means that there may exist valid reasons in particular circumstances when the listed behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
"MAY"	This word means that this item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because it enhances the product, for example; another vendor may omit the same item.

## 2 REFERENCES

### 2.1 Normative References

The following documents contain provisions, which, through reference in this text, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreement based on this standard are encouraged to investigate the possibility of applying the most recent editions of the documents listed below.

- [RFI2] ANSI/SCTE 79-1 2016 DOCSIS 2.0 Part 1: Radio Frequency Interface
- [G.812] ITU-T Recommendation. Gr.812, Timing Requirements of slave clocks suitable for use as node clocks in synchronization networks, June 1998.
- [G.823] ITU-T Recommendation. Gr. 823, The control of jitter and wander within digital networks which are based on the 2048 kbit/s hierarchy, March 2000.
- [G.824] ITU-T Recommendation. Gr. 824, The control of jitter and wander within digital networks which are based on the 1544 kbit/s hierarchy, March 2000.
- [GR1244] Telcordia, Clocks for the Synchronization Network: Common Generic Requirements, December 2000.
- [IEEE 802.3] IEEE Std. 802.3-2002, Part3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) access method and physical layer specifications, March 8, 2002.
- [T1.101] ANSI T1.101, Synchronization Interface Standard T1.101-1999, September 7, 1999.

### 2.2 Informative References

This specification uses the following informative references.

- [DEPI] SCTE 137-2 2017, DOCSIS Downstream External PHY Interface.
- [ERMI] SCTE 137-4 2017, Edge Resource Manager Interface.
- [MOSSI] SCTE 137-3 2017, M-CMTS Operations Support System Interface.

### 2.3 Reference Acquisition

- American Institute of Electrical Engineers, Internet: <http://www.ieee.org/portal/site>
- American National Standards Institute, Internet: <http://webstore.ansi.org>
- International Telecommunications Union (ITU), Internet: <http://www.itu.int>
- Internet Engineering Task Force (IETF), Internet: [www.ietf.org](http://www.ietf.org)
- Telcordia Technologies, Internet: <http://telecom-info.telcordia.com>

### 3 TERMS AND DEFINITIONS

This specification uses the following terms:

<b>bridging mode</b>	A short-term operating condition of the DTI clock where the DTI client has recently lost its controlling input and is using stored data, acquired while in normal or fast mode operation, to control its output. While in bridging the degree of deviation of the output is deemed to be such that DTI client clock is still performing within normal or acceptable limits. If an outage period persists the DTI client clock will transition to the holdover mode indicating that the DTI client clock output may be degraded.
<b>DTI Minimum Clock oscillator</b>	An oscillator that supports all the client clock performance requirements with holdover limited to the minimum bridging time. A non-ovenized oscillator can be used to support this oscillator category.
<b>fast mode</b>	An operating condition of a clock in which it is locked to an external reference and is using time constants, which are reduced to quickly bring the local oscillator's frequency into approximate agreement with the synchronization reference frequency.
<b>free-run mode</b>	An operating condition of a DTI clock whose output signals are internally controlled by the DTI server. The clock has never had, or has lost, external reference input and has no access to stored data that was acquired from a previously connected external reference during the time after the last power cycle. Free-run ends when the clock output is influenced by an external reference or the process to achieve lock to an external reference. Free-run may provide needed stability when external reference has been lost or not equipped.
<b>geoid</b>	The equipotential surface of the Earth's gravity field which best fits, in a least squares sense, global mean sea level.
<b>holdover mode</b>	An operating condition of a DTI clock that has lost its controlling input and is using stored data, acquired while in normal or fast mode operation, to control its output. The stored data is filtered to minimize the effects of short-term variations and to establish a predictor of oscillator behavior during the reference outage. This permits the output deviation from normal operation to be minimized.
<b>Maximum time interval error (MTIE)</b>	<p>For a sequence of time delay samples <math>x_i</math>, MTIE at observation time (S) is:</p> $MTIE(S) = \max_{j=1}^{N-n+1} \left[ \max_{i=j}^{n+j-1} (x_i) - \min_{i=j}^{n+j-1} (x_i) \right]$ <p>MTIE Measurement:          where:  <math>\tau_0</math> = sample period  <math>N</math> = number of samples in the sequence  <math>n = \lceil S/\tau_0 \rceil + 1</math>  <math>S</math> = observation time  <math>x_i</math> = time delay sample</p>
<b>normal mode</b>	An operating condition of a clock in which the output signals are controlled by an external input reference. The expected mode and state permits each clock within a distribution to have the same long-term average frequency and time. Clocks in this mode are referred to as locked, meaning that they are in tight relationship with the DTI root clock. A DTI server clock in a fault free free-run mode will be considered in normal mode
<b>Root DTI Server:</b>	The DTI server that is the source of traceable time and frequency for all subtending DTI servers and clients in a building.

## 4 ABBREVIATIONS

This specification uses the following abbreviations:

<b>(EQAM), Edge QAM</b>	A network element which receives MPEG-TS frames over a network interface such as Ethernet, and modulates them onto QAM carriers for use on a HFC plant.
<b>CM</b>	Cable Modem
<b>CMTS</b>	Cable Modem Termination System
<b>CRC</b>	Cyclic Redundancy Check
<b>DEPI</b>	Downstream External PHY Interface
<b>DOCSIS®</b>	Data-Over-Cable Service Interface Specifications
<b>DS</b>	Downstream
<b>DTI</b>	DOCSIS Timing Interface
<b>DTS</b>	32-bit DOCSIS Time Stamp
<b>ERMI</b>	Edge Resource Manager Interface
<b>GE</b>	Gigabit Ethernet (1 Gbps)
<b>IE</b>	Information Element. An element of a MAP message.
<b>IP</b>	Internet Protocol
<b>MAC</b>	Media Access Control. Used to refer to the layer 2 element of the system, which would include DOCSIS framing and signaling.
<b>M-CMTS</b>	Modular CMTS
<b>MPEG</b>	Motion Picture Experts Group
<b>MPEG-TS</b>	Motion Picture Experts Group Transport Stream
<b>MTIE</b>	Maximum Time Interval Error
<b>PCR</b>	Program Clock Reference
<b>PHY</b>	Physical Layer. Used to refer to the downstream QAM transmitters and the upstream burst demodulators (receiver).
<b>PID</b>	Packet Identifier used in MPEG-TS
<b>PUSI</b>	Payload Unit Start Indicator
<b>QAM</b>	Quadrature Amplitude Modulation

<b>S-CDMA</b>	Synchronous Code Division Multiple Access
<b>TBD</b>	To Be Determined (or To Be Deferred)
<b>TDM</b>	Time Division Multiplexing
<b>TDMA</b>	Time Division Multiple Access
<b>T-Services</b>	Legacy T1 or T3 voice and/or data transport
<b>UDP</b>	User Datagram Protocol
<b>US</b>	Upstream
<b>UTC</b>	Coordinated Universal Time; also known as Greenwich Mean Time (GMT) or Zulu time

## 5 PHYSICAL LAYER REQUIREMENTS

This section is normative and specifies the physical layer requirements of the DTI protocol.

### 5.1 Introduction

The DTI link comprises the connection between a DTI Server and various other elements, such as M-CMTS CORE, EQAM and US Receivers collocated in the node that derive their clocking (time and frequency) from the DTI Server. Since the link resembles an Ethernet (802.3) 10BaseT link, it is advantageous to leverage the availability and cost effectiveness of this standard. This document defines the differences between the DOCSIS Timing Interface and a conventional 802.3-10BaseT interface. Table 5–1 indicates the similarity between the two links.

**Table 5–1 - Physical Layer Comparisons**

Characteristic	Ethernet 802.3-10BaseT	DOCSIS Timing Interface
Data Rate (Mbps)	10	5.12
Accuracy	Free-running; 100 ppm	Traceable to Master Clock
Transmission mode(s)	Full/half Duplex <sup>(1)</sup>	Ping-Pong <sup>(1)</sup>
Topology	Star <sup>(2)</sup>	Star <sup>(2)</sup>
Maximum Segment Length	100 m	200 m
Media	UTP	UTP
Signaling method	Baseband	Baseband
Modulation	Manchester	Manchester

**Table Notes:**

1. Conventional Ethernet 802.3 transceivers use separate wire pairs for transmission in the two directions. The DTI uses a ping-pong scheme whereby the same pair is used for transmission in both directions. This ensures the maximal reciprocity, minimizing the asymmetry in transmission delay between the two directions and minimizes crosstalk.
2. Conventional Ethernet installations utilize a star-wiring configuration where the "common" point is a switch or hub. In the DTI scenario the common point is the DTI Server.

The governing standard for Ethernet (physical layer) is [IEEE 802.3]. The physical layer specifications (primarily section 14 of the standard) are applicable here.

### 5.2 Physical Connector Description

The DTI Server, as well as the DTI Client, MUST have an RJ-45 (female) connector for each DTI link. This permits conventional Ethernet cabling techniques to be applied. See [IEEE 802.3] section 14.5.1. One distinction is that the DTI utilizes a single pair for transmission in both directions. Therefore a crossover function is not required. Contacts 1 and 2, labeled "SIG+" and "SIG–" will be used for the ping-pong transmission.

The cable interconnect MUST be as defined in Table 5–2.

**Table 5–2 - DTI RJ45 Interconnect**

RJ45 Pin Number	Signal	Comment
1	SIG+	10BaseT Compliant
2	SIG-	10BaseT Compliant
3	NC	
4	NC	



RJ45 Pin Number	Signal	Comment
5	NC	
6	NC	
7	NC	
8	NC	

### 5.3 Cable Requirements

The DTI MUST operate normally over a maximum of 200 meters of UTP cable rated at category 5E or better.

### 5.4 Electrical Description

#### 5.4.1 Impedance

The termination impedance MUST be 100 ohms. The differential output impedance as measured on the TD circuit MUST be compliant with [IEEE 802.3] section 14.3.1.2.2. The transmitter impedance balance, or the ratio of common-mode to differential-mode impedance balance of the TD circuit MUST exceed  $29 - 17 \log_{10}(f/10)$  dB, where  $f$  is the frequency in MHz, over the frequency range 1.0 MHz to 20 MHz (see [IEEE 802.3] section 14.3.1.2.4).

#### 5.4.2 Isolation

There MUST be isolation between the physical layer circuits, including frame ground, and all leads including those not used by the DTI link. The isolation requirement follows [IEEE 802.3] section 14.3.1.

#### 5.4.3 EMI Considerations

The DTI link will comply with applicable local and national codes for limitation of electromagnetic interference.

#### 5.4.4 Signal Strength (voltage)

The per symbol peak differential voltage on the SIG± signals at the DTI Server, when terminated in a 100 Ω resistive load MUST be between 2.2 Vpp (Volts peak-to-peak) and 2.8 Vpp for all data sequences. For an all-ones Manchester-encoded pattern, any even harmonic measured on the TD circuit MUST be at least 27 dB below the fundamental. The magnitude of the total common-mode output voltage MUST be less than 50 mV peak.

#### 5.4.5 Common-mode rejection

Given application of common-mode voltage of 15 V peak ~10 MHz sine wave (see [IEEE 802.3] section 14.3.1.2.6) the differential-mode voltage MUST NOT change by more than 100 mV for all data sequences. The edge jitter introduced MUST be less than 3 ns.

#### 5.4.6 Signal Description

The transmission between the DTI Server and the DTI client MUST be ping-pong in nature, using the same pair of wires for transmission in both directions. The data pattern MUST be Manchester encoded with an underlying bit-rate of 5.12 Mbps, locked to the DTI Master Clock.

## 6 DOCSIS TIMING PROTOCOL

This section is normative. See Section 1.1 for an informational description of the DTI protocol.

### 6.1 DTI Timing Entities

The DTI protocol is supported by two terminated entities.

1. DTI Server
2. DTI Client

DTI Server and Client functional entities are part of M-CMTS architecture. The DOCSIS Client function is structured to permit a low cost client clock function within all EQAMs, upstream receivers, and M-CMTS CORE entities.

The DTI Server SHOULD support multiple client functions to support scalable growth of DTI port capacity within a node or building.

The DTI Server MUST support an SNMP management interface, IP addressing, and use RJ45 connectors.

DTI server in a single building MAY consist of active and backup servers components in a common shelf and/or DTI servers in subtending shelves.

A DTI server MAY support at least one of the following two capabilities:

1. Operation as a subtending server.
2. Provide DTI outputs capable of supporting DTI subtending servers.

The following accuracy requirements may be confirmed with a delay-calibrated client. A delay-calibrated client provides sufficient measurement points to determine current calibrated delay bias.

All DTI root server outputs MUST meet a 1.25 ns accuracy requirement when tested with a delay calibrated client with respect to the DTI root server master clock test port with correction for test port group delay.

All DTI subtending server outputs MUST meet a 2.5 ns accuracy requirement when tested with a delay calibrated client with respect to the DTI root server master clock test port with correction for test port group delay.

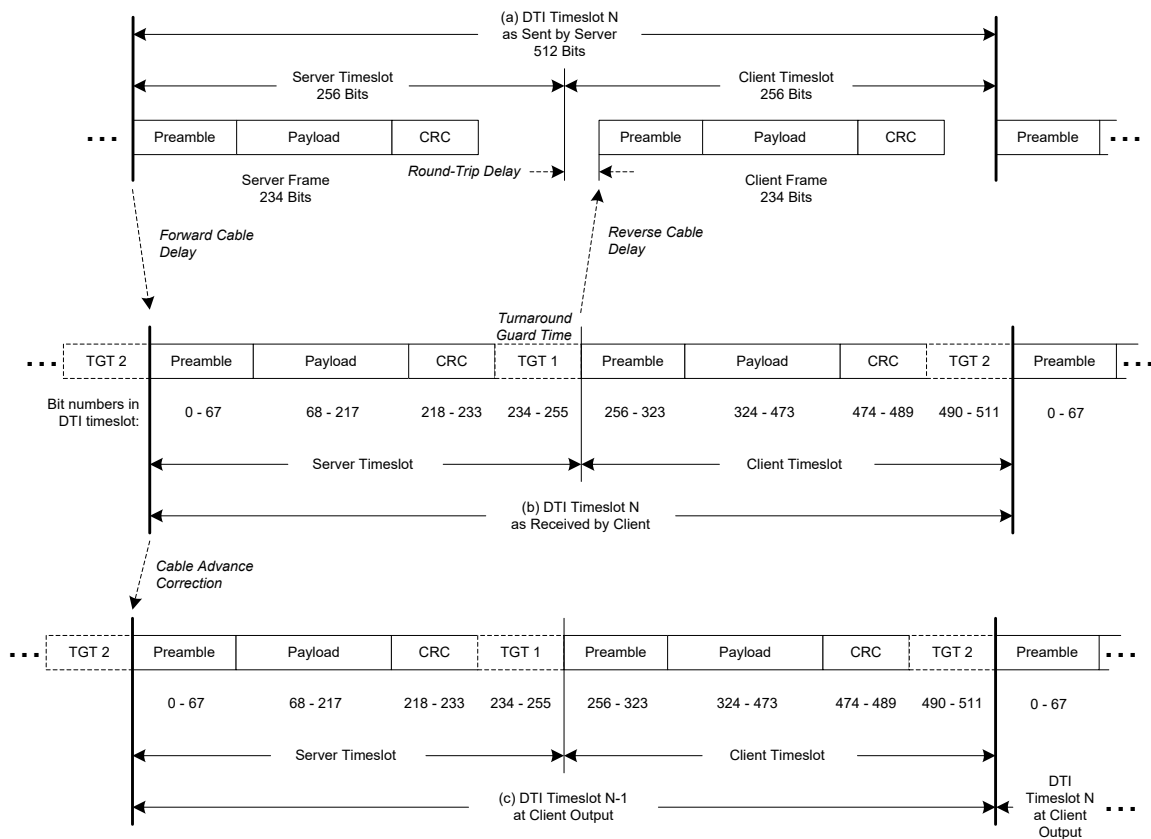
A DTI output that can support a subtending DTI server MUST meet a 1.25 ns accuracy requirement when tested with delay-calibrated client under normal operating conditions. An M-CMTS EQAM, M-CMTS CORE or a separate upstream receiver MUST support at least one DTI client interface.

A DTI server in slave mode is intended to only accept a DTI input from a root server.

An M-CMTS device MAY support multiple DTI clients if protection for a DTI interconnect cable loss is desired.

If an M-CMTS device has more than one DTI client, it MUST not switch to a protection client as a result of a loss of DTI input until a minimum timeout of 500 ms.

## 6.2 DTI Timing Structure



**Figure 6-1 - DTI Protocol Timing Structure**

Figure 6-1 defines the structure of the DOCSIS Timing Protocol as (a) sent by the Server, (b) received by the Client after the cable delay, and (c) output at the Client Test Port after the cable advance correction. The highest-level structure is the DTI Timeslot. A DTI Timeslot occupies 1024 master clock (10.24 MHz) periods spanning  $1/10 \text{ kHz} = 100 \text{ us}$ . A bit period consists of two consecutive master clock counts starting with an even count, resulting in 512 bit periods in a DTI timeslot. The timeslot is synchronized to the DOCSIS 32-bit timestamp (DTS), such that the current bit period of a timeslot is the integer, as follows:

$$\text{current\_bit\_count} = \text{floor}[(\text{DTS mod } 1024)/2]$$

The Client's received DTI Timeslot timing is offset from the Server's DTI Timeslot timing by the cable propagation delay (plus any group and logic delays). Since the Client when in steady state tracks the Server, the Client counts off a continuous succession of bit slots timed on its own 5.12 MHz bit clock, repeating DTI Timeslots seamlessly modulo 512 bits. The Client always receives the first preamble bit from the Server in bit slot 0. When it transmits, the Client MUST transmit the first bit of its preamble in bit slot 256.

The DTI protocol is ping-pong in nature. The Server and Client share the line using time division duplexing (TDD), one transmitting while the other receives. The DTI timeslot is divided into two equal intervals, the Server Timeslot and Client Timeslot. Layer 2 communication protocol data units (PDUs) are termed Frames. Both Server and Client Frames are 234 bits in length, and each is followed by a 22 bit Turnaround Guard Time (TGT). The purpose of TGT1, located after the Server Frame, is to allow time for the Client to complete its CRC processing and perform receive/transmit switching on the line after receipt of the Server Frame. The purpose of TGT2, located after the Client Frame, is to provide for the round-trip delay of the cable, ensuring that the Client Frame has been received by the Server before the Server begins transmitting the next Server Frame, including time for the Server to perform

receive/transmit switching on the line after receipt of the Client Frame. Each TGT provides approximately 4.3  $\mu$ s of guard time, which well exceeds the maximum round-trip delay for 200 m cable of approximately 2  $\mu$ s.

At the Client output, the DTI Frame timing is adjusted earlier in time by the cable advance correction sent by the Server, in order to align the Client output frame timing closely with the Server frame timing. To allow for causality, the data received during the previous frame is buffered and output by the Client after the CRC check has completed.

### 6.3 Traceability of DOCSIS Timestamp

Time of Coincidence (TOC) is an important concept to understand the relationship of any DTI Timestamp (DTS) 32-bit counter to GPS system time. GPS system time began on January 6, 1980. GPS receivers will provide a 32-bit gpssec timestamp that was zero at the January 6 start epoch. The gpssec is a 32-bit timestamp counter that is incremented every second. The DTS is also a 32-bit timestamp counter that is incremented every 10.24 MHz master clock. The objective is to map the current GPS system time to the current DOCSIS Timestamp in a coherent manner.

By definition, the DTS can be assigned the value of zero at the same January 6, 1980 start epoch. At the next second, the DOCSIS timestamp will advance 10,000 timeslots with 1024 master clocks per timeslot, so the DTS should be 10240000.

The TOC is the next integer gpssec when the DTS will be exactly zero. This will occur every 262144 seconds (~ every 3 days)<sup>1</sup>. If a reset to zero process were used to synchronize every DTS counter in a server then it would require up to 3 days to align a server at start-up. The approach is to generate an initial value for DTS by a mapping function so that the alignment can occur on any one second boundary.

The mapping function from gpssec to DTS MUST be:

$$DTS = 2^{10} * [10,000 * (\text{gpssec} \bmod 262144)] \bmod 2^{22}$$

Or,

The mapping function from gpssec to DTS MUST be:

$$DTS = [2^{10} * \{10,000 * (\text{gpssec} \bmod 262144)\}] \bmod 2^{32}$$

The four time-of-day setting modes are:

1. GPS
2. User Time Set
3. Default Time Set
4. Network Timing Protocol (NTP) version 4 or greater

The GPS traceability mode provides the most precise time setting capability. Regardless of the Time of Day mode, the DTI server MUST convert time of day information to the equivalent gpssec value.

The user time set permits a user to enter an approximate time of day setting through either a local or remote user interface. The default time set operation, if selected, will establish a coarse time of day setting after a reset or power cycle. The time of day is based on the current value of the real time clock in the server.

The NTP time set mode if supported will allow the DTI server to establish a time setting via the NTP protocol as currently configured in the DTI server. If NTP time mode is supported, the DTI server MUST support NTP version 3 or higher.

The DTI server MUST support a means to configure the time of day setting modes of User Time Set, Default Time Set, NTP or GPS.

---

<sup>1</sup> The gpssec timescale will rollover every 136 years. Since there are exactly 16,384 TOC events between rollovers, there is no disruption of the DTS.

### 6.3.1 GPS Frequency Mode

In order to provide continuity in the DTI timebase when transitioning from free run mode or network mode to GPS mode, an intermediate state, "GPS frequency mode," is defined. When the DTI server transitions from free run or network mode to GPS frequency mode, the DTI server's 10.24 MHz output frequency is adjusted smoothly so as to track the GPS system frequency, while maintaining continuity of the DTI time counters. The system may remain in GPS frequency mode for an extended period of time. If and when the DTI server transitions from GPS frequency mode to GPS mode, the DTI time counters are realigned to GPS time; this may be done in a discontinuous manner during a scheduled maintenance period.

When transitioning from free-run or network mode to GPS mode, the DTI server **MUST** enter the GPS frequency mode first.

In GPS frequency mode the DTI server **MUST** maintain the existing time setting mode and validity state unless changed as a result of user time setting input.

The DTI server **MUST** support a user input to schedule a transition time from GPS frequency mode to GPS mode.

The DTI server operating in normal lock in GPS frequency mode **MUST** meet the same MTIE requirements as in GPS mode.

During the transition from free-run to GPS frequency mode normal lock, all DTI server free run requirements **MUST** be met.

During the transition from normal lock network to GPS frequency mode normal lock, all DTI server normal lock network MTIE requirements **MUST** be met.

Under extended GPS holdover conditions it is possible to enter the degraded MTIE region of performance. The DTI server needs to support a graceful recovery from this state if it should arise. The following requirement applies:

When recovery from a GPS degraded condition, the DTI server **MUST** support the ability to slew the frequency to accommodate at least 1 ms of offset over a recovery period of less than 24 hours without exceeding the free run performance limits.

In user mode and default mode there are no requirements to adjust the DTI timescale and TOD output after the initial time setting. The DTI time services are for local use only and not traceable to GPS system time.

If the DTI server supports frequency adjustment of the timescale in NTP mode, the vendor **SHOULD** specify the time accuracy performance under vendor defined NTP configuration(s).

If the DTI server supports frequency adjustment of the timescale in NTP mode, the DTI server output performance requirements **MUST** meet the performance requirement of the system operation mode (freerun or network).

If frequency adjustment of the timescale in NTP mode needs to be suspending to prevent degradation of output, the DTI server **MUST** report this condition.

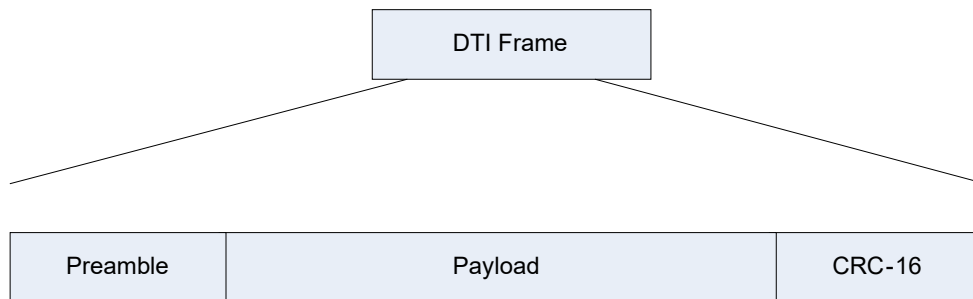
If the user requests a transition from user or default time setting to NTP after warmup, the DTI server **MAY** reject the request if the transition would degrade the output performance.

### 6.4 DTI Frame Structure Requirements

The DTI frame structure **MUST** consist of a preamble followed by the payload and finally by a 16 bit CRC as shown in Figure 6–2. The length of the DTI frame, including the preamble and the CRC **MUST** be 234 bits. The CRC **MUST** be calculated only on the payload bits. The rising edge of the 10 kHz DTI frame clock **MUST** be aligned with the leading edge of the first bit of the preamble of the DTI server. The client digital clock recovery circuit **MUST** be tolerant to missing clocks since any bit error in the frame will prevent the CRC OK flag in the client CRC checker from being asserted.

### 6.4.1 Conventions for this Specification

In this specification the following convention applies any time a bit field is displayed in a figure. The bit field should be interpreted by reading the figure from left to right, then from top to bottom, with the MSB being the first bit so read and the LSB being the last bit so read. The DTI frames are transmitted most significant element first. Within bit fields, the most significant bit MUST always transmitted on the wire first. There are nine bit fields that constitute a frame in each direction. Fields MUST transmit in numerical order from lower to highest value (see Table 6–1 and Table 6–2 for field order).



**DTI Frame Structure**

*Figure 6–2 - DTI Frame Structure*

### 6.4.2 SERVER TO CLIENT

The frame structure in the server to client direction MUST operate as shown below in Table 6–1.

**Table 6–1 - DTI Server Frame Structure**

DTI Server Frame Structure			
FIELD	NAME	SIZE (Bits)	DESCRIPTION
1	PREAMBLE	68	Preamble of 0xAAAA AAAA AAAA AAAA 9 prior to Manchester encoding
2	DEVICE TYPE	8	Byte describing type of server.
3	SERVER STATUS FLAGS	8	8 flag bits identifying server status.
4	DOCSIS UPPER TIMESTAMP	22	22 Most Significant Bits of the DTS
5	TIME OF DAY	10	Field supports serial TOD message over multiple frames.
6	CABLE ADVANCE	24	Integer and Fractional Cable Advance
7	PATH TRACEABILITY FIELD	10	Field supports serial Path Traceability Message over multiple frames.
8	RESERVED	68	All bits set to one.
9	CRC16	16	16 bit CRC, which covers all bits except preamble.
	Total Payload Bits	234	

In the server to client direction, the DTI frame structure details are as follows.

#### **6.4.2.1.1 Preamble**

The 68-bit preamble is used to align the digital clock recovery circuit on the client and to locate the bit transitions in the Manchester coding that contain information. 0x9 "1001" pattern at the end of the preamble locates the beginning of the payload.

#### **6.4.2.1.2 Device type**

The 8-bit device type field is used to identify the type of server and its timing source. The bits in this field **MUST** be assigned as shown below:

##### **Bits 7:5 External Timing Source (for root server)**

- 000: Server has no external timing source
- 001: Server is receiving external time and frequency from GPS
- 010: Server is receiving timing from a network
- 011: Server is receiving external frequency only from GPS (no time)
- 111: Reserved

##### **Bit 4:3 Server Hop Count**

- 00: Server is the root DTI Server for the office distribution
- 01: Server is directly connected to root DTI server.
- 10-11: Reserved

##### **Bits 2:0 Root Server Clock Type:**

- 000: Server clock is ITU type I
- 001: Server clock is ITU type II
- 010: Server clock is ITU type III
- 011: Server clock is ANSI T1.101 ST3 (back-up operation only)
- 100-111 reserved

#### **6.4.2.1.3 Server Status Flags**

This 8-bit field is used to send the server status to a client. The information in this field relates to the DTI Server transmitting the DTI protocol. The bits **MUST** be assigned as shown below:

##### **Bit 7: Reserved**

##### **Bit 6: Client Performance Stable**

This bit, when set to 1, indicates that the server verifies the client phase error measurement is within acceptable operating performance.

##### **Bit 5: Cable Advance**

This bit, when set to 1, indicates that the cable delay has been calculated and that the value in the cable advance field is valid.

##### **Bit 4: Holdover mode**

This bit, when set to 1, indicates that the server has lost its timing reference and is in holdover.

##### **Bit 3: Normal mode**

This bit, when set to 1, indicates that the clock is stable, has locked on to the timing reference, and is compliant with the appropriate clock standard.

**Bit 2: Fast mode**

This bit, when set to 1, indicates that the clock is using a short time constant. A shorter time constant is used in the clock circuit to shorten the initial lock time when the server first powers up or receives a reference for the first time.

**Bit 1: Freerun mode**

This bit, when set to 1, indicates that the server is operating with output frequency that has not been influenced by local external reference signals.

**Bit 0: Warmup**

This bit is set to 1 to indicate that the reference oscillator has not stabilized yet.

**6.4.2.1.4 DOCSIS Upper Time Stamp**

These 22 bits MUST contain the most significant portion of the DOCSIS 32-bit timestamp, which is the portion of the timestamp that remains constant for an entire DTI Frame period (1/10 kHz = 100 us). These bits are used to load and/or monitor the upper 22 bits of the 32-bit DOCSIS timestamp counter in the client. The 10 least significant bits of the DOCSIS timestamp counter in the client represent the number of 10.24 MHz clocks since the beginning of the DTI Frame. Together, these two fields create the entire 32-bit DOCSIS timestamp.

**6.4.2.1.5 Time of Day**

The Time of Day (TOD) message provides time in binary format and optionally in ASCII format, each with 1-second resolution. The gpssec time and leap second are sent in binary format. When in verbose mode, calendar time in ASCII format is also sent, including a modified Julian date and local date/time information. The TOD message is transmitted at a rate of 10 bits (one byte of payload and two control bits) per DTI frame, and is subcommutated to span multiple DTI frames. The first control bit indicates that the location of the pulse per second will be coincident with the beginning of the next frame. The second control bit is a data valid flag which applies to the payload byte. The data valid flag may be used to stop and start the subcommutated byte stream. When the data valid flag is 0, the payload byte MUST contain all ones. When the data valid flag, is 1, the payload byte MUST contain the next serial byte in the TOD message. The TOD message corresponding to a given PPS count MUST start transmission one or more frames after the frame where the PPS flag bit has been set, and complete transmission within 100 ms. The bit assignments for the 10-bit TOD message field in each DTI frame MUST be as follows:

**Bit 9: PPS flag**

This bit MUST be set to '1' when the beginning of the next frame is coincident with the DTI server pulse per second<sup>2</sup>. The asserted "1" indicates that the next DTI server frame PPS flag position bit is the on-time mark for PPS information just transferred during the last one second period. <sup>3</sup>The PPS flag bit is asserted in every 10,000<sup>th</sup> frame.

**Bit 8: Data Valid bit**

This bit is set to '1' to indicate that the data in the payload byte (following eight bits) contains valid data.

---

<sup>2</sup> The Pulse Per Second is free-running except in GPS mode where it aligned to the GPS 1PPS.



**Bit 7-0:**

This field contains the subcommutated TOD payload byte, which MUST be in accordance with Table 6–2.

**Table 6–2 - TOD subcommutated message format**

Byte Number	Length	Type	Format	Name	Description
1	1	Binary	See TOD Status Field description below this table.	TOD Status	
2-5	4	Binary		gpssec	32-bit gpssec timestamp (MSB is most significant byte is zero)
6	1	Binary		Leap Seconds	Cumulative leap seconds between gpssec and UTC
7	1	7-bit ASCII	ASCII "*" (0x2A) denotes Valid TOD Calendar Time ASCII "!" (0x21) denotes Invalid Calendar Time	TOD Calendar Time Valid	
8-12	5	7-bit ASCII	MMMMM Where M is [0-9] ASCII	MJD	Modified Julian Date (number of days since November 17,1958)
13	1	7-bit ASCII	ASCII Decimal Point "." (0x2E)		
14-23	10	7-bit ASCII	Year/Month/Day Year::[0000-9999] Month:[1-12] Day:[1-31]	Date	Local Date
24	1	7-bit ASCII	ASCII Decimal Point "." (0x2E)		
25-32	8	7-bit ASCII	Hour:Min:Sec Hour:[00-23] Min:[00-59] Sec:[00-60] <sup>4</sup>	Time	Local Time
33	1	7-bit ASCII	ASCII Decimal Point "." (0x2E)		
34:38	5	7-bit ASCII	SHH.F S: sign [+ , -] H: offset in hours F: [0] or ([5] for 30 minute offsets)	Time Zone Offset	Local time zone offset
39	1	7-bit ASCII	ASCII Decimal Point "." (0x2E)		
40 <sup>5</sup>	1	7-bit ASCII	Leap Leap: +, 0, -	Leap Second Indicator	"+" for adding leapsecond "0" for no leapsecond "-" for subtracting leapsecond
41	1	7-bit ASCII	ASCII Carriage Return (0x0D)		

**6.4.2.1.6 TOD Status Field**

The 8-bit is used to report the status of the Time Of Day Message. The bits in the TOD Status Field MUST be assigned as shown below:

**Bits 7: 4 Time Setting Mode**

<sup>4</sup> 60 second indicator may be used to indicate a leap second

- 0000: Default Time Setting Mode
- 0001: User Time Setting Mode
- 0010: NTP Time Setting Mode
- 0011: GPS Time Setting Mode
- 0100-1111: Reserved

**Bit 3:2 TOD state**

- 00: TOD is currently not valid
- 01: TOD is valid
- 10-11: Reserved

**Bits 1:0 TOD Message Mode**

- 00: Short Message Mode. Message bytes 0-5 (binary only)
- 01: Verbose Message Mode: Message includes all fields in Table 6–2
- 10-11: Reserved

The DTI server **MUST** support the Short Message Mode for TOD delivery.

The Verbose Message Mode **SHOULD** be supported by the DTI server.

If more than one TOD message mode is supported the default **MUST** be the Short Message Mode.

If more than one TOD message mode is supported the mode **SHOULD** be configurable on a port basis.

**6.4.2.1.7 Cable Advance**

Cable Advance is described in 7.1.2. The cable advance information **MUST** be contained in this 24-bit field. The 16 most significant bits of the cable advance **MUST** contain the integer portion of the cable advance value that is in 149.8 MHz<sup>6</sup> sample clock cycles. The remaining 8 bits, the least significant byte of the cable advance, **MUST** be the fractional portion of the cable advance and is in 1/256 of a 149.8 MHz clock cycle.

**6.4.2.1.8 Path Traceability Field**

The path traceability field is used to transfer messages containing path information from a server to a client. A path message is a concatenated set of type, length, value (TLV) data items as defined in this section. Sub-commutating the path message over multiple DTI timeslots supports the transport of path messages. A path message slot is a 10ms time interval and **MUST** begin when the DTI upper time-stamp modulo 100 is equal to zero. By design the maximum length of a path message **MUST** be less than or equal to 64 bytes. If a message is pending, the server **MUST** assert the start of message flag within 1 ms of the start of a path message slot and **MUST** complete delivery within 9 ms of the start a path message slot. The message transmission **MUST** start coincident with the start of message flag or one frame after the start of message flag.

The DTI server **MUST** support path traceability message. For the path traceability message the maximum transfer rate **MUST** be less than or equal to one message per second. For the path traceability message the minimum transfer rate **MUST** be greater than or equal to one message every 5 seconds. The path traceability message **MUST** utilize the first path message slot after the frame where the PPS flag bit has been set.

After a path traceability message is transferred, the DTI server **MUST NOT** transmit other messages until the next 1PPS event.

The Path Traceability Field is 10-bit field that **MUST** be used to send byte oriented data to the client. The path traceability field **MUST** be filled as follows:

**Bit 9: Start of Message**

---

<sup>6</sup> The 149.8 MHz high-speed clock is more precisely 10.24MHz \* 512/35.

The path traceability field start of message bit MUST be set high for one frame to indicate the beginning of a status message.

**Bit 8: Data Valid Bit**

The path traceability field data valid bit MUST be set to ‘1’ to indicate that the next eight data bits contain valid data. When the path traceability field data valid bit is ‘0’, the client MUST ignore the contents of the eight data bits that follow.

**Bit 7:0: Data Byte**

This data byte MUST contain the serial message data bytes.

**Table 6–3 - Path Traceability Message Format**

Name	Type (1 byte)	Length (1 byte)	Value (variable length)
Root DTI Server IPV4 Address	1	4	IP address of Root DTI Server (highest level in tree)
Root DTI Server Output Port #	2	1	Root server output port number for this DTI interface. Output port number counting starts at 0 and goes through 255.
DTI Server IPV4 Address (if not root server)	3	4	IP address of DTI Server source DTI interface (if not root server)
DTI Server Output Port #	4	1	Server output port number for this DTI interface. Ports start counting at 0 and continue to 255.
Root DTI Server IPV6 Address	5	16	IP address of Root DTI Server (highest level in tree)
DTI Server IPV6 Address (if not root server)	6	16	IP address of Root DTI Server (highest level in tree)
Root DTI Server DTI Version <sup>7</sup>	7	1	DTI version number running on Root Server
DTI Server DTI Version	8	1	DTI version number running on DTI Server (if not root server)
EOT	9	1	0x00 The EOT identifies the end of the Path Traceability Field.

If there is subtending DTI Server, the root DTI server address MUST be passed through the subtending DTI server to DTI clients. In other words, a DTI client connected to a root server through an intermediate server will have both a Root Server IP address and a Server IP address in the path traceability field. The root server MUST not transmit the non-root IP information. The path traceability field MUST follow the Type, Length, and Value sequence, as shown below, and follows the bit/byte ordering already defined.

The DTI server MAY support the path location message (Table 6–4). The path location message provides the GPS positioning information (if available) associated with the root server. Since all clients are within a maximum radial distance of 400m from the root server, this position information can support position related algorithms, routing other related applications. For the path location message the maximum transfer rate MUST be less than or equal to one message per second. For the path location message the minimum transfer rate MUST be greater than or equal to one message every 5 seconds. A subtending server MUST relay the path location message from the root to all downstream DTI client ports.

<sup>7</sup> For DTI servers compliant with the first release of this standard this value shall be zero.

**Table 6–4 - Path Position Message Format**

Name	Type (1 byte)	Length (1 byte)	Value (variable length)
UTC Time Stamp	11	6	UTC Time of position fix "12": hh 00-23 hours "34":mm 00-59 minutes "56":ss 00-59 seconds
Latitude	12	9	"34": degree 00-90 "44": minute (integer) 00-59 "0000":minute (fractional) 0000-9999 "N": North/South N or S
Longitude	12	10	"135": degree 000-180 "44": minute (integer) 00-59 "0000":minute (fractional) 0000-9999 "E": East/West E or W
GPS Quality Indication	13	1	"0" Fix not available or invalid "1" Fix is valid
No. of satellites used for positioning	14	2	00-12
Dilution of Precision	15	5	02.34 Note 00.00 when position is interrupted or suspended
Altitude	16	9	"+" +/- sign relative to geoid 12345.6 00000.0 to 04000.0 "M" unit meters
EOT	17	1	0x00 The EOT marks the end of the Path Position Message.

**6.4.2.1.9 Reserved**

The remaining 68 bits MUST be reserved for future use. They will be filled with ‘1’s until defined otherwise.

**6.4.2.1.10 CRC 16**

The CRC-16 field MUST contain the 16 bit CRC16 cyclic redundancy check. The CRC-16 algorithm is defined as Annex C, DTI-S CRC-16.

**6.4.3 CLIENT TO SERVER**

The frame structure for the client to server direction MUST operate as shown below in Table 6–5.

**Table 6–5 - DTI Client Frame Structure.**

DTI CLIENT FRAME STRUCTURE			
Field	Name	Size	Description
1	PREAMBLE	68	Preamble of 0xAAAA AAAA AAAA AAAA 6 prior to Manchester encoding
2	DEVICE TYPE	8	Byte describing type of client

<b>DTI CLIENT FRAME STRUCTURE</b>			
<b>Field</b>	<b>Name</b>	<b>Size</b>	<b>Description</b>
3	CLIENT STATUS FLAGS	8	8 flag bits identifying client status
4	RESERVED	22	Not used; set to one.
5	RESERVED	10	Not used; set to one.
6	CLIENT CLOCK INTEGRATED PHASE	24	16-bit 2-complement value of the local frame clock
7	CLIENT DTI VERSION	10	Client DTI Version Number
8	RESERVED	68	All bits set to one.
9	CRC16	16	16 bit CRC, which covers all bits except preamble.
	Total Payload Bits	234	

In the client to server direction the DTI frame structure details are shown below.

**6.4.3.1.1 Preamble**

The 68-bit preamble is used to align the clock recovery circuit on the server and to locate the bit transitions in the Manchester coding that contain information. 0x6 "0110" pattern at the end of the preamble locates the beginning of the payload.

**6.4.3.1.2 Device type**

The 8-bit device type field is used to identify the type of client and its timing source. The bits in the device type MUST be defined as follows:

**Bits 7:4 Timing source**

Timing source does not have to be reported to the server

These bits are reserved and will be filled with 1's

**Bits 3:0 Client Clock Type**

0000: Client oscillator is an ITU type 1

0001: Client oscillator is an ITU type 2

0010: Client oscillator is an ITU type 3

0011: Client oscillator is an ITU Stratum 3

0100: Client oscillator is a DTI Minimum Clock oscillator

0101-1111: Reserved.

**6.4.3.1.3 Client Status Flags**

This 8-bit field is used to send the client status to a server. The client status flag bits MUST be defined as follows:

**Bits 6 & 7: Reserved for future use**

**Bit 5: Bridging mode**

This bit, when set to 1, indicates that the client has lost its timing reference and is maintaining acceptable performance.

**Bit 4: Holdover mode**

This bit, when set to 1, indicates that the client has lost its timing reference and is in holdover.

**Bit 3: Normal mode**

This bit, when set to 1, indicates that the clock is stable, has locked on to the timing reference, and is compliant with the appropriate clock standard.

**Bit 2: Fast mode**

This bit, when set to 1, indicates that the clock is using a short time constant. A shorter time constant can be used in the clock circuit to reduce the initial lock time when the client first powers up or receives a reference for the first time.

**Bit 1: Freerun mode**

This bit, when set to 1, indicates that the client is operating with output frequency that has not been influenced by a client or DTI server.

**Bit 0: Warmup**

This bit when set to 1 indicates that the client oscillator has not stabilized yet.

**6.4.3.1.4 Client Clock Integer Phase Error**

This 24-bit field MUST return a snapshot of the client phase error. The value will be in units of 149.8 MHz sample clock cycles and will reside in the 16 most significant bits. The lower eight bits of the 24-bit field MUST be padded with zeros and MUST NOT be used by the DTI server. The value MUST be a signed 2's complement number. If the DTI client supports more bits of resolution, the DTI client MUST round the reported value to the nearest integer sample clock cycle.

**6.4.3.1.5 Client DTI Version & Path**

This 10-bit field MUST report the current DTI protocol version supported by the client expressed as an unsigned integer.

This 10-bit field MAY support reverse communication of path information from the client to server. If reverse communication is not supported then this field MUST be set to zero.

The Client DTI Version and Path is used to transfer messages containing version and path information from a client to the server. The Client DTI Version and Path message is a concatenated set of type, length, value (TLV) data items as defined in Table 6–6. Sub-commutating the message over multiple DTI timeslots supports the transport of the messages. A message slot is a 10ms time interval and MUST begin when the receive DTI upper time-stamp modulo 100 is equal to zero. By design the maximum length of a message MUST be less than or equal to 64 bytes. If a message is pending, the client MUST assert the start of message flag within 1 ms of the start of a message slot and MUST complete delivery within 9 ms of the start a message slot. The Client DTI Version and Path messaging is optional. A Client not supporting it then the field MUST be filled with all zeros. For the Client DTI Version and Path message the minimum transfer rate MUST be greater than or equal to one message every 5 seconds if this option is supported.

If used, the Client DTI Version and Path field MUST be filled as follows:

**Bit 9: Start of Message**

The Client DTI Version and Path field start of message bit MUST be set high for one frame to indicate the beginning of a status message.

**Bit 8: Data Valid Bit**

The Client DTI Version and Path field data valid bit MUST be set to '1' to indicate that the next eight data bits contain valid data. When the Client DTI Version and Path field data valid bit is '0', the client MUST ignore the contents of the eight data bits that follow.

**Bit 7:0: Data Byte**

This data byte MUST contain the serial message data bytes.

**Table 6–6 - Client DTI Version and Path Message Format**

<b>Name</b>	<b>Type (1 Byte)</b>	<b>Length (1 Byte)</b>	<b>Value</b>
DTI Client Version Number	1	1	Most recent issue of the DTI specification that the client supports. Range [0-255]. 0 for I01 & I02 1 for I03
DTI Client IPV4 Address	2	4	IP Address of DTI client.
DTI Client Port Number	3	1	Assigned port number for the physical DTI port range [0-255].
DTI Client IPV6 Address	4	16	IP Address of DTI client.
Client type	5	1	"C" "C" for DTI client not associated with a server; "S" for subtending server.
Number of client ports	6	1	Number of client ports associated with the device. Client port number starts at zero. Range [0-255].
Active client port	7	1	Port number of input signal currently in use by the client. Range [0-255].
EOT	8	1	0x00 The EOT marks the end of the Path Position Message.

**6.4.3.1.6 Reserved**

The remaining 68 bits MUST be reserved for future use. They will be filled with ‘1’'s until defined otherwise.

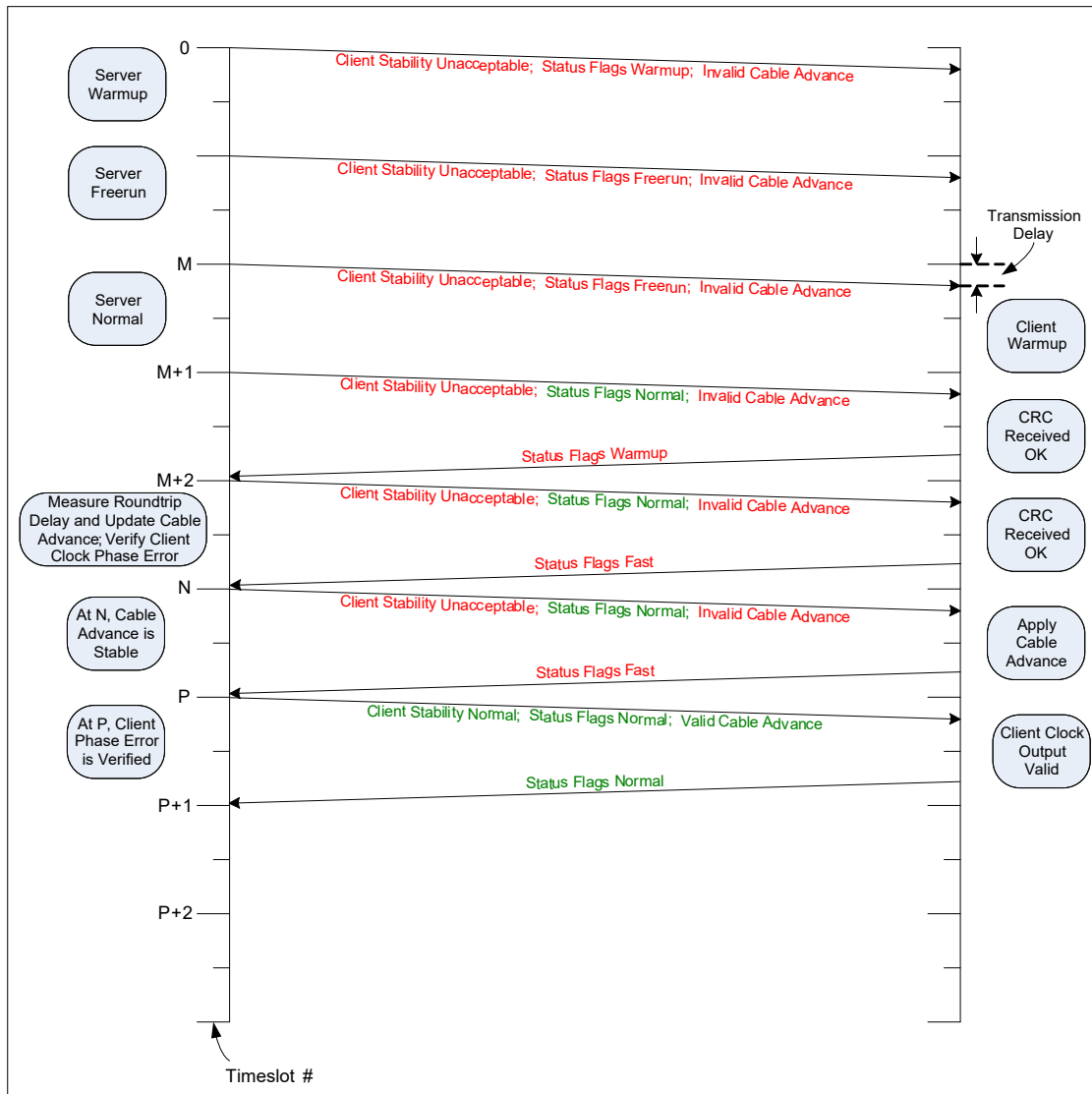
**6.4.3.1.7 CRC 16**

This 16-bit field MUST contain the CRC16 cyclic redundancy check. The CRC-16 algorithm is defined in Annex C, DTI-S CRC-16.

**6.5 DTI Server-Client Protocol Interaction**

The following protocol exchange diagram, Figure 6–3, will be used to illustrate the protocol rule set. Specific requirements relative to server and client operation supporting the protocol are delineated in Section 0.

The exchange starts with both the Server and Client inactive (powered down). The first step is Server warmup. The principle sub-system requiring warmup in a DTI server is the local oscillator. During warmup, a server is transmitting DTI messages. with the warmup state indicated. The server will stay in the warmup until the oscillator is stable and a stable time of day setting is acquired. The Server transitions to the free-run state and will persist in this state until locked to an external reference. Figure 6–3 illustrates the client power-up occurring during timeslot m. The client will not transmit until loss of receive frame condition has cleared (see Section 7.2.3). During timeslot m, the DTI Server enters the normal condition as shown. The next server frame indicates this normal condition in the service status flags. Note that the client can begin locking, using valid server frames as soon as the server is out of warmup, establish a time of day setting, and is actively transmitting. Assuming that the client has established receive framing and it receives the current frame with a valid CRC, it responds with a client frame in timeslot m+1. The client clock is assumed to be in warmup at this time, as reflected in the client status flag lock, set to invalid. At timeslot m+2, the server can begin to process the client frame by measuring round-trip delay, as well as collect the client clock phase error measurement data returned in the client frame.



**Figure 6–3 - Example of DTI Server-Client Protocol**

The server filters the round-trip measurement data to establish a stable cable advance correction that will not degrade the timing performance of the client clock. Figure 6–3 illustrates the Server achieving stable cable advance just prior to timeslot p. At timeslot p, the Server transmit frame indicates this by setting the Cable Advance flag to valid. During the same timeslot, the client clock is assumed to achieve a local phase lock and the client message indicates fast mode. Finally, just prior to timeslot P+1 the server has verified, using the client clock phase error data since timeslot M+1, that the client is in proper phase lock with respect to master clock and reports this at P+1 by setting the client stability condition flag to valid.



## 7 DTI CLIENT AND SERVER OPERATION

This section is normative. See Section 1.1 for an informational description of the DTI protocol.

### 7.1 DTI Server Modes

The DTI Server can support three modes of operation:

1. Free running Master: no external source is provided to control frequency or timestamp accuracy.
2. GPS Traceable: source of timestamp and frequency traceability is GPS.
3. Network Traceable: a standard network PDH, SONET, or SDH interface is used to provide traceability to a Stratum 1 frequency reference.

In M-CMTS, deployment where all the elements are collocated GPS traceability is not required.

The DTI server **MUST** support free-running master clock mode. In this mode, no external sources of traceability are required.

The DTI Server **MAY** support the GPS traceable mode of operation.

The DTI Server **MAY** support the network traceable mode of operation.

#### 7.1.1 Free Running Master Mode

The free-running master clock requirements are related closely to the existing DOCSIS Clocking Requirements, with additional margin included to ensure the DTI client master clock output performance meets all requirements.

To support common tracking of all downstream elements including DTI clients, the free running master mode **MUST** limit output frequency slew rate to less than  $1e-9$  over a ten second period.

The free-running master mode output frequency accuracy **MUST** be better than 1 ppm over a temperature range of 0 to 40 C for up to 10 years from the date of manufacturer.

The DTI server high frequency jitter (above 10 Hz) in free running master mode **MUST** be less than 50 ps RMS.

The DTI server ranging wander when observed through the ranging wander qualification filter<sup>8</sup> **MUST** be less than 250 ps RMS.

DTI server port-to-port performance applies to ports on a single root server or ports between a root server and a subtending server. The DTI server port-to-port ranging wander when observed through the ranging wander qualification filter **MUST** be less than 125 ps RMS.

#### 7.1.2 External Reference Modes

##### 7.1.2.1 Common Synchronization Performance Requirements

**Note:** The requirements in this section do not apply in Free Run Master mode.

###### 7.1.2.1.1 DTI Server Holdover Performance

To maintain compatibility with the T-Service synchronization hierarchy, the DTI Server clock **MUST** operate with holdover performance in one of the categories specified in Table 7-1. These categories are consistent with controlling standards for existing T-services [G.812] and [T1.101].

---

<sup>8</sup> The ranging wander measurement filter is described in Appendix IV.

**Table 7-1 - DTI Server Holdover Requirements**

ITU G.812 Type	ANSI T1.101-1999 Stratum Level	Holdover Aging Allowance (ppb per day) <sup>9</sup>	Holdover Temperature Allowance <sup>10</sup> ppb	24-hour Holdover Total Allocation (aging and temperature) ppb
Type II	Stratum 2	NA	NA	0.1
Type I	NA	0.2	2.0	2.2
Type III	Stratum 3E	1.0	10.0	11.0

#### 7.1.2.1.2 DTI Server Slew Rate Performance

The DTI server needs to constrain the phase slew rate during the 35 seconds of the DOCSIS maximum ranging interval. The DTI Server MUST maintain a phase slew rate of 5 ns over a ten second period during normal and acceptable operation.

#### 7.1.2.1.3 DTI Server Timing Jitter Performance

The DTI server ranging timing wander over a 35 second period with the mean subtracted needs to be allocated as part of the 1 ns RMS requirement for chip timing jitter for synchronous operation as discussed in Appendix III. Two collocated servers supporting two clients and the associated PHY components need to be considered in the overall budget. The following requirements support a 400 ps RMS allocation of the 1 ns RMS requirement to the DTI components.

- The DTI server high frequency jitter (above 10 Hz) in free running master mode MUST be less than 50 ps RMS.
- The DTI server ranging wander, when observed through the ranging wander bandpass filter, MUST be less than 250 ps RMS.
- DTI server port-to-port performance applies to ports on a single root server or ports between a root server and a subtending server. The DTI server port-to-port ranging wander, when observed through the ranging wander qualification filter, MUST be less than 125 ps RMS

#### 7.1.2.2 GPS Traceable

The DTI server is required to support both DOCSIS system timing requirements as well as the synchronization timing requirements of the T-services that may need to be supported. With the modular CMTS architecture, DTI elements, such as upstream receivers, EQAMS, and M-CMTS Cores may be in different nodes or buildings. The DTI servers in different buildings MUST operate with sufficient coherence to support all time and frequency dependent functions particularly ranging, latency management and T-service synchronization when operating in GPS mode.

<sup>9</sup> Represents the average frequency drift caused by aging. This value is derived from typical aging characteristics after 60 days of continuous operation. It is not intended to measure this value on a per day basis, as the temperature effect will dominate.

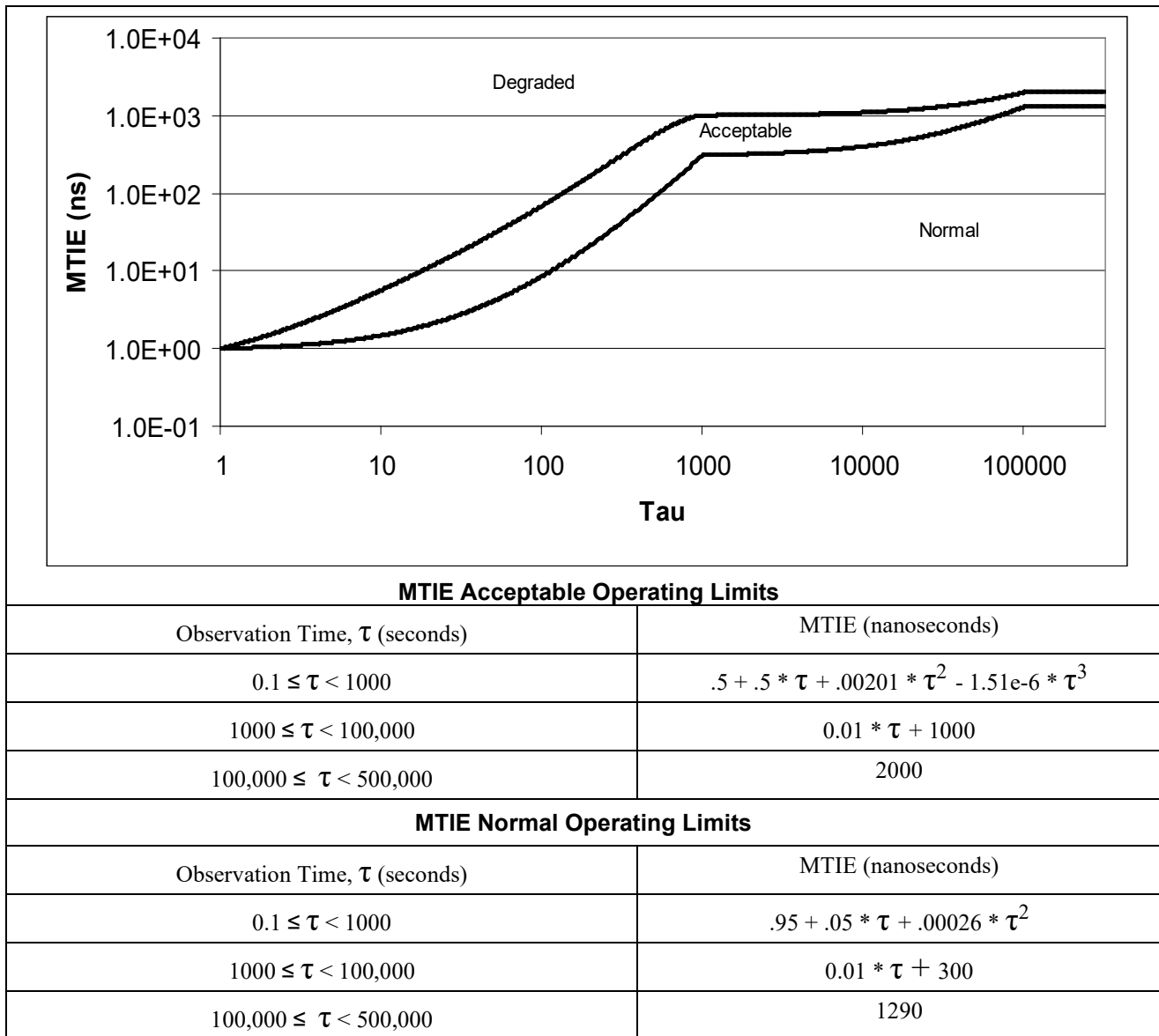
<sup>10</sup> Operating temperature conditions are in the range of 0-40C ambient at a maximum rate of 10C per hour.

The DTI server **MUST** operate within the Maximum Time Interval requirements specified in Figure 7–1 in GPS traceable mode. Figure 7–1 defines three levels of server operation:

1. Normal - this is the performance level anticipated under the normal operating environment. While the server should operate over the full environmental requirements specified by the MSO, it is recommended that the vendor document the environmental limits, such as limited temperature slew associated with normal operation. Normal operation does not include loss of all external references.
2. Acceptable - this is the performance level the Server **MUST** meet over the full environmental requirements specified by the MSO. The server **MUST** meet this level of MTIE over limited outage periods of the input references. It is recommended that the vendor document the outage periods that can be supported.
3. Degraded - The server **MUST NOT** enter the degraded level of operation unless there is an extended duration outage beyond the vendor specified limit or the environmental limits are exceeded.

Additionally, the following operating conditions apply for normal and acceptable service levels:

- The maximum ambient temperature slew rate is less than 10°C per hour.
- If there is more than one valid input a reference switch from one valid input to another is considered normal and acceptable.
- The DTI server is not operating in warmup or free-run.
- The DTI server has been continuously powered for at least 1 hour.



**Figure 7-1 - GPS Traceable Mode MTIE Requirements**

When the GPS traceable DTI server is required to operate with a second GPS traceable server in a different location, the time error of the DTI server master clock with respect to UTC frequency reference<sup>11</sup> over a 35 second time window with the mean subtracted MUST be less than 300 ps RMS under normal operating conditions.

**7.1.2.3 Network Traceable Mode**

The Network Mode of operation utilizes synchronization embedded on PDH, SONET, and SDH physical layer and framing to provide frequency traceability to Stratum 1 network clock. Synchronization traceability can potentially be delivered by:

- A Plesiochronous Digital Network (PDH) DS1 signal that is a traffic-bearing signal. A traffic-bearing signal is either traceability to a stratum 1 clock within the transport network or stratum traceable source

<sup>11</sup> In practice a cesium reference clock or equivalent may be used to approximate a UTC frequency reference.

from the remote terminating equipment. The wander and jitter requirements are less stringent than the synchronization-bearing mode described in the next paragraph. In addition, a traffic-bearing interface may incur transients associated with SONET or SDH pointer adjustment events.

- A PDH signal that is a synchronization-bearing signal. A synchronization-bearing signal is traceable to a stratum 1 clock within a transport network and is required to meet tighter wander and jitter bounds.
- A SONET/SDH signal from a traceable network element. SONET /SDH networks are required to be synchronous and supplied the traceable synchronization from a synchronization port on the terminating equipment. The derived synchronization interface port can support synchronization status message to provide a level of verification.

The Network Mode of operation SHOULD NOT utilize references operating at the traffic-bearing performance limits as specified in [G.824]. Vendors MAY support filtering of traffic bearing signals, but this operation is beyond the scope of this standard.

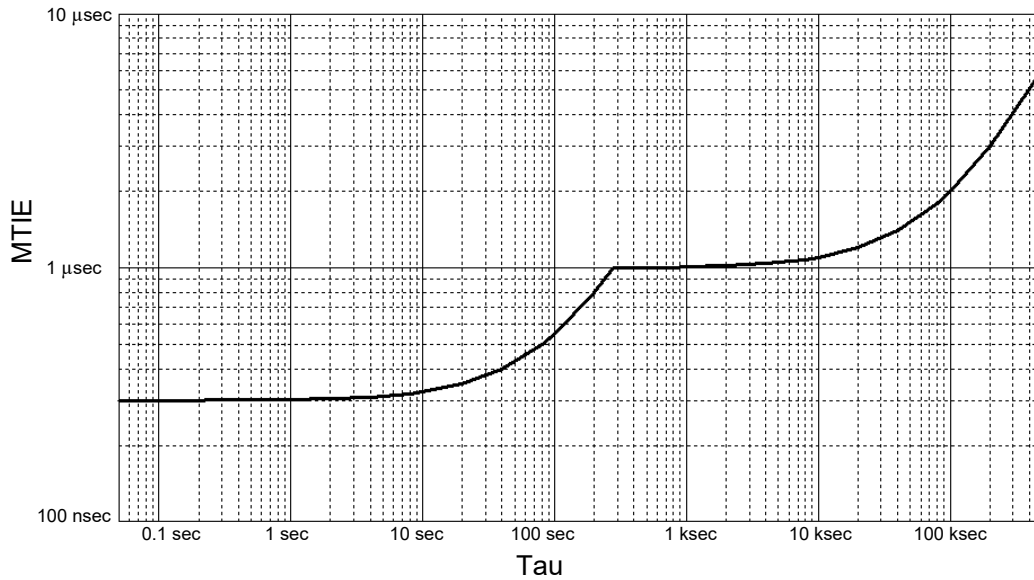
The Network Mode provides precise frequency (rate) synchronization, but not precise time synchronization. When the signal is used with a properly designed filter clock in a DTI server, the output DTI timing signals will support all M-CMTS requirements when all the DOCSIS elements are collocated. The Network Mode (as well as the GPS mode) will also enable delivery of synchronization required for commercial T1 services.

The DTI Server MUST meet the output MTIE and common synchronization requirements under both normal and acceptable input conditions (see 7.1.2.3.1).

#### *7.1.2.3.1 Normal and Acceptable Input Conditions*

Normal and acceptable input conditions are defined as follows:

- The maximum input jitter is less than 3.24 microseconds peak-to-peak from 10 Hz to 40 kHz.
- The maximum input wander is limited to the MTIE mask in Figure 7–2.
- The maximum ambient temperature slew rate is less than 10°C per hour.
- If there is more than one valid input a reference switch from one valid input to another is consider normal and acceptable.
- The DTI server is not operating in warmup or free-run.
- The DTI server has been continuously powered for at least 1 hour.

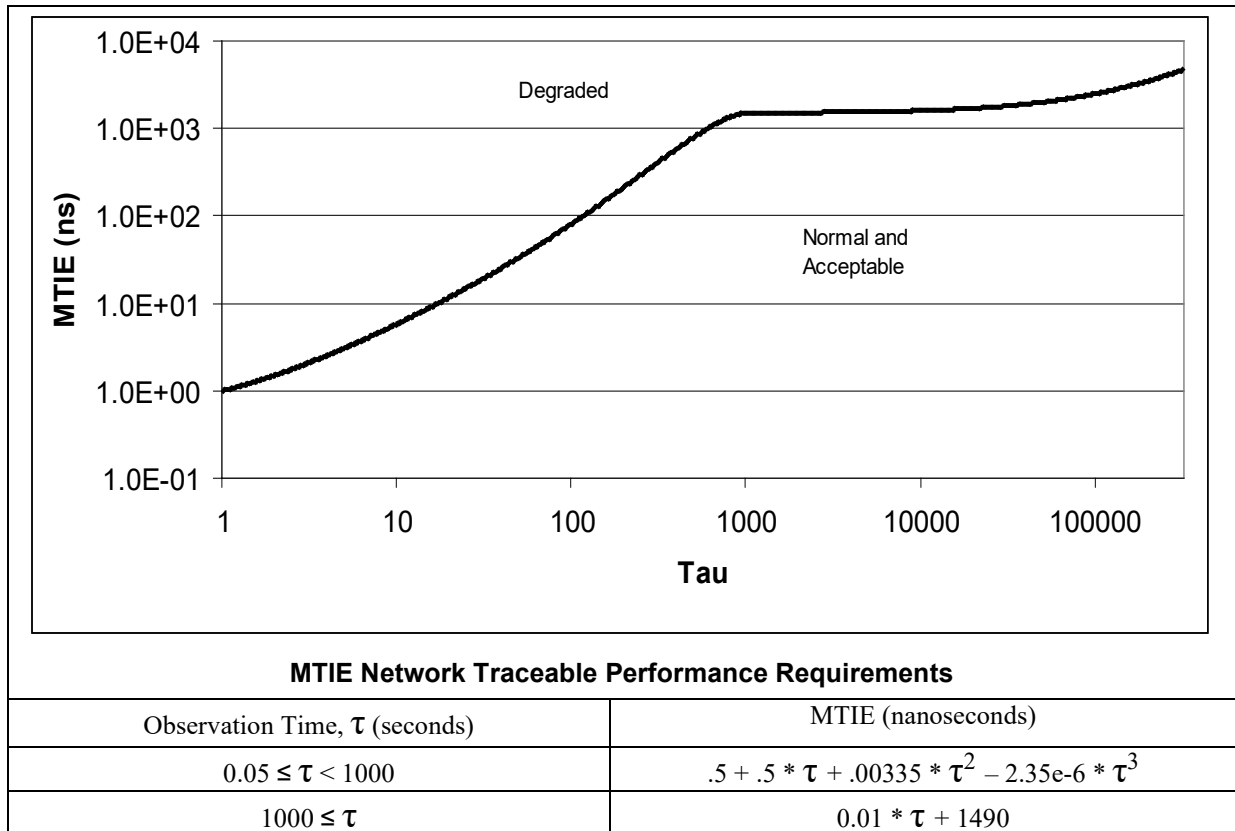


MTIE Synchronization-bearing Input Reference	
Observation Time, $\tau$ (seconds)	MTIE (nanoseconds)
$0.1 \leq \tau < 280$	$2.5 * \tau + 300$
$280 \leq \tau$	$0.01 * \tau + 997$

Figure 7-2 - Network Input MTIE Requirements

7.1.2.3.2 Network Traceable MTIE Performance Requirements

The DTI server MUST meet the normal and acceptable MTIE performance requirements in Figure 7-3 under the input conditions stipulated in Section 7.1.2.3.1.



**Figure 7-3 - Network Traceable Mode MTIE Requirements**

### 7.1.3 DTI Server Functional Requirements

The Server Clock MUST support the clock fast mode, clock free-run mode, clock holdover mode, and clock normal mode, as defined in [T1.101].

During warmup, a server MUST transmit DTI messages with the Client Performance Stable flag and the Cable Advance flag set to false.

The DTI server MUST NOT exit warmup until it has established an initial Time of Day setting.

Until a stable cable advance correction is established, the Server MUST indicate an invalid cable advance in the Server Status Flags.

Under normal operating condition with a properly functioning client clock and a normal transport bit error rate BER ( $< 1 \times 1e-8$ ), the Server MUST establish a stable cable advance within 20 seconds from the first valid client response.

The DTI server MUST support both a manual and automatic mode for cable advance.

The DTI server cable advance mode MUST be settable on a per port basis.

Regardless of the cable advance mode, the DTI server MUST minimize changes in the cable advance value once the stable cable advance flag is asserted to ensure that all DTI server performance requirements are met.

In manual cable advance mode, the cable advance correction value MUST be unchanged unless the user requests an update is requested by the user. Transitioning from automatic mode to manual mode MUST freeze the current cable advance value. Transitioning from manual mode to automatic mode MUST resume automatic adjustment starting from the current cable advance value.

In automatic cable advance mode, the DTI server MUST automatically adjust the cable advance to slow changes in the cable delay associated with normal environmental conditions.

In automatic cable advance mode under stable cable advance operation (server status cable advance flag = 1), the rate of change of the cable advance MUST be no more than one LSB of the fractional cable advance word (26 ps) per second.

The DTI server MAY provide an option to set the cable advance to a user selectable value under manual cable advance mode.

When switching from a working DTI server to a protection DTI server in a common shelf, the DTI output signal MUST be active within 500 ms.

Under normal operating condition with a properly functioning client clock a normal transport BER ( $<1 \times 1e-8$ ), a protection DTI Server that has just become active MUST establish a stable cable advance within 1000 ms from the first valid client response.

### 7.1.4 DTI Server Test Signal Mode

The DTI Server MUST support a test mode for each output port of the Server. The test signal MUST be a continuous stream of all ones prior to Manchester encoding. The measurement of phase bias requires accounting for the group delay in any common path elements between the output port connector and the point where the transmit and receive paths split within the server. The delay bias measurement should be referenced to this transmit receive split point.

The DTI Server MUST support a 10.24 MHz master clock test port. This port MUST be phase accurate with less than 2500 ps bias when compared to a delay calibrated DTI Client with correction for test port group delay. The master clock test port output jitter MUST be less than 25 ps RMS.

The DTI Server MUST support a 10 kHz master frame clock test port. This port MUST be phase accurate with less than 2500 ps bias when compared to a delay calibrated DTI Client with correction for test port group delay. The master frame clock test port output jitter MUST be less than 50 ps RMS.

## 7.2 DTI Client Operation

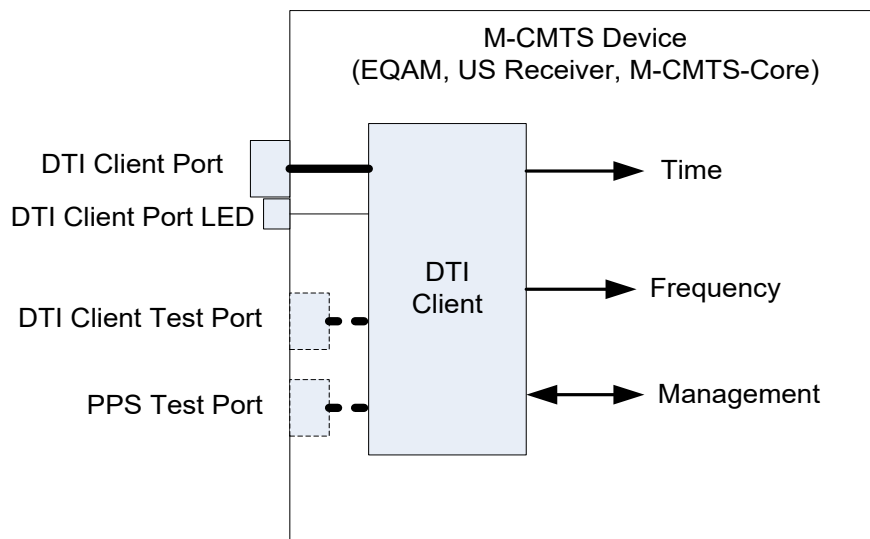


Figure 7-4 - DTI Client System Interface



The DTI Client is located within a host M-CMTS Device, such as an EQAM, US receiver, or M-CMTS Core. The function of the DTI Client is to interface with the DTI Server using the DTI Protocol, and to provide Time, Frequency, and Management interfaces to the M-CMTS Device.

Since the DTI Client is likely to be integrated into the M-CMTS Device, the internal Time, Frequency, and Management Interfaces may be proprietary, and as such are not defined by this specification. The test port requirements specified herein are intended to verify the performance of the corresponding operational signals.

DTI clients residing in EQAM, US Receiver, and M-CMTS Core Devices:

- MUST provide a DTI Client Port.
- MUST provide a DTI Status LED.
- MUST employ a one-sided 3 dB loop filter bandwidth in the range of 1-10 Hz to track the DTI Server timing.
- MUST support a pull-in range capable of locking to a properly functioning server over normal operating conditions, as specified in Section 7.2.1, for a period of 10 years from date of manufacturer.

DTI clients residing in US Receiver and M-CMTS Core Devices MUST provide a DTI Client Test Port.

DTI clients residing in EQAM M-CMTS Devices MAY provide a DTI Client Test Port.

DTI clients residing in M-CMTS EQAM devices that do not provide a DTI Client Test Port MUST provide a PPS Test Port and support verification of DTI Client operation as per Section 5.

Since the DTI Client Test Port and the EQAM DTI Test Port are not required for normal operation, it may be necessary to remove panels or covers on the M-CMTS Device in order to access these ports.

**7.2.1 DTI Normal operating conditions**

The DTI client MUST meet all specifications when under Normal Operating Conditions defined as:

- The ambient temperature range is between 0 to 40 C.
- The maximum ambient temperature slew rate is less than 10°C per hour.
- The DTI Transport BER is less than 1e-8.
- The cable length is between zero and 200 meters and is operating in compliance with Section 5.

**7.2.2 DTI Client Operational Modes**

The DTI Client MUST support and report the operational modes described in Figure 7-2.

**Table 7-2 - Client Operating States**

Mode	Description
WARMUP	Oscillator has not yet stabilized.
FREE-RUN	Client has not had a valid timing source since reset, or has had to abort acquisition.
FAST	Client is using a short acquisition time constant so as to reduce the initial lock time.
NORMAL	Clock is stable, has locked on to the timing reference, and is fully compliant.
BRIDGING	Client has lost its timing reference but is maintaining acceptable performance.
HOLDOVER	Client has lost its timing reference but is attempting to maintain last valid frequency.

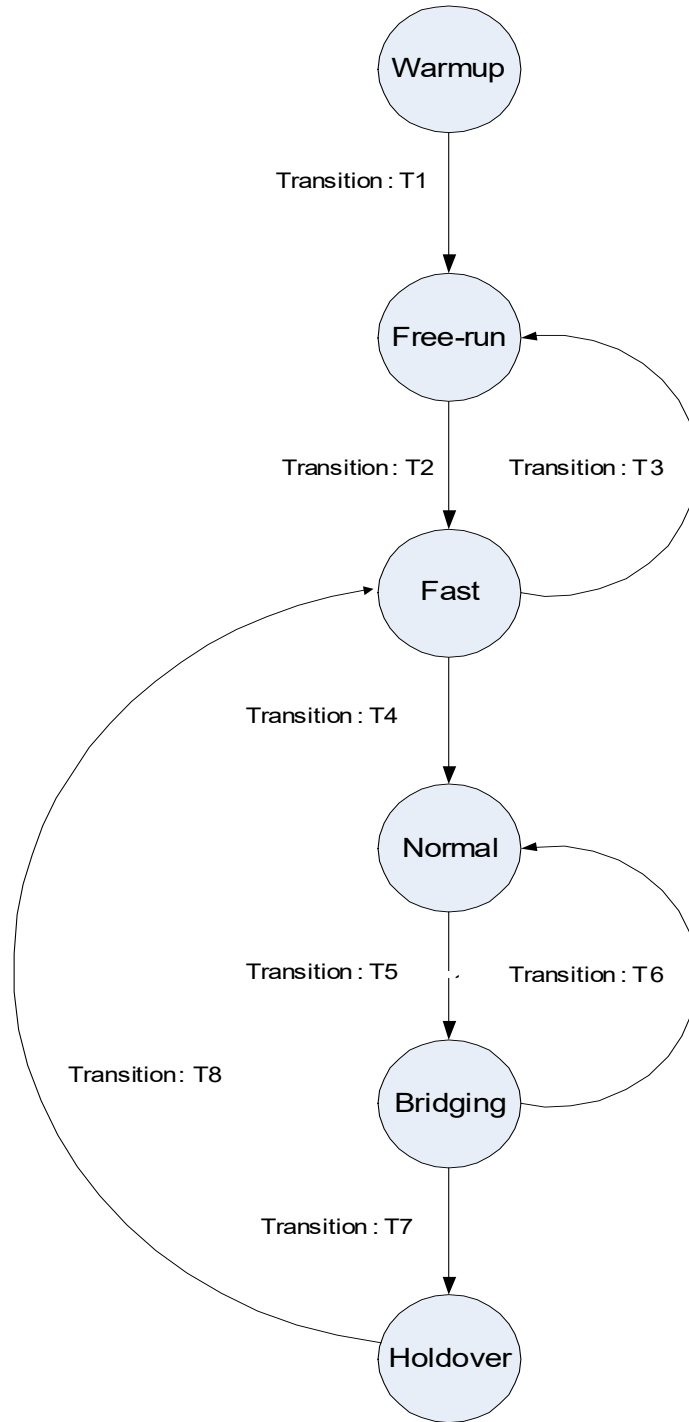
### 7.2.3 DTI Client Mode Transition Diagram

The DTI client MUST implement mode transition similar to those described in Table 7–3 and shown in Figure 7–5.

With a valid Server connected, and under normal operating conditions, the DTI Client MUST transition from FREE-RUN to NORMAL within a period of 20 seconds.

**Table 7–3 - Client Mode Transitions**

Transition	From	To	Condition for Transition	Comment
T1	WARMUP	FREE-RUN	Vendor specific Timeout. (Timeout less than 20 ms)	Allows sufficient time for oscillator to stabilize.
T2	FREE-RUN	FAST	FER $\leq$ 0.02 over 50 ms AND Server Status Flag bit 0 $\neq$ 1 (warm-up)	Sufficient number of valid frames from Server
T3	FAST	FREE-RUN	FER $\geq$ 0.05 over 50 ms OR Server Status Flag bit 0 = 1 (warm-up)	Acquisition aborted.
T4	FAST	NORMAL	FER $\leq$ 0.02 over 50 ms AND Server Status Flag bit 5 = 1 (cable advance) AND Server Status Flag bit 6 = 1 (client performance)	Acquisition complete.
T5	NORMAL	BRIDGING	FER $\geq$ 0.05 over 50 ms OR Server Status Flag bit 0 = 1 (warm-up) OR Server Status Flag bit 5 $\neq$ 1 (cable advance) OR Server Status Flag bit 6 $\neq$ 1 (client performance)	Insufficient number of valid frames from Server
T6	BRIDGING	NORMAL	FER $\leq$ 0.02 over 50 ms AND Server Status Flag bit 5 = 1 (cable advance) AND Server Status Flag bit 6 = 1 (client performance) AND Server Status Flag bit 0 $\neq$ 1 (warm-up)	
T7	BRIDGING	HOLDOVER	2 second Timeout.	
T8	HOLDOVER	FAST	FER $\leq$ 0.02 over 50 ms	Sufficient number of valid frames from Server



**Figure 7-5 - Client Mode Transition Diagram**

## 7.2.4 Functional Requirements

The client MUST NOT use the DTI frames as a timing reference while in the warm-up, free run, or hold-over modes. The client MUST NOT transmit unless the most recent CRC-16 received from the server is valid. This guarantees that transmission does not occur during periods of high frame error rates.

In order to support the cable advance ranging measurement, the client frame MUST have less than 15 ns peak-to-peak jitter when in normal DTI client mode.

## 7.2.5 DTI Client Port

The DTI Client Port MUST be an RJ45 Female connector with pinout as described in Table 7-4.

## 7.2.6 DTI Client Test Port

The DTI Client Test Port MUST provide the following signals:

- 10.24 MHz clock (100 ohm differential LVDS)
- 10 kHz frame clock (50 ohm LVTTTL)
- DTI frame serialized data (50 ohm LVTTTL)

The DTI Client Test Port MUST use a standard seven pin Serial-ATA header such as the Molex SD-67800-005 with the following pin assignments.

**Table 7-4 - DTI Client Test Port**

Pin	Signal
1	GND
2	10.24 MHz +
3	10.24 MHz -
4	GND
5	10 kHz frame clock
6	Serialized Data (client only)
7	GND

The 10.24 MHz clock will be a duplicate version of the master clock provided by the DTI client. Great care should be taken to minimize its delay and maximize its fidelity. This clock will be used for jitter, phase alignment and wander measurements

## 7.2.7 DTI Client Test Port Clock

The DTI Client Test Port Clock MUST meet the double sideband phase noise requirements shown in Table 7-5 over the specified frequency ranges. The DTI client clock MAY provide enhanced phase noise performance as described in Annex A.

**Table 7–5 - Client Phase Noise**

	<b>Double Sideband Phase Noise Requirements</b>	<b>Jitter</b>
10 Hz to 100 Hz	< -53 dBc	<0.035 ns RMS
100 Hz to 1 kHz	< -61 dBc	<0.014 ns RMS
1 kHz to 10 kHz	< -53 dBc	<0.035 ns RMS
10 kHz to 5.12 MHz	< -53 dBc	<0.035 ns RMS

If the DTI Client is in the NORMAL or BRIDGING mode, the DTI Client Test Port Clock MUST exhibit wander below 10 Hz with a standard deviation of less than 270 ps relative to a properly functioning root DTI server 10.24 MHz test port clock.

If the DTI Client is in the NORMAL mode then the DTI Client Test Port Clock MUST maintain absolute phase alignment of  $\pm 5$  ns with a fixed offset less than  $\pm 50$  ns specified by the vendor. The measurement is made with respect to a properly functioning root DTI server 10.24 MHz test port clock over the full 0 to 200 meter cable length.

**7.2.7.1 DTI Client Test Port Data**

The DTI Client Test Port Data is a delayed version of the 5.12 Mbps DTI frame data. The data present on the line during the previous frame N-1 is buffered and output by the Client during frame N after the client checks the CRC received from the server in frame N-1. If the CRC check on frame N-1 passes, the entire 512-bit DTI frame N-1 (server preamble, server payload, server CRC, TGT1, client preamble, client payload, client CRC, and TGT2, as illustrated in Figure 6–2 (c) and detailed in Section 6.4) MUST be output serially on the Client Test Port during frame N. Also when the CRC check passes, the Client Test Port MUST output 22 zeros during each turnaround guard time TGT1 and TGT2. If the check on the CRC received from the server in frame N-1 fails, the Client Test Port MUST output a dummy frame consisting of 512 ones during frame N. In any case, the Client Test Port Data output is always a continuous stream of data at a 5.12 Mbps rate. The data MUST be clocked out off a half rate of the 10.24 MHz clock where each bit corresponds to two 10.24 MHz clock cycles and the test port frame clock identifies both the start of the frame data, and the data bit alignment with respect to the 10.24 MHz clock. The alignment of the frame clock and test data is described in greater detail in the next section.

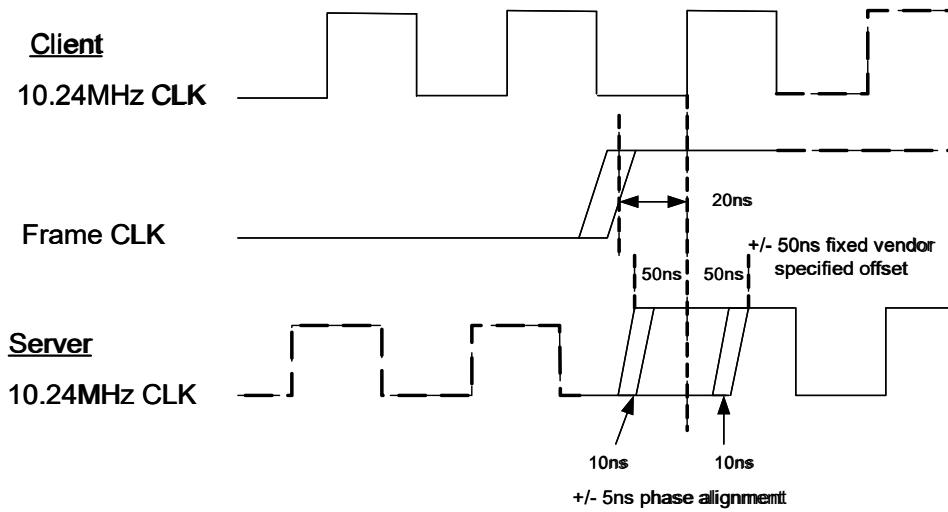
**7.2.7.2 DTI Test Port Frame Clock**

The 10 kHz frame clock is used to frame both the serialized data and the 10.24 MHz clock edge for phase alignment measurements. Its rising edge serves a dual purpose: it identifies the first bit of the server-to-client frame preamble, and it identifies the 10.24 MHz ATP clock cycle edge that is phase aligned to the DTI server ATP output. The frame clock falling edge identifies the first bit of the client-to-server frame preamble.

Both the frame clock and serialized data MUST maintain a minimum of 20 nS setup time and a minimum of 0 ns hold time with respect to the 10.24 MHz clock.

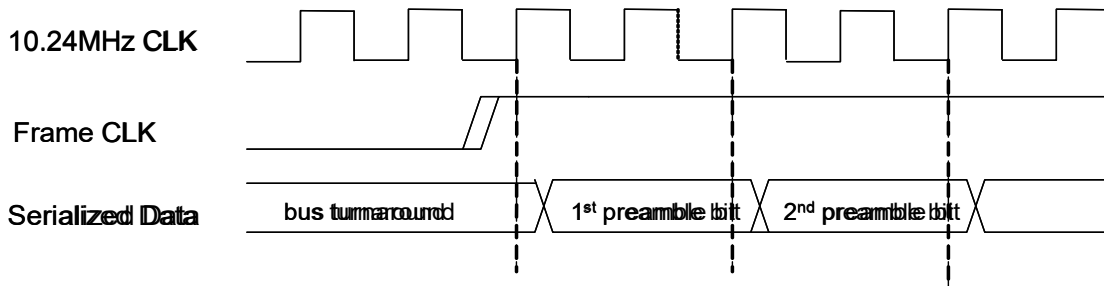
A timing diagram of the frame clock and its relation to the 10.24 MHz clock and the serialized data is shown below.

### 10.24MHz Phase Alignment



### Serialized Data

#### Server to Client



#### Client to Server

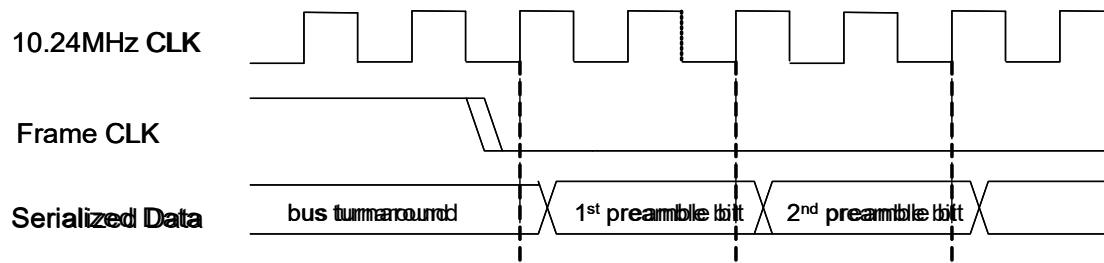


Figure 7-6 - Test Port Timing Diagram

### 7.2.8 Alternative EQAM DTI Testing

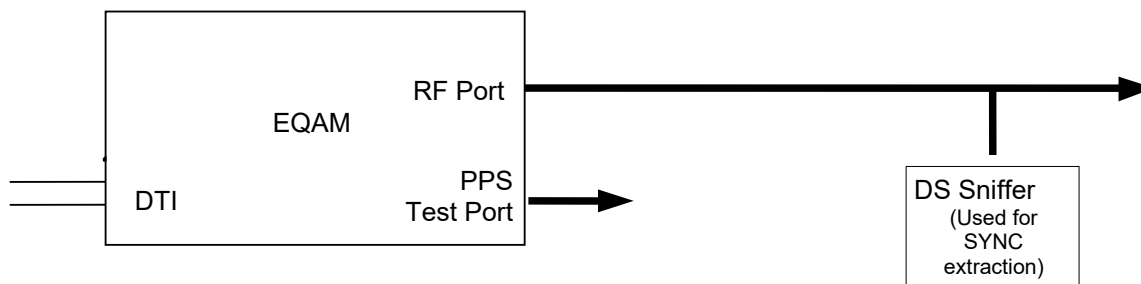
EQAMs MAY use their RF port in conjunction with a PPS Test Port instead of their DTI client test port for DTI ATP testing. The downstream symbol clock integrated phase noise measurements are measured directly on the RF port in lieu of DTI master clock jitter measurements. The SYNC messages are observed on the RF port to check DTS synchronization. A PPS Test Port connector MUST be available for compliance test purposes.

The PPS Test Port connector MAY be normally externally accessible. In this case the PPS Test Port connector MUST be a female BNC 50ohm type connector.

The PPS Test Port connector MAY not be normally externally accessible and will only be provided for compliance test purposes. In this case the PPS Test Port connector MUST be a cable mounted male BNC 50ohm type connector and the cable which connects the PPS Test Port connector to the DTI Client MUST be less than 1 meter in length.

The method of coupling this cable to the DTI Client is left to the discretion of the vendor.

Figure 7-7 shows this alternative test arrangement.



**Figure 7-7 - Edge QAM test port option**

The PPS Test Port serves a dual purpose: it identifies the gpssec time mark, and provides a phase timing reference to facilitate the measurement of two-way DTI ranging phase alignment with respect to a DTI server. If the DTI Client is in the NORMAL mode then the active edge of the PPS MUST maintain absolute phase alignment with respect to the Server master clock test port output of  $\pm 5$  ns peak, with a vendor-specified fixed offset of up to  $\pm 50$  ns.

### 7.2.9 DTI Status LEDs

The DTI Status LEDs MUST be either a single Green/Yellow Bi-Color LED or a set of two LEDs, one Green and one Yellow.

The DTI Status LEDs MUST be externally viewable and be integrated into, or in close proximity to, the DTI Client Port Connector.

The DTI Status LEDs MUST reflect the status of the DTI Client as described in Table 7-6.

**Table 7-6 - DTI Status LEDs**

DTI Status LEDs	DTI Client Mode
Off	WARMUP, FREE-RUN or HOLDOVER
Yellow	FAST
Green	NORMAL or BRIDGING

### 7.3 DTI Distribution Fallback Strategies

Figure 7–8 shows several levels of protection that can be provisioned to establish various degrees of operational confidence in the timing distribution network.

The DTI server provides the point-to-point connections to the DTI clients. Each DTI link (a connection from server to client) is sourced from the server via a passive backplane where the physical connections are made. The server can be provisioned to provide backup for each of these connections via server-based protection cards. The protection is shown on Shelf A, but not on Shelf B. With protection cards, should an active card fail, the associated protection card provides seamless continuous function.

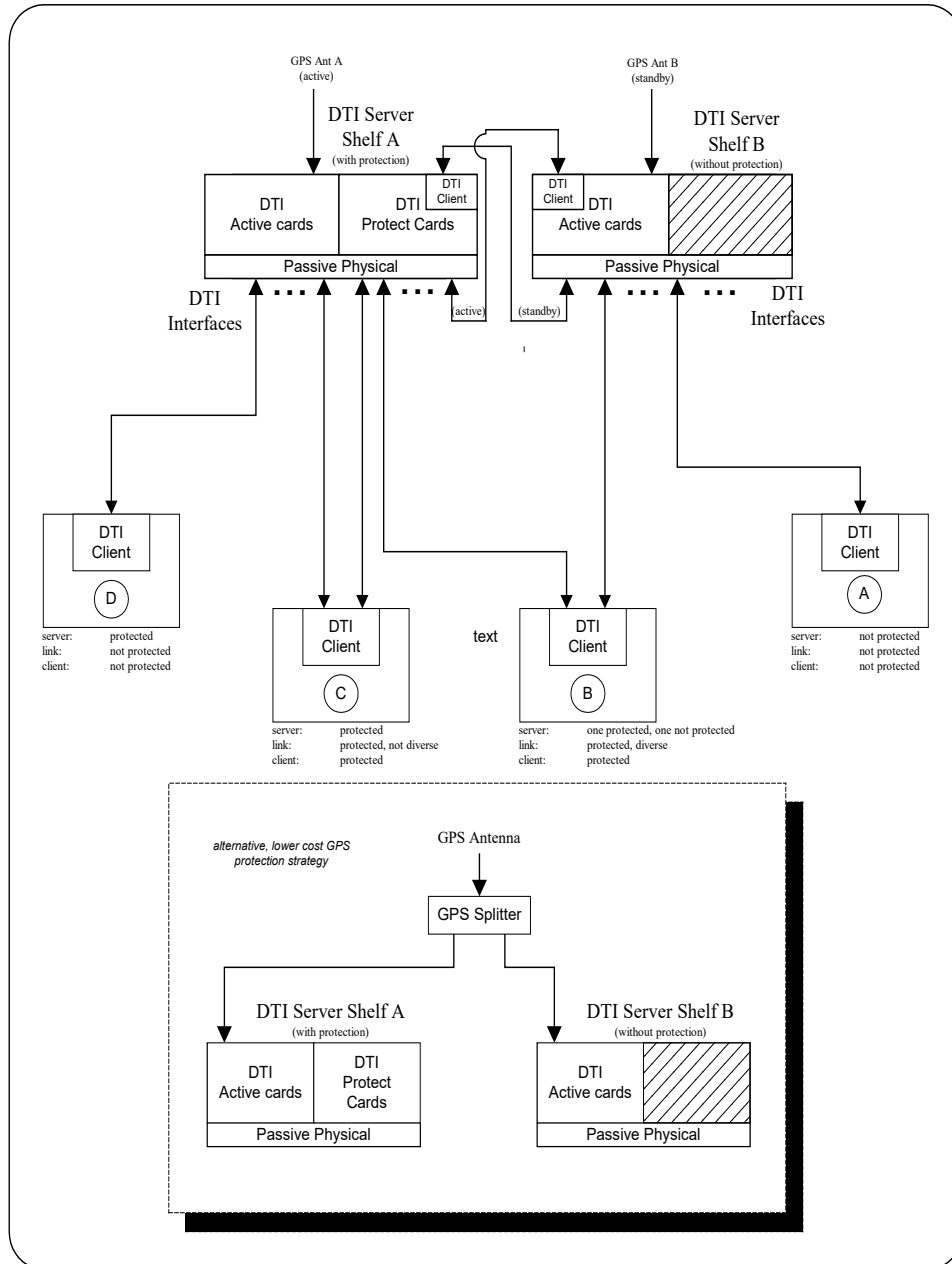


Figure 7–8 - DTI Distribution Fallback Strategies



The figure also shows several levels of fallback provisioning at the DTI client:

- DTI client A illustrates minimal fallback. A non-protected server has a single connection to a client.
- DTI client D illustrates a more robust fallback implementation. As with client A, client D accepts a single DTI connection, but in this case a protected server is providing it.
- DTI client C illustrates a more robust fallback implementation. Here the client accepts 2 DTI connections, allowing for normal operation in the event of a failure on either link.
- DTI client B illustrates a more robust fallback implementation. Client B has the same 2 DTI connections as client C, but in this implementation the links themselves are from diverse servers, which reduces the likelihood of both links failing (from the server side).

Multiple DTI server interconnections are also shown in Figure 7–8. To ensure best coherence (overall alignment of all timing within the timing distribution network) the preferred method is to have a single reference driving the network.<sup>12</sup> This reference is shown in the figure as GPS Antenna A, which is actively being used by DTI server shelf A. Note that a DTI link is shown as active between server shelf A and server shelf B. Server shelf B uses a standard DTI client interface as its reference (much like any client would).

In normal operation, server shelf B is essentially "slaved" to server shelf A. In the event of a fallback on server shelf A, such as a loss of GPS via antenna A, server shelf B can become the driving shelf, using GPS antenna B (which would now become active). Server shelves A and B would reverse roles so that A would now be driven from B via the DTI link from shelf B (labeled "standby") to the DTI client shown in server shelf A.

Regarding GPS fallback in general, the figure shows each of the server shelves connected to GPS from a different antenna, which provides fallback protection in the event of an individual antenna (or its cabling) becoming defective. The bottom of the figure shows an alternative, lower-cost approach where a single GPS antenna is shared by the server shelves via a splitter. Of course, the antenna and cable redundancy is lost, but this method still provides GPS-quality reference availability to the timing network in the event of a single DTI server shelf failure.

---

<sup>12</sup> Alternatively when there is no need for external traceability, the root DTI server can be free running with no external references.

## Annex A Ranging Wander Qualification Filter

This annex specifies an ATP test fixture or method that permits the measurement of server performance as it affects the wander between two DTI clients, due to the clients' tracking of the server. Unlike a CMTS where the upstream receiver and downstream EQAM share a common chassis and implicitly negligible wander associated with the master clock, a modular system introduces distributed components and requires an allocation of ranging wander to address practical limits. The controlling specification from [RFI2] is cited for reference.

### A.1 Chip Timing Jitter for Synchronous Operation

For S-CDMA mode, upstream chip clock timing error (with the mean error subtracted out) relative to the CMTS master clock needs to be less than 0.005 RMS of the chip period over a 35-second measurement interval.

Note that  $0.005 \text{ chip}/5.12 \text{ Mcps} = 1.0 \text{ ns RMS}$ . The following figure illustrates this DOCSIS system timing specification.

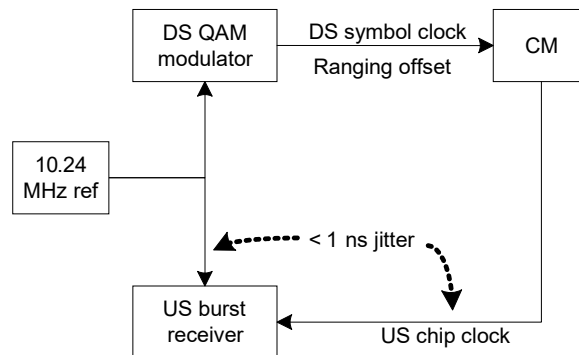


Figure A-1 - Current non-modular CMTS implementation

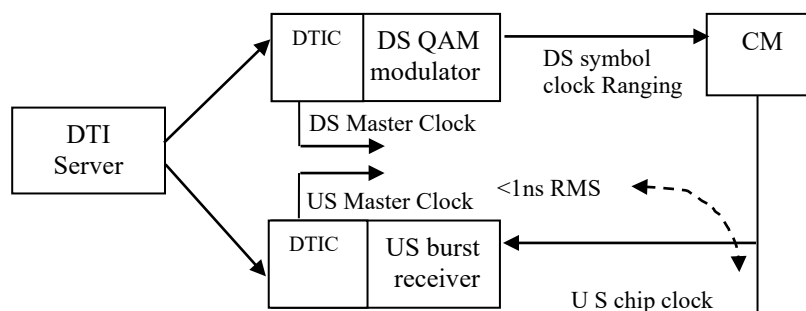


Figure A-2 - New modular CMTS implementation

As can be seen in the second figure delay variation between the DS QAM version of the master clock and the US burst receiver version of the master clock will contribute to the 1 ns RMS overall system requirement. The proposed new budget is based on minimizing this effect while maintain practical cost constraints. The 1 ns RMS budget is partitioned into a 916 ps RMS component allocated to the existing CM ranging process and a 400 ps RMS allocation to the DTI related effects. The DTI allocation is further budgeted as shown:

1. DTI Server Port : 250 ps RMS ranging wander
2. Subtending Server Port additional wander: 125 ps RMS
3. Client Clock local oscillator: 100 ps RMS additional wander

Although a part of the wander will be common-mode between the two clients, the worse case boundary is to assume that each client clock will have sufficient variation in loop parameters (bandwidth and damping factor) so that wander is uncorrelated between the two clients. Vendor implementation and aging of components (varactor gain) may contribute to this effect.

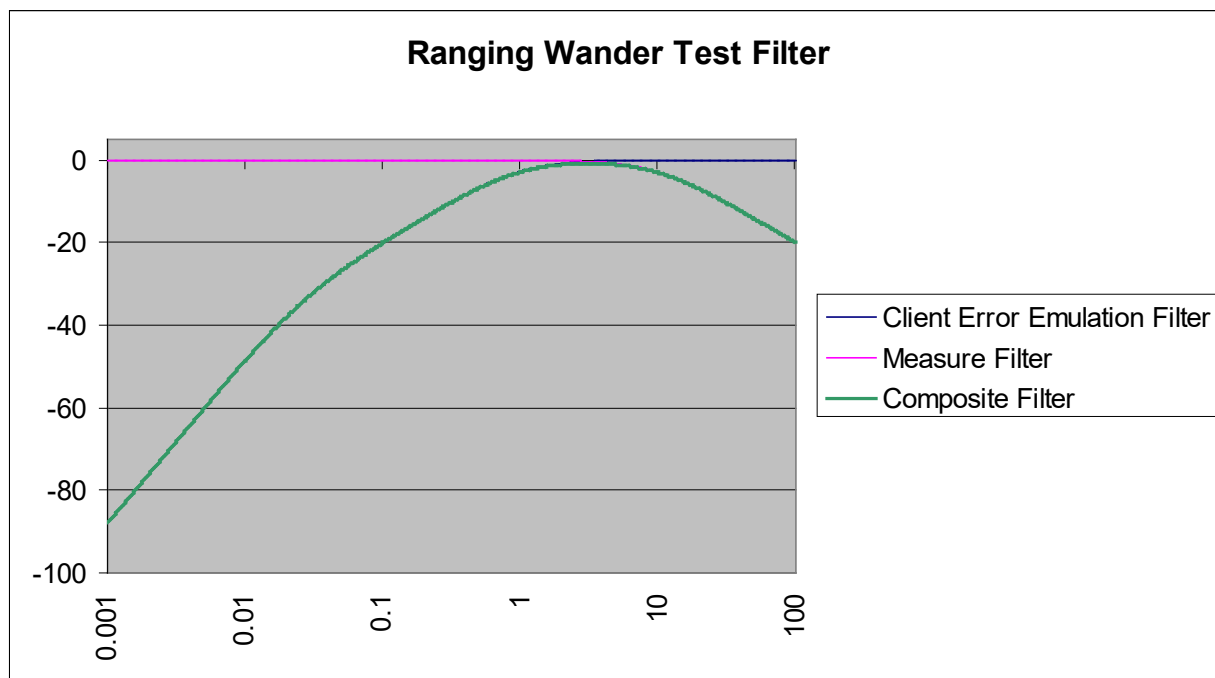
To determine the level of ranging wander noise introduced by a DTI server, the critical issue is the degree to which a client clock can track the server. The model of the client tracking suggests a bandpass filter:

1. A low pass measurement filter to capture the capability of the client to filter high frequency server noise.
2. A high pass error signal filter to capture the client ability to track the server low frequency delay variations.

The overall ranging filter is a composite of these two sections generating a bandpass filter.

The jitter filter is bounded by the minimal jitter filtering capability of a client. The maximum bandwidth is 10 Hz and the assume filter is single pole (20 dB/decade).

The tracking filter is bounded by the minimum 1 Hz bandwidth allowance and assumes a damping factor of 3 and a type II PLL. This high pass filter will attenuate low frequency components with a 40 dB/decade roll off. The combination of these two filters performance can be seen below:



**Figure A-3 - Ranging Wander Test Filter**

The ranging wander qualification filter R(S) is defined to be:

$$R(S) = E(S)M(S)$$

Where E(S) is the high pass tracking filter and M(S) is the low pass jitter filter.

$$E(S) = \frac{S^2}{(S^2 + 5.934 S + 0.9784)}$$

$$M(S) = \frac{1}{1 + (1/(2\pi * 10 \text{ Hz})S)}$$

In the special case where the DOCSIS EQAM and upstream receivers are not collocated, there is an additional allocation of 300 ps RMS for each root server to address the phase variation over a 35 second maximum ranging interval. Stated another way, each root server is allowed a 300 ps RMS wander with respect to UTC frequency (with the mean removed). The DTI servers will be required to be traceable to GPS in both locations to obtain this level of phase coherency.

## Annex B DTI-MIB (normative)

```

DTI-MIB DEFINITIONS ::= BEGIN
IMPORTS
    MODULE-IDENTITY,
        Integer32,
        Unsigned32,
        Counter32,
    OBJECT-TYPE          FROM SNMPv2-SMI
    TruthValue,
    TEXTUAL-CONVENTION  FROM SNMPv2-TC
    OBJECT-GROUP,
    MODULE-COMPLIANCE   FROM SNMPv2-CONF
    InetAddressType,
    InetAddress          FROM INET-ADDRESS-MIB
    ifIndex              FROM IF-MIB
    PhysicalIndex        FROM ENTITY-MIB
    EntitySensorValue,
    EntitySensorDataScale FROM ENTITY-SENSOR-MIB
    clabProjDocsis      FROM CLAB-DEF-MIB
;

dtiMib MODULE-IDENTITY
    LAST-UPDATED "200812090000Z" -- December 9, 2008
    ORGANIZATION "CableLabs Cable Television Laboratories, Inc"
    CONTACT-INFO
        "Postal: Cable Television Laboratories, Inc
         858 Coal Creek Circle
         Louisville, CO 80027
         U.S.A.
        Phone: +1 303-661-9100
        Fax:   +1 303-661-9199
        E-mail: mibs@cablelabs.com"
    DESCRIPTION
        "This MIB module provides the management objects necessary
        to configure and manage the DOCSIS Timing Interface
        devices.
        Copyright 2005-2008 Cable Television Laboratories, Inc.
        All rights reserved."
    REVISION "200812090000Z" -- December 9, 2008
    DESCRIPTION
        "Revised Version includes ECN DTI-N-08.0698-2
        and is published as CM-SP-DTI-I05-081209."
    REVISION "200712060000Z" -- December 6, 2007
    DESCRIPTION
        "Revised Version includes ECN M-OSSI-N-07.0562-5
        and published as I07."
    REVISION "200708030000Z"
    DESCRIPTION
        "Revised Version includes ECN M-OSSI-N-07.0451-3
        and published as I06."
    REVISION "200606280000Z" -- June 28, 2006
    DESCRIPTION
        "Revised Version
        includes ECN M-OSSI-N-06.0278-4."
    REVISION "200508050000Z" -- August 5, 2005
    DESCRIPTION
        "Initial version of the DTI MIB module.
        This revision is published as part of the CableLabs
        M-CMTS specification."
 ::= { clabProjDocsis 7 }

```

```
-- Textual Conventions

DtiCableAdvance ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "2d-1d"
    STATUS      current
    DESCRIPTION
        "The early time adjustment the Client applies to the
        DTI timing frame to mitigate the propagation delay
        between the DTI server and client.  The cable advance
        value is calculated by the DTI server and sent to the
        DTI client.

        The DTI cable Advance is a 24-bit field.  The first two
        bytes represent the integer value of the 149.8 MHz sample
        clock cycles that measure the Cable advance value; the
        third octet represents the remaining fraction of the cable
        advance value in units of the 256th of the 149.8 MHz
        clock cycle."
    SYNTAX      OCTET STRING (SIZE (3))

-- Administrative assignments
dtiNotifications      OBJECT IDENTIFIER ::= { dtiMib 0 }
dtiMibObjects         OBJECT IDENTIFIER ::= { dtiMib 1 }
dtiProtocolObjects    OBJECT IDENTIFIER ::= { dtiMibObjects 1 }
dtiServerObjects      OBJECT IDENTIFIER ::= { dtiMibObjects 2 }
dtiClientObjects      OBJECT IDENTIFIER ::= { dtiMibObjects 3 }
dtiServerProperties    OBJECT IDENTIFIER ::= { dtiServerObjects 1 }
dtiServerGlobalParameters OBJECT IDENTIFIER ::= { dtiServerObjects 2 }

--
-- The following group describes the objects that apply to
-- both DTI Server and DTI Client
--

dtiProtocolTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF DtiProtocolEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This table describes the DTI protocol related information
        contained in Server and Client Frames of a DTI
        client-server pair."
    ::= { dtiProtocolObjects 1 }

dtiProtocolEntry OBJECT-TYPE
    SYNTAX      DtiProtocolEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A conceptual row entry in the DTI Protocol table.
        A DTI server has an entry for each client interface.
        A DTI client has an entry for each server connection.
        A DTI server (proxy) is a client of the root server DTI
        input."
    INDEX { ifIndex }
    ::= { dtiProtocolTable 1 }

DtiProtocolEntry ::= SEQUENCE {
    dtiProtocolEntityType      INTEGER,
    dtiProtocolClientClockType INTEGER,
    dtiProtocolServerStatusFlag INTEGER,
    dtiProtocolClientStatusFlag INTEGER,
```

```

dtiProtocolServerToDState      INTEGER,
dtiProtocolServerToDType      INTEGER,
dtiProtocolServerToDValue     OCTET STRING,
dtiProtocolServerCableAdvanceFlag  INTEGER,
dtiProtocolServerCableAdvanceValue DtiCableAdvance,
dtiProtocolClientPhaseError   Integer32,
dtiProtocolClientVersion      Unsigned32,
dtiProtocolClientPathTraceability Unsigned32,
dtiProtocolServerClientStableFlag  INTEGER
}

```

dtiProtocolEntityType OBJECT-TYPE

```

SYNTAX      INTEGER {
    root(1),
    server(2),
    client(3)
}
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "The type of DTI interface of this entry in the managed
    entity.
    'root'
        indicates the interface generates DTI Server
        Frames and processes DTI Client Frames from a proxy
        server or a client.
    'server'
        indicates the interface generates DTI Server Frames
        and processes DTI Client Frames from a DTI client.
    'client'
        indicates the interface processes DTI Server Frames and
        generates DTI Client Frames."
 ::= { dtiProtocolEntry 1 }

```

dtiProtocolClientClockType OBJECT-TYPE

```

SYNTAX      INTEGER {
    ituI(1),
    ituII(2),
    ituIII(3),
    st3(4),
    dtiClock(5)
}
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "The clock Type of the client as reported in DTI client
    Frames."
 ::= { dtiProtocolEntry 2 }

```

dtiProtocolServerStatusFlag OBJECT-TYPE

```

SYNTAX      INTEGER {
    unknown(0),
    warmup(1),
    freerun(2),
    fastTrackingMode(3),
    normalMode(4),
    holdoverMode(5),
    clientStable(6), -- deprecated
    testMode(7)
}
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION

```

"The current Operational Status of the server side of the DTI interface. This MIB object should be updated no more than once per second. The Server reports the status of the DTI interface associated to this entry as follows:

```
'warmup'
    The server clock has not yet stabilized and has not
    acquired a stable Time of Day.
'freerun'
    The server has a stable clock and Time of Day but
    is not locked to an external clock reference.
'fastTrackingMode'
    The server is in the process of locking to the
    timing reference.
'normalMode'
    The server is locked to the timing reference.
'holdover'
    The server has lost the timing reference and is in
    holdover mode.
'clientStable'
    The server has verified that the Client phase error
    is acceptable for this DTI interface entry. This
    value is deprecated as does not indicate the status
    of the server side of the DTI interface. The performance
    stable client is reported by
    dtiProtocolServerClientStableFlag.

'testMode'
    The server has entered in Test Mode for this
    DTI entry interface due a set to 'true' of
    dtiProtocolControlTestMode or because is receiving
    test mode frames from the DTI client remote end."
```

REFERENCE

"DOCSIS Timing Interface Specification, DTI Server-Client Protocol Interaction."

```
::= { dtiProtocolEntry 3 }
```

dtiProtocolClientStatusFlag OBJECT-TYPE

```
SYNTAX      INTEGER {
    unknown(0),
    warmup(1),
    freerun(2),
    fastTrackingMode(3),
    normalMode(4),
    holdoverMode(5),
    bridgingMode(6),
    testMode(7)
}
```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The current Operational Status of the client side of the DTI interface. This MIB object should be updated no more than once per second.

The Client status values are:

```
'warmup'
    The client clock has not stabilized yet.
'freerun'
    The client has a stable clock but has not locked
    to an external clock reference yet.
'fastTrackingMode'
    The client is in the process of locking to the
```

```

        timing reference.
'normalMode'
    The client is locked to the timing reference.
'holdover'
    The client has lost the timing reference and is in
    holdover mode.
'bridgingMode'
    The client has lost its timing reference but is under
    acceptable operational conditions based on previous
    data acquired in 'normalMode' state.
'testMode' (Rephrase this definition)
    The client has entered in Test Mode for this DTI
    interface entry due to a set to 'true' of
    dtiProtocolControlTestMode or because is receiving
    Test Signaling frames from the DTI Server remote end."
REFERENCE
    "DOCSIS Timing Interface Specification, DTI Client
    Operation."
 ::= { dtiProtocolEntry 4 }

dtiProtocolServerToDState OBJECT-TYPE
    SYNTAX      INTEGER {
        valid(1),
        invalid(2)
    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Indicates the validity of the Time of Day.
        Possible values are 'valid' and 'invalid'.
        When the object dtiProtocolControlToDValue
        Corresponding to this Client-Server connection or the
        global object dtiServerGlobalToDValue
        is set by a manager rather than acquired via the protocol,
        'valid'.
        A ToD invalid state could be a consequence of an improper
        configuration of dtiServerToDSources or connectivity
        problems within the ToD protocol."
    ::= { dtiProtocolEntry 5 }

dtiProtocolServerToDType OBJECT-TYPE
    SYNTAX      INTEGER {
        default(1),
        userTime(2),
        ntpv4(3),
        gps(4)
    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Indicates the current Time of Day source for DTI
        Server-Client connection.
        'default' indicates the ToD of the DTI server is used and
        implies that this value is not updated via GPS or NTP.
        'userTime' indicates the ToD value was configured by the
        management interface. This object reports 'userTime'
        when dtiProtocolServerToDValue is set by a manager
        entity.
        'ntpv4' indicates NTP, Network Time Protocol version 4.
        'gps' indicates GPS ToD gpssec."
    ::= { dtiProtocolEntry 6 }

dtiProtocolServerToDValue OBJECT-TYPE

```



```
SYNTAX      OCTET STRING (SIZE (0|33))
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The value of the Time of Day sent by the server in the
    format DDDDD.YYYY/MM/DD.HH:MM:SS.SHH:F.D independent of
    being transmitted in short or verbose message format in
    the DTI Server Frame. Reading this object returns the ToD
    value for the specific DTI Client-Server connection."
 ::= { dtiProtocolEntry 7 }

dtiProtocolServerCableAdvanceFlag OBJECT-TYPE
SYNTAX      INTEGER {
    valid(1),
    invalid(2),
    manual(3)
}
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "This object indicates the cable advance status of the DTI
    Server Frame. This object is not updated more than once per
    second. If dtiProtocolServerCableAdvanceValue is set to a
    valid value by a manager entity, this objects reports
    'manual'."
 ::= { dtiProtocolEntry 8 }

dtiProtocolServerCableAdvanceValue OBJECT-TYPE
SYNTAX      DtiCableAdvance
UNITS       "clockSamples"
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
    "A Set to zero-length string returns the control of the
    Cable Advance Field and Cable Advance Flag Bit to the DTI
    server.

    Setting this object to a valid value sets the DTI Server
    Frames Frame Cable Advance field to the specified value and
    sets the Cable Advance Status Flag BIT to '1' (valid), and
    sets the object dtiProtocolServerCableAdvanceFlag to
    'manual' for this entry."
 ::= { dtiProtocolEntry 9 }

dtiProtocolClientPhaseError OBJECT-TYPE
SYNTAX      Integer32 (-32767..32767)
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The phase error measured at the client as reported to the
    server in the DTI Client Frames. This object is quantified
    in units of a 149.8 MHz clock cycles. This object
    represents the 16 MSB bits of the 3 octet Client Clock
    Integrated Phase field."
 ::= { dtiProtocolEntry 10 }

dtiProtocolClientVersion OBJECT-TYPE
SYNTAX      Unsigned32 (0..1023)
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "This object reports the client DTI protocol version as
    indicated in the 10-bit field 'CLIENT DTI VERSION'."
```

```

 ::= { dtiProtocolEntry 11 }

dtiProtocolClientPathTraceability OBJECT-TYPE
    SYNTAX      Unsigned32 (0..4294967295)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "A reference to the index of dtiPathTraceabilityTable to
        which this DTI Client-Server connection has associated its
        traceable path. It is likely that many traceable paths of
        DTI clients would have the same rootServer and proxy server
        for multiple Client Server. A manager application can build
        the Traceability Path, e.g., of a DTI client,
        if reading this entry from a proxy DTI server as follows:

        Client:
            This entry DTI Client IP Address,
            This entry ifIndex, (see below)
            dtiProtocolClientVersion
        Proxy Server:
            dtiPathTraceabilityServerInetAddr,
            dtiPathTraceabilityServerOutPhyIdx,
            dtiPathTraceabilityServerProtVersion
        RootServer:
            dtiPathTraceabilityRootServerInetAddr,
            dtiPathTraceabilityRootServerOutPhyIdx,
            dtiPathTraceabilityRootServerProtVersion

        For the DTI client the Physical port would correspond to:
            PhysicalIndex where
            entAliasMappingIdentifier.PhysicalIndex = ifIndex"
 ::= { dtiProtocolEntry 12 }

dtiProtocolServerClientStableFlag OBJECT-TYPE
    SYNTAX      INTEGER {
        valid(1),
        invalid(2)
    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object indicates the client performance stable status of the DTI
        Server Frame. This object is not updated more than once per
        second."
 ::= { dtiProtocolEntry 13 }

--
-- DTI Protocol Control
-- Provides mechanisms to control the DTI Client-Server connections
-- on a pair basis

dtiProtocolControlTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF DtiProtocolControlEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This table is an extension of dtiProtocolTable for the
        control of individual DTI Client-Server connections."
 ::= { dtiProtocolObjects 2 }

dtiProtocolControlEntry OBJECT-TYPE
    SYNTAX      DtiProtocolControlEntry
    MAX-ACCESS  not-accessible

```

```
STATUS      current
DESCRIPTION
    "A conceptual row entry of this table."
INDEX { ifIndex }
 ::= { dtiProtocolControlTable 1 }

DtiProtocolControlEntry ::= SEQUENCE {
    dtiProtocolControlTimeInterval      Unsigned32,
    dtiProtocolControlErrorRateInterval Unsigned32,
    dtiProtocolControlJitterTimeInterval Unsigned32,
    dtiProtocolControlTestMode         TruthValue,
    dtiProtocolControlToDValue         OCTET STRING
}

dtiProtocolControlTimeInterval OBJECT-TYPE
SYNTAX      Unsigned32 (0..3600)
UNITS       "seconds"
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
    "The time interval used to calculate
    dtiProtocolPerformanceDelay and
    dtiProtocolPerformanceWanderTSeconds for this Client-Server
    connection.
    The value zero indicates the Wander for T Seconds and
    Protocol Delay is not calculated."
DEFVAL { 0 }
 ::= { dtiProtocolControlEntry 1 }

dtiProtocolControlErrorRateInterval OBJECT-TYPE
SYNTAX      Unsigned32 (0..3600)
UNITS       "seconds"
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
    "The time interval used to calculate the value of
    dtiProtocolPerformanceFrameErrorRate for this Client-Server
    connection. The value zero indicates no current FER
    measurements are being taken."
DEFVAL { 0 }
 ::= { dtiProtocolControlEntry 2 }

dtiProtocolControlJitterTimeInterval OBJECT-TYPE
SYNTAX      Unsigned32 (0..3600)
UNITS       "seconds"
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
    "The time interval used to calculate
    dtiProtocolPerformancePeakToPeakJitter for this
    Client-Server connection.
    The value zero indicates no Jitter measurements are being
    taken."
DEFVAL { 0 }
 ::= { dtiProtocolControlEntry 3 }

dtiProtocolControlTestMode OBJECT-TYPE
SYNTAX      TruthValue
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
    "If set to 'true' starts the sequence of Test Mode
    on this DTI Client-Server connection."
```

```

        The Test Mode consist in a sequence of all DTI frames
        set to '1' before the Manchester line code is applied."
 ::= { dtiProtocolControlEntry 4 }

dtiProtocolControlToDValue OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE (0|33))
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The control object to specify a particular ToD value to
        a Client-Server connection. The format of this object is
        DDDD.YYYY/MM/DD.HH:MM:SS.SHH:F.D independent of being
        transmitted in short or verbose message format in the DTI
        Server Frame.

        Setting this object to a zero length string resets the ToD
        source of this DTI Client-Server connection to the current
        Server ToD reference of dtiServerGlobalToDMethod and
        dtiServerGlobalToDValue.

        Reading this value returns the last set value of this
        object or zero length string if a global ToD value and/or
        method were lately performed."
 ::= { dtiProtocolControlEntry 5 }

--
-- DTI protocol performance
--

dtiProtocolPerformanceTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF DtiProtocolPerformanceEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The DTI protocol performance metrics of a DTI
        Client-Server Connection. In the DTI server an entry
        of this table registers the DTI Client frames performance.
        In the DTI client the measurements
        correspond to the DTI server frames performance."
 ::= { dtiProtocolObjects 3 }

dtiProtocolPerformanceEntry OBJECT-TYPE
    SYNTAX      DtiProtocolPerformanceEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A conceptual row entry in the DTI performance table.
        This table is an extension of dtiProtocolTable for
        entries with dtiProtocolEntityType equals to 'root'
        and 'server'."
    INDEX { ifIndex }
 ::= { dtiProtocolPerformanceTable 1 }

DtiProtocolPerformanceEntry ::= SEQUENCE {
    dtiProtocolPerformanceDelay          Unsigned32,
    dtiProtocolPerformanceFrameErrorRate EntitySensorValue,
    dtiProtocolPerformancePeakToPeakJitter Integer32,
    dtiProtocolPerformanceWander35Second Unsigned32,
    dtiProtocolPerformanceWanderTSeconds Unsigned32,
    dtiProtocolPerformanceFrameErrorRateScale EntitySensorDataScale
}

dtiProtocolPerformanceDelay OBJECT-TYPE

```

```

SYNTAX      Unsigned32
UNITS       "nanoseconds"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The roundtrip delay measured in nanoseconds."
 ::= { dtiProtocolPerformanceEntry 1 }

```

dtiProtocolPerformanceFrameErrorRate OBJECT-TYPE

```

SYNTAX      EntitySensorValue
UNITS       "FER"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "Frame Error Rate (FER) corresponds to the ratio of DTI
    protocol frames in error and the total of frames received
    during a period of time.
    Frames in error and total frames correspond to the delta
    of ifErrors and ifUcastPackets MIB objects respectively
    during the period of time defined by
    dtiProtocolControlErrorRateInterval object. For the DTI
    server the FER value is based on the DTI client interval
    frames and for the DTI client, corresponds to the DTI
    server interval frames.
    This value is read in conjunction with
    dtiProtocolPerformanceFrameErrorRateScale which determines
    The exponential value of this object. Although RFC 3434
    recommends the use of the EntitySensorValue and
    EntitySensorDataScale data types in conjunction with an
    objects of type EntitySensorDataScale and
    EntitySensorPrecision, Those two data types are not defined
    in this table as those are not used in the context of a
    sensor like in entPhySensorTable where specific units and
    precision is relevant."
 ::= { dtiProtocolPerformanceEntry 2 }

```

dtiProtocolPerformancePeakToPeakJitter OBJECT-TYPE

```

SYNTAX      Integer32 (-10000..10000)
UNITS       "picoseconds"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The maximum phase variation (with frequency components
    > 10 Hz) in the client frame with respect to the master
    clock and reported in the DTI client frame for the time
    interval specified in dtiProtocolControlJitterTimeInterval
    object."
 ::= { dtiProtocolPerformanceEntry 3 }

```

dtiProtocolPerformanceWander35Second OBJECT-TYPE

```

SYNTAX      Unsigned32(1..4294967295)
UNITS       "picoseconds"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The largest wander value observed by the server in a
    35-second interval for the corresponding Client-Server
    connection."
 ::= { dtiProtocolPerformanceEntry 4 }

```

dtiProtocolPerformanceWanderTSeconds OBJECT-TYPE

```

SYNTAX      Unsigned32(1..4294967295)
UNITS       "picoseconds"

```

```
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The largest wander value observed by the server in a time
    interval defined by dtiWanderTimeInterval for the
    corresponding Client-Server connection."
 ::= { dtiProtocolPerformanceEntry 5 }

dtiProtocolPerformanceFrameErrorRateScale OBJECT-TYPE
SYNTAX EntitySensorDataScale
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The scale used to measure the Frame Error Rate (FER)
    for dtiProtocolPerformanceFrameErrorRate."
 ::= { dtiProtocolPerformanceEntry 6 }
--
-- DTI Client State Machine
--

dtiProtocolClientFsmStatsTable OBJECT-TYPE
SYNTAX SEQUENCE OF DtiProtocolClientFsmStatsEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
    "Statistics associated to the DTI client state machine.
    This table contains counters of the most significant
    DTI finite State machine transitions and states as
    reported in the DTI Client Frame.

    Below is a description of the State Transition and
    counters supported (not all State Transitions need
    object definitions):

    T1: Warmup to Free-run
        Normally once after the clock warms up.
        Rather than defining a counter for this
        state transition, the managed system updates
        the object ifDiscontinuity with the time when
        the clock performs a transition to Free-run
        State
    T2: Free-run to Fast
        Not defined counter. T2 = T3 + 1
    T3: Fast to Free-run (dtiProtocolClientFsmStatsT3Count)
    T4: Fast to Normal (dtiProtocolClientFsmStatsT4Count)
        Counts the times the clock enters to the normal
        mode of operation.
    T5: Normal to Bridging
        Not defined counter. T5 = T6 + T7
    T6: Bridging to Normal (dtiProtocolClientFsmStatsT6Count)
    T7: Bridging to Holdover (dtiProtocolClientFsmStatsT7Count)
    T8: Holdover to Fast
        No counter defined.
        If DTI Client State is Holdover, T8 = T7 - 1
        other DTI Client State, T8 = T7

    This table also includes the active time the DTI client is
    in Normal state(dtiProtocolClientFsmStatsNormalActiveTime)
    and Holdover state
    (dtiProtocolClientFsmStatsHoldoverActiveTime)."
```

REFERENCE

"DOCSIS Timing Interface Specification, DTI Client Mode Transition Diagram."

```

 ::= { dtiProtocolObjects 6 }

dtiProtocolClientFsmStatsEntry OBJECT-TYPE
    SYNTAX      DtiProtocolClientFsmStatsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A conceptual row entry in the DTI FSM Statistics table.
        In the DTI Server (proxy) this table has an entry for
        each DTI client interface as well as an entry for each
        Server (proxy) interface to root server. In the DTI client
        there is an entry for each client interface."
    INDEX { ifIndex }
 ::= { dtiProtocolClientFsmStatsTable 1 }

DtiProtocolClientFsmStatsEntry ::= SEQUENCE {
    dtiProtocolClientFsmStatsT3Count      Counter32,
    dtiProtocolClientFsmStatsT4Count      Counter32,
    dtiProtocolClientFsmStatsT6Count      Counter32,
    dtiProtocolClientFsmStatsT7Count      Counter32,
    dtiProtocolClientFsmStatsNormalActiveTime Counter32,
    dtiProtocolClientFsmStatsHoldoverActiveTime Counter32
}

dtiProtocolClientFsmStatsT3Count OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Counts the Transition from Fast to Free-run state (T3).
        This transition occurs when the client aborts the clock
        acquisition due to a high Frame Error Rate."
    REFERENCE
        "DOCSIS Timing Interface Specification, DTI Client Mode
        Transition Diagram."
 ::= { dtiProtocolClientFsmStatsEntry 1 }

dtiProtocolClientFsmStatsT4Count OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Counts the Transition from Fast to Normal state (T4).
        This transition occurs when the client acquires for the
        first time or recovers a stable clock reference."
    REFERENCE
        "DOCSIS Timing Interface Specification, DTI Client Mode
        Transition Diagram."
 ::= { dtiProtocolClientFsmStatsEntry 2 }

dtiProtocolClientFsmStatsT6Count OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Counts the Transition from Bridging to Normal state (T6).
        This transition occurs when in Bridging mode the Frame
        Error rate is reduced and the DTI client recovers the clock
        reference before a bridging mode timeout."
    REFERENCE
        "DOCSIS Timing Interface Specification, DTI Client Mode
        Transition Diagram."
 ::= { dtiProtocolClientFsmStatsEntry 3 }

```

```
dtiProtocolClientFsmStatsT7Count OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Counts the Transition from Bridging to Holdover state
        (T7). This transition occurs when the client is in Bridging
        mode, the bridging timeout expires and the DTI client
        enters in Holdover mode."
    REFERENCE
        "DOCSIS Timing Interface Specification, DTI Client Mode
        Transition Diagram."
    ::= { dtiProtocolClientFsmStatsEntry 4 }
```

```
dtiProtocolClientFsmStatsNormalActiveTime OBJECT-TYPE
    SYNTAX      Counter32
    UNITS       "milliseconds"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The cumulative count in 10 KHz clock cycles the DTI client
        is being in Normal Mode.
        The percentage of time the DTI client is being operating in
        Normal Mode is calculated as the ratio of this object and
        sysUpTime within two consecutive measurements. Note that
        the value sysUpTime is given in hundreds of seconds,
        therefore it needs to be multiplied by 100."
    REFERENCE
        "DOCSIS Timing Interface Specification, DTI Client Mode
        Transition Diagram."
    ::= { dtiProtocolClientFsmStatsEntry 5 }
```

```
dtiProtocolClientFsmStatsHoldoverActiveTime OBJECT-TYPE
    SYNTAX      Counter32
    UNITS       "milliseconds"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The cumulative count in 10 KHz clock cycles the DTI client
        is being in Holdover Mode.
        The percentage of time the DTI client is being operating in
        Holdover Mode is calculated as the ratio of this object and
        sysUpTime within two consecutive measurements. Note that
        the value sysUpTime is given in hundreds of seconds;
        therefore it needs to be multiplied by 100."
    REFERENCE
        "DOCSIS Timing Interface Specification, DTI Client Mode
        Transition Diagram."
    ::= { dtiProtocolClientFsmStatsEntry 6 }
```

```
--
-- Path Traceability Table
--
```

```
dtiPathTraceabilityTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF DtiPathTraceabilityEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The Traceable Path information sent from the DTI server
        to the client. The traceability path includes node
        information about DTI root server and DTI server (proxy)."
```



The entries of this table are pointed by dtiProtocolTable in both DTI Server and Client.  
 The DTI server may aggregate in a single entry all the root-to-server traceable paths that are common across multiple DTI clients."

::= { dtiProtocolObjects 4 }

dtiPathTraceabilityEntry OBJECT-TYPE

SYNTAX DtiPathTraceabilityEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A conceptual row entry in the DTI Path Traceability table. This table is an expansion of the dtiProtocolTable for entries with dtiProtocolEntityType equal to 'server' and 'client'."

INDEX { dtiPathTraceabilityIndex }

::= { dtiPathTraceabilityTable 1 }

DtiPathTraceabilityEntry ::= SEQUENCE {

dtiPathTraceabilityIndex	Unsigned32,
dtiPathTraceabilityRootServerInetAddrType	InetAddressType,
dtiPathTraceabilityRootServerInetAddr	InetAddress,
dtiPathTraceabilityRootServerOutPhyIdx	PhysicalIndex,
dtiPathTraceabilityServerInetAddrType	InetAddressType,
dtiPathTraceabilityServerInetAddr	InetAddress,
dtiPathTraceabilityServerOutPhyIdx	PhysicalIndex,
dtiPathTraceabilityRootServerProtVersion	Unsigned32,
dtiPathTraceabilityServerProtVersion	Unsigned32

}

dtiPathTraceabilityIndex OBJECT-TYPE

SYNTAX Unsigned32 (1..4294967295)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The index of this conceptual entry."

::= { dtiPathTraceabilityEntry 1 }

dtiPathTraceabilityRootServerInetAddrType OBJECT-TYPE

SYNTAX InetAddressType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The type of internet address for dtiPathTraceabilityRootServerInetAddr."

::= { dtiPathTraceabilityEntry 2 }

dtiPathTraceabilityRootServerInetAddr OBJECT-TYPE

SYNTAX InetAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The IP address of the clock synchronization root server. If this management entity has dtiServerExternalSource 'notExternal' this value returns an empty octet string and dtiPathTraceabilityRootServerInetAddrType. The type of this address is determined by the value of the dtiPathTraceabilityRootServerInetAddrType object."

::= { dtiPathTraceabilityEntry 3 }

dtiPathTraceabilityRootServerOutPhyIdx OBJECT-TYPE

SYNTAX PhysicalIndex

```

MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The physical port number of the clock synchronization
    root server."
 ::= { dtiPathTraceabilityEntry 4 }

dtiPathTraceabilityServerInetAddrType OBJECT-TYPE
SYNTAX InetAddressType
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The type of internet address for
    dtiPathTraceabilityServerInetAddr."
 ::= { dtiPathTraceabilityEntry 5 }

dtiPathTraceabilityServerInetAddr OBJECT-TYPE
SYNTAX InetAddress
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The IP address of the clock synchronization server
    associated with this management entity. The type of this
    address is determined by the value of the
    dtiPathTraceabilityServerInetAddrType object."
 ::= { dtiPathTraceabilityEntry 6 }

dtiPathTraceabilityServerOutPhyIdx OBJECT-TYPE
SYNTAX PhysicalIndex
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The physical port number of the clock synchronization
    server associated with this management entity."
 ::= { dtiPathTraceabilityEntry 7 }

dtiPathTraceabilityRootServerProtVersion OBJECT-TYPE
SYNTAX Unsigned32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The DTI Protocol Version of the clock synchronization root
    server."
 ::= { dtiPathTraceabilityEntry 8 }

dtiPathTraceabilityServerProtVersion OBJECT-TYPE
SYNTAX Unsigned32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The DTI Protocol Version of the clock synchronization
    server associated with this management entity."
 ::= { dtiPathTraceabilityEntry 9 }

--
-- DTI server objects
--

dtiServerRootClockType OBJECT-TYPE
SYNTAX INTEGER {
    ituI(1),
    ituII(2),
    ituIII(3),

```

```
        st3(4)
    }
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION
    "The clock type of the root server this server is
    connected. If dtiServerHopCount is 'root' the clock type
    is the self clock.
    'ituI' corresponds to an ITU Type I clock.
    'ituII' corresponds to an ITU Type II clock.
    'ituIII' corresponds to an ITU Type III clock.
    'st3' corresponds to an ANSI T1.101 ST3 clock."
 ::= { dtiServerProperties 1 }

dtiServerHopCount OBJECT-TYPE
SYNTAX        INTEGER { root(1), proxy(2) }
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION
    "This object indicates the type of the DTI server.
    'root' indicates the server is the DTI root server. 'proxy'
    indicates the DTI server is connected to a root DTI
    server."
 ::= { dtiServerProperties 2 }

dtiServerExternalTimingSource OBJECT-TYPE
SYNTAX        INTEGER {
    noExternal(1),
    gps(2),
    network(3)
}
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION
    "The type of external clock that provides the reference
    for this DTI server.
    'none' indicates the DTI server has no timing reference
    input.
    'gps' indicates the timing source is GPS.
    'network' indicates the timing is obtained from a network
    such as a PDH, or SONET network."
 ::= { dtiServerProperties 3 }

dtiServerToDSources OBJECT-TYPE
SYNTAX        BITS {
    default(0),
    userTime(1),
    ntpv4(2),
    gps(3)
}
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION
    "Indicates the DTI Server's registered Time of Day sources.
    A bit set to '1' has the following meaning:

    'default' the internal DTI Server ToD reference without
    external updates.
    'userTime' indicates the capability of ToD being set by a
    manager entity either setting
    dtiServerGlobalToDValue or
    dtiProtocolControlToDValue.
    'ntpv4' indicates support of NTP Network Time Protocol
```

```

        version 4.
        'gps' indicates support of GPS ToD gpssec.

        The initialization or configuration of GPS or NTPv4 ToD
        sources is not defined in this MIB module."
 ::= { dtiServerProperties 4 }
--
--
--
dtiServerGlobalTimeInterval OBJECT-TYPE
SYNTAX      Unsigned32 (0..3600)
UNITS       "seconds"
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
    "The time interval used to calculate
    dtiProtocolPerformanceDelay and
    dtiProtocolPerformanceWanderTSeconds for all Client-Server
    connections of the managed system.
    A set to this object overrides all the entries of
    dtiProtocolControlTimeInterval previously configured.
    The value zero indicates the Wander for T Seconds and
    Protocol Delay is not calculated."
DEFVAL { 0 }
 ::= { dtiServerGlobalParameters 1 }

dtiServerGlobalErrorRateInterval OBJECT-TYPE
SYNTAX      Unsigned32 (0..3600)
UNITS       "seconds"
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
    "The time interval used to calculate the value of
    dtiProtocolPerformanceFrameErrorRate for all Client-Server
    connections of the managed system.
    A set to this object overrides all the entries of
    dtiProtocolControlErrorRateInterval previously configured.
    The value zero indicates no current FER measurements are
    being taken."
DEFVAL { 0 }
 ::= { dtiServerGlobalParameters 2 }

dtiServerGlobalJitterTimeInterval OBJECT-TYPE
SYNTAX      Unsigned32 (0..3600)
UNITS       "seconds"
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
    "The time interval used to calculate
    dtiProtocolPerformancePeakToPeakJitter for all
    Client-Server connections of the managed system.
    A set to this object overrides all the entries of
    dtiProtocolControlErrorRateInterval previously configured.
    The value zero indicates no Jitter measurements are being
    taken."
DEFVAL { 0 }
 ::= { dtiServerGlobalParameters 3 }

dtiServerGlobalToDMethod OBJECT-TYPE
SYNTAX      INTEGER {
    default (1),
    userTime (2),

```

```

        ntpv4(3),
        gps(4)
    }
MAX-ACCESS    read-write
STATUS        current
DESCRIPTION
    "Selects the type of ToD Source for the DTI server
    operation. The set of this object to its enumerated
    values is described below:

    'default' instructs the DTI server to use the last ToD
        reference as the value for its internal clock.
    'userTime' instructs the DTI server to use the value of
        dtiServerGlobalToDValue for the DTI server internal
        time ticks clock.
    'ntpv4' instructs the DTI server to use the NTP Network
        Time Protocol version 4 as the ToD reference.
    'gps' instructs the DTI server to use the GPS gpssec ToD
        as the ToD reference.

    It is recommended to reject an SNMP SET to a particular ToD
    method if such ToD capability is not active or requires
    other configuration steps.

    A successful set of this object sets to a zero length
    string all the DTI Client-Server connections that
    previously were set via dtiProtocolControlToDValue to
    particular ToD Values."
 ::= { dtiServerGlobalParameters 4 }

dtiServerGlobalToDValue OBJECT-TYPE
SYNTAX        OCTET STRING (SIZE (0|33))
MAX-ACCESS    read-write
STATUS        current
DESCRIPTION
    "The control object to specify the ToD value for all DTI
    server Client-Server connections. The format of this object
    is DDDDD.YYYY/MM/DD.HH:MM:SS.SHH:F.D independent of being
    transmitted in short or verbose message format in the DTI
    Server Frame.

    A set of this object to a valid value different of zero
    length string automatically sets dtiServerGlobalToDMethod
    to 'userTime'. Setting this object to zero re-establishes
    the ToD source to the current Server ToD reference
    dtiServerGlobalToDMethod 'default', 'ntpProtocol'
    or 'gps'."
 ::= { dtiServerGlobalParameters 5 }
--
--
--

-- Conformance information
dtiMibConformance OBJECT IDENTIFIER ::= { dtiMib 2 }
dtiMibCompliances  OBJECT IDENTIFIER ::= { dtiMibConformance 1 }
dtiMibGroups       OBJECT IDENTIFIER ::= { dtiMibConformance 2 }

-- Compliance statements
dtiMibCompliance MODULE-COMPLIANCE
    STATUS        current
    DESCRIPTION
        "The compliance statement for DTI."
MODULE

```

```

-- unconditionally mandatory groups
MANDATORY-GROUPS {
    dtiBaseGroup
}

-- conditionally mandatory groups
GROUP dtiServerGroup
    DESCRIPTION
        "Mandatory group for DTI server."

GROUP dtiClientGroup
    DESCRIPTION
        "Mandatory group for DTI client."

OBJECT dtiProtocolServerStatusFlag
    SYNTAX      INTEGER {
        unknown(0),
        warmup(1),
        freerun(2),
        fastTrackingMode(3),
        normalMode(4),
        holdoverMode(5),
        testMode(7)
    }
    DESCRIPTION
        " The value 'clientStable' is deprecated and not used."

OBJECT      dtiProtocolServerCableAdvanceValue
    MIN-ACCESS read-only
    DESCRIPTION
        "This object is read-write for DTI server interfaces and
        read-only for DTI client interfaces (e.g., a DTI interface
        connected to a DTI server or a DTI root server)."
```

```

OBJECT dtiProtocolControlErrorRateInterval
    SYNTAX      Unsigned32 (1..3600)
    WRITE-SYNTAX Unsigned32 (0..3600)
    MIN-ACCESS read-only
    DESCRIPTION
        "This object is read-write for DTI server interfaces and
        write access for DTI client interfaces is not required.
        (e.g., a DTI interface connected to a DTI server or a DTI
        root server).The Error Rate Interval for DTI client
        interfaces is vendor-specific with a suggested value in
        the interval from 1 to 60 seconds."

 ::= { dtiMibCompliances 1 }

dtiBaseGroup OBJECT-GROUP
    OBJECTS {
        dtiProtocolServerToDState,
        dtiProtocolServerToDValue,
        dtiProtocolServerCableAdvanceFlag,
        dtiProtocolServerCableAdvanceValue,
        dtiProtocolClientPhaseError,
        dtiProtocolClientVersion,
        dtiProtocolEntityType,
        dtiProtocolClientClockType,
        dtiProtocolServerStatusFlag,
        dtiProtocolClientStatusFlag,
        dtiProtocolServerToDState,
        dtiProtocolServerToDType,
    }

```

```

dtiProtocolServerToDValue,
dtiProtocolServerCableAdvanceFlag,
dtiProtocolServerCableAdvanceValue,
dtiProtocolClientPhaseError,
dtiProtocolClientVersion,
dtiProtocolClientPathTraceability,
dtiPathTraceabilityRootServerInetAddrType,
dtiPathTraceabilityRootServerInetAddr,
dtiPathTraceabilityRootServerOutPhyIdx,
dtiPathTraceabilityServerInetAddrType,
dtiPathTraceabilityServerInetAddr,
dtiPathTraceabilityServerOutPhyIdx,
dtiPathTraceabilityRootServerProtVersion,
dtiPathTraceabilityServerProtVersion,
dtiProtocolControlErrorRateInterval,
dtiProtocolPerformanceFrameErrorRate,
dtiProtocolPerformanceFrameErrorRateScale,
dtiProtocolServerClientStableFlag
}
STATUS      current
DESCRIPTION
    "Group of object applicable to DTI Server and DTI Client."
 ::= { dtiMibGroups 1 }

```

```

dtiServerGroup OBJECT-GROUP
OBJECTS {
dtiProtocolControlTimeInterval,
dtiProtocolControlJitterTimeInterval,
dtiProtocolControlTestMode,
dtiProtocolControlToDValue,
dtiProtocolPerformanceDelay,
dtiProtocolPerformancePeakToPeakJitter,
dtiProtocolPerformanceWander35Second,
dtiProtocolPerformanceWanderTSeconds,
dtiServerRootClockType,
dtiServerHopCount,
dtiServerExternalTimingSource,
dtiServerToDSources,
dtiServerGlobalTimeInterval,
dtiServerGlobalErrorRateInterval,
dtiServerGlobalJitterTimeInterval,
dtiServerGlobalToDMethod,
dtiServerGlobalToDValue
}
STATUS      current
DESCRIPTION
    "Group of objects applicable to DTI Server only."
 ::= { dtiMibGroups 2 }

```

```

dtiClientGroup OBJECT-GROUP
OBJECTS {
dtiProtocolClientFsmStatsT3Count,
dtiProtocolClientFsmStatsT4Count,
dtiProtocolClientFsmStatsT6Count,
dtiProtocolClientFsmStatsT7Count,
dtiProtocolClientFsmStatsNormalActiveTime,
dtiProtocolClientFsmStatsHoldoverActiveTime
}
STATUS      current
DESCRIPTION
    "Group of objects applicable to DTI Client."
 ::= { dtiMibGroups 3 }

```

END





## Annex C DTI-S CRC-16

The CRC-16 generator MUST be initialized with '1's during the preamble bits. After the last bit of preamble, the payload data MUST be shifted through the CRC-16 code generator. Following the last bit of the payload, the serial output MUST be switched from the transmit shift register to the output of the CRC-16 generator and 16 more '1's MUST be shifted through the generator to shift out the 16 bit CRC.

During Initialization and CRC output, the CRC-16 MUST be implemented as a shift register:

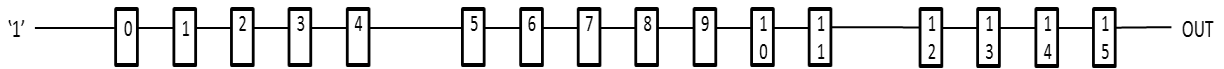


Figure C-1 - CRC Initialization

During Payload CRC checking, the CRC-16 MUST be implemented as shown.

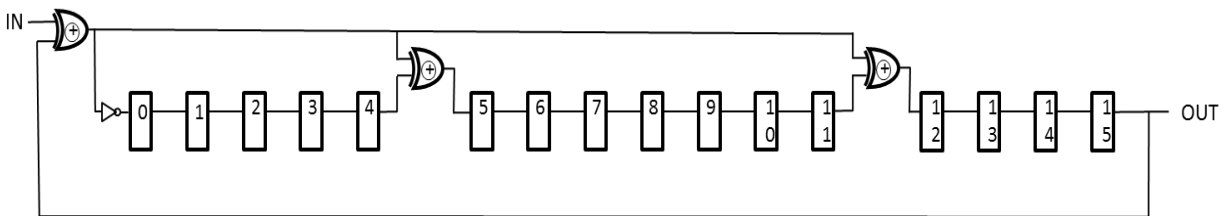


Figure C-2 - CRC Algorithms

As an example of DTI-S CRC16, when inputs are:

0x313233343536373839,

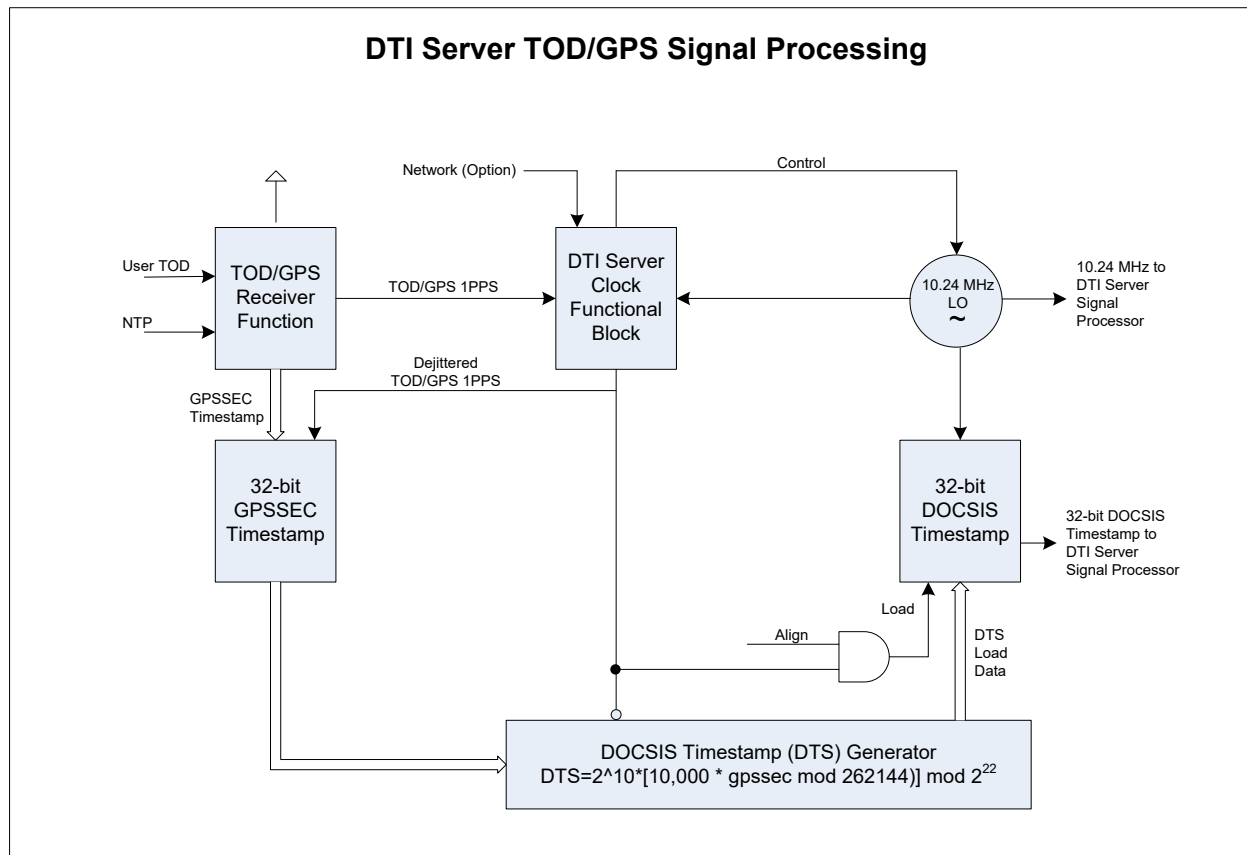
the CRC16 values are:

0xE4E0.

## Appendix I DTI Server Functional Description

This Appendix provides supporting information to assist development of compliant DTI servers.

Figure I-1 illustrates a reference diagram showing the server processing function relating to external references. Vendors are free to implement alternate internal architecture as long as all DTI server requirements are met. The relationship between the gpssec timescale and the DTI timestamp is discussed in detail in Section 6.2. This appendix discusses one method to support these requirements.

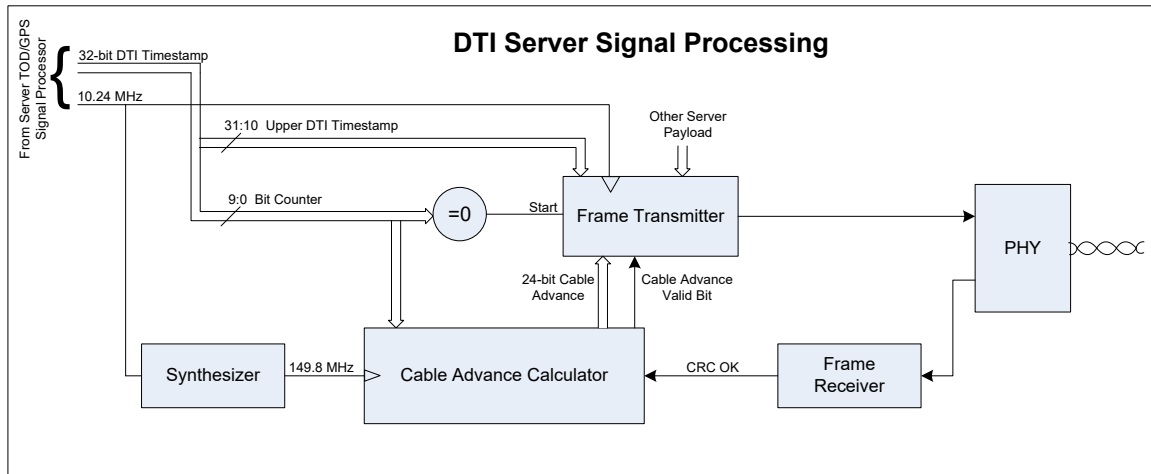


**Figure I-1 - Server TOD/GPS/Network Signal Processing Reference Block Diagram**

The core functional block in a Server is the DTI Server Clock. The Server Clock principle function is to control the 10.24 MHz master clock and a precise timing tick (shown as the dejittered 1PPS in the diagram) based on error measurement with respect to the external input. The external input may be either GPS or a Network reference. The Server supports a function to align the DOCSIS Timestamp to the GPS timescale (or externally supplied estimate of GPS time). The DOCSIS Timestamp Generator functional block illustrates this operation. The generator calculates the next DOCSIS Timestamp based on the current gpssec value. This value is loaded synchronously with the dejittered TOD/GPS 1PPS signal if the alignment control is asserted. The dejittered TOD/GPS 1PPS output is maintained in tight coherency with the 10.24 MHz clock to permit a synchronous alignment of the DOCSIS Timestamp within one masterclock period.

### I.1 Server DTI Signal Processing

A block diagram of the DTI Server signal processor is shown below in Figure I-2.



**Figure I-2 - DTI Server Reference Signal Processing Block Diagram**

The DTI Server Signal Processor generates the DTI timing signal, receives the reply from the client, calculates the round trip cable delay, and relays it back to the client as a cable advance value. The DTI Server Signal Processor receives (from the TOD/GPS/External Reference signal processor) a 10.24 MHz clock, and the 32-bit DOCSIS timestamp. The lower 10 bits of the DOCSIS timestamp are the actual bit counter for the DTI frame. The lower 10 bits of the DOCSIS timestamp will be referred to as "bit counter" for the rest of this explanation. The DTI frame is launched when the bit counter is zero. The DTI timing signal is clocked out using the 10.24 MHz clock (2 clocks per Manchester symbol). The DTI Frame transmitter will append the 68-bit preamble to the beginning of the frame and a 16-bit CRC to the end.

The client will receive the DTI timing signal and respond when its frame counter is at 512. The server DTI Frame Receiver will receive the response from the client and will issue its CRC OK after the 16<sup>th</sup> CRC bit has been received.

A cable delay measurement circuit measures the delay of the received CRC OK flag from where it would appear if the delay were zero. The measurement process may be started when the 10.24 MHz DTI counter is at  $746^{13}$  modulo 1024. Conceptually, if there is zero round-trip cable delay, the last bit of the CRC returning from client will arrive at the 746 server counter value. The received CRC\_OK flag terminates the measurement process. This produces a raw cable delay values that is updated at a 10 kHz rate. The raw cable delay value has a resolution of one 149.8 MHz clock cycle.

A cable delay filter then processes the output of the measurement circuit. The filtering process will remove any transient values and will average the delay value. The averaging process will also produce the fractional delay value.

The 24-bit Cable Advance value is derived by dividing the cable delay by 2.

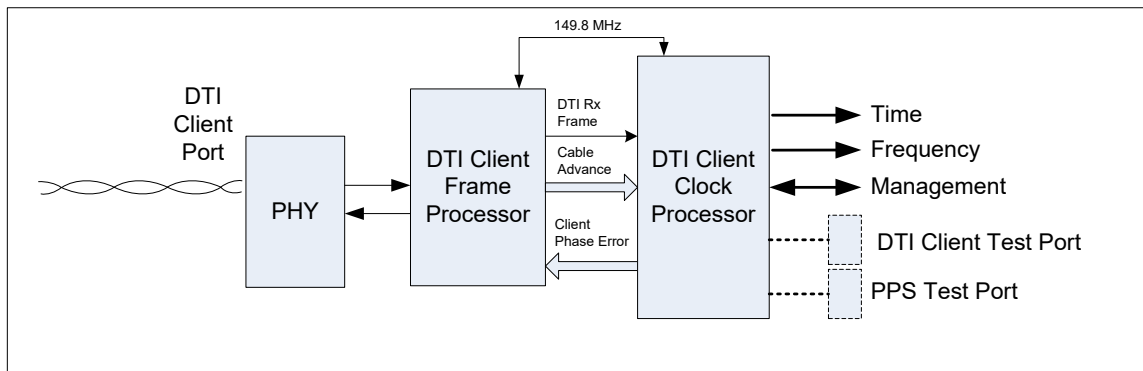
<sup>13</sup>  $746 = 512 + 234$

## Appendix II DTI Client Functional Description

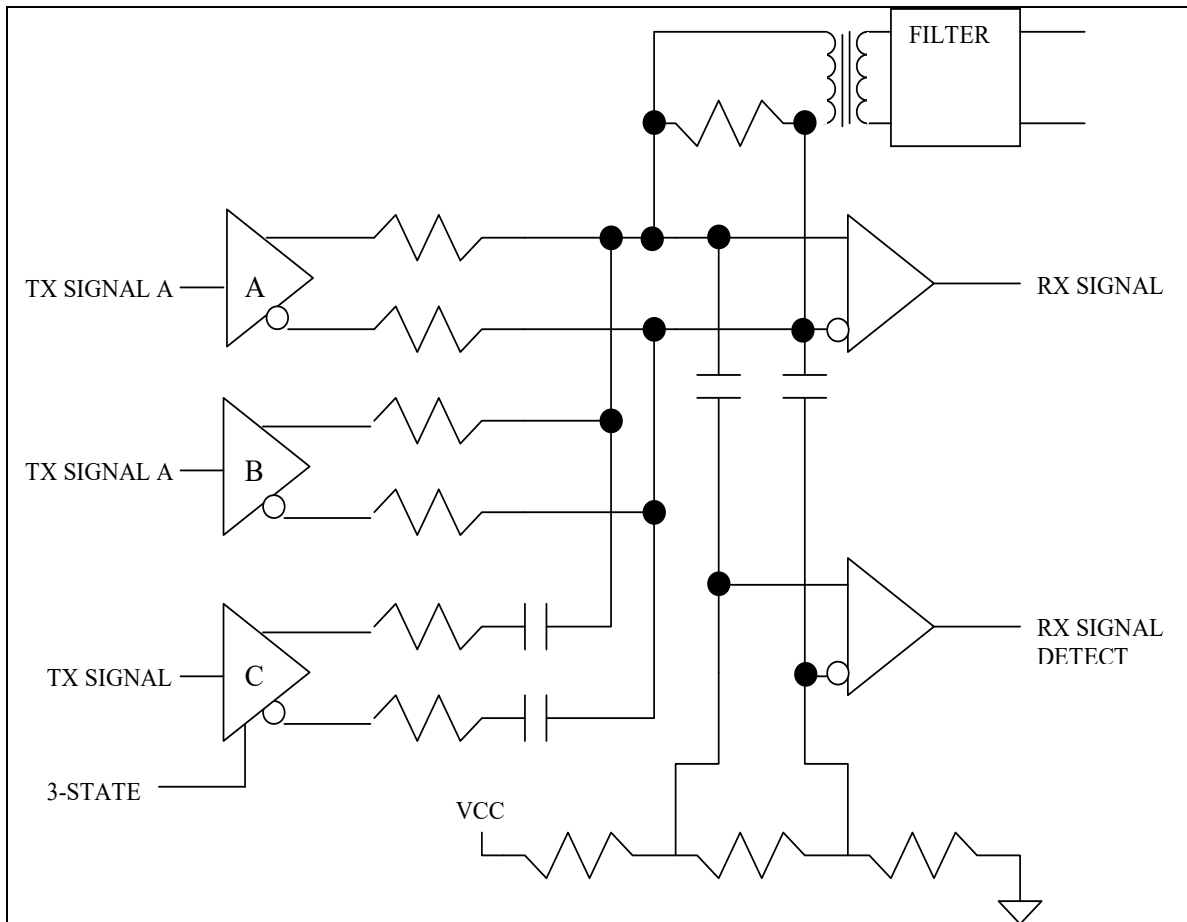
This appendix provides supporting information to assist development of compliant DTI servers.

### DTI Client Block Diagram

The block diagram shown below in Figure II-1 shows the data flow of the DTI client signal processing.



**Figure II-1 - DTI Client Block Diagram**



MOD	DRIVER A	DRIVER B	DRIVER C
Transmit	DTI Signal	DTI Signal	DTI Signal
Receive	'1'	'0'	High Impedance

Figure II-2 - Example DTI PHY Interface Circuit

## II.1 DTI Client PHY

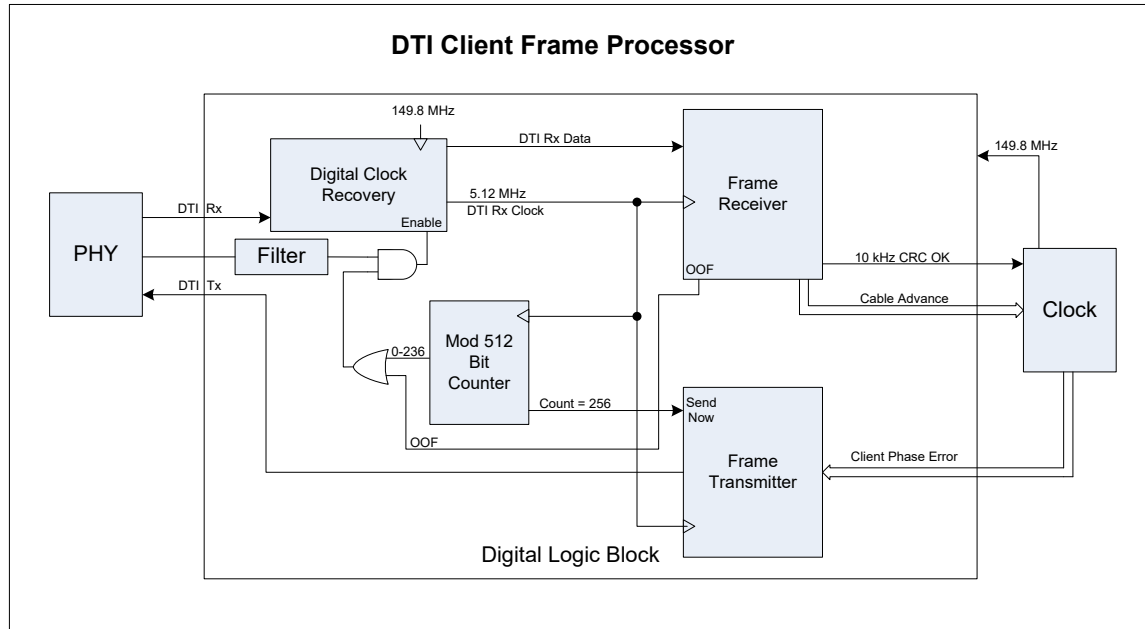
The ping-pong DTI timing signal is passed through an EMI filter to ensure EMI compliance and minimize susceptibility. The signal then flows through a transformer to block common mode noise on the DTI timing signal. When receiving the DTI signal, driver A is fixed at '1', driver B is fixed at '0', and driver C is set to high impedance which provides a 100 Ohm termination that is biased at the midpoint. The receive signal is sliced at the zero crossing to recover the data. A receive signal less than a nominal level (400 mV) is not interpreted as data. This can be accomplished by sending the receive signal through a digital comparator with a bias on the inputs as shown in Figure II-2.

The output of the burst detector is filtered so that it will provide a steady active state while data is present. The burst detect signal is used to qualify the RX\_SIG.

In the transmit direction the drivers A, B and C are active and in phase. Drivers A and B generate the NRZ Manchester symbols simultaneously. Driver C provides pre-emphasis for the transmitted DTI signal.

## II.2 DTI Client Frame Processor

A block diagram of an example DTI Frame processor is shown below in Figure II–3. The input to the DTI Frame processor block is an NRZ Manchester encoded digital signal from the PHY. The burst detect signal is also provided by the PHY (if available). The output from the DTI Frame Transmitter to the DTI PHY is a tri-state differential digital signal (DTI\_TX).



**Figure II–3 - DTI Client Frame Processor**

The digital clock recovery utilizes a 149.8 MHz sample clock. This sample clock is derived from 10.24 MHz using a 512/35 multiplier. The digital clock recovery circuit operates in 2 states: Tracking and Flywheel. The clock recovery tracks the input signal only while the receive signal is present, and does not track the transmit signal that will be present at the input when the transmitter is enabled.

While the receive signal is not being used, the clock recovery is in the flywheel state and uses the 149.8 MHz sample clock, which is tuned to the correct frequency by the clock, to generate the carrier frequency.

Initially the clock recovery may be gated by the filtered burst detect signal from the DTI PHY until framing is established. The client will not be transmitting until framing is established. Once framing is established, the mod 512 bit counter is aligned. Then the bit counter is decoded and utilized to enable the clock recovery only when the client transmitted burst is completed and the burst detect is triggered.

The entire DTI Client Frame Processor operates from the recovered 10.24 MHz server clock (and the associated 5.12 MHz Manchester bit clock) since the roundtrip measurement process in the DTI Server requires a fixed 256-bit 5.12 MHz symbol clock delay.

The Frame Receiver block detects the preamble, receives the DTI payload, checks the data integrity, and generates a 10 kHz CRC OK signal that is used to steer the clock. The device type, server status flags, and server cable advance, which are part of the DTI payload, are not updated if the frame has a CRC error.

The mod 512 bit counter is decoded to locate the beginning of the transmit slot. The frame transmitter generates the preamble, serializes the payload, appends a CRC16 cyclic redundancy checksum, and sends the serial bit stream to the DTI PHY. The transmitter also reports the current client clock phase error measurement. The control signal for the transmit portion of the DTI PHY needs to be decoded from the mod 512 bit counter, and is only asserted if the receiver has properly framed.

The receive signal from the PHY will be processed by the DTI Frame Processor. The DTI Frame Processor decodes the Manchester signal, locates the end of the preamble, and then extracts the payload data. A CRC16 check is done on the receive data, and is used to validate the payload data and generate the DTI RX FRAME signal. The Cable Advance from the payload data and the DTI RX FRAME signal are used to synchronize the client clock.

After the DTI Frame Processor has synchronized to the incoming DTI timing signal, and if the received frame is error free, the client response is launched when the bit counter in the DTI Frame Processor reaches 256.

### II.3 DTI Client Clock Processor

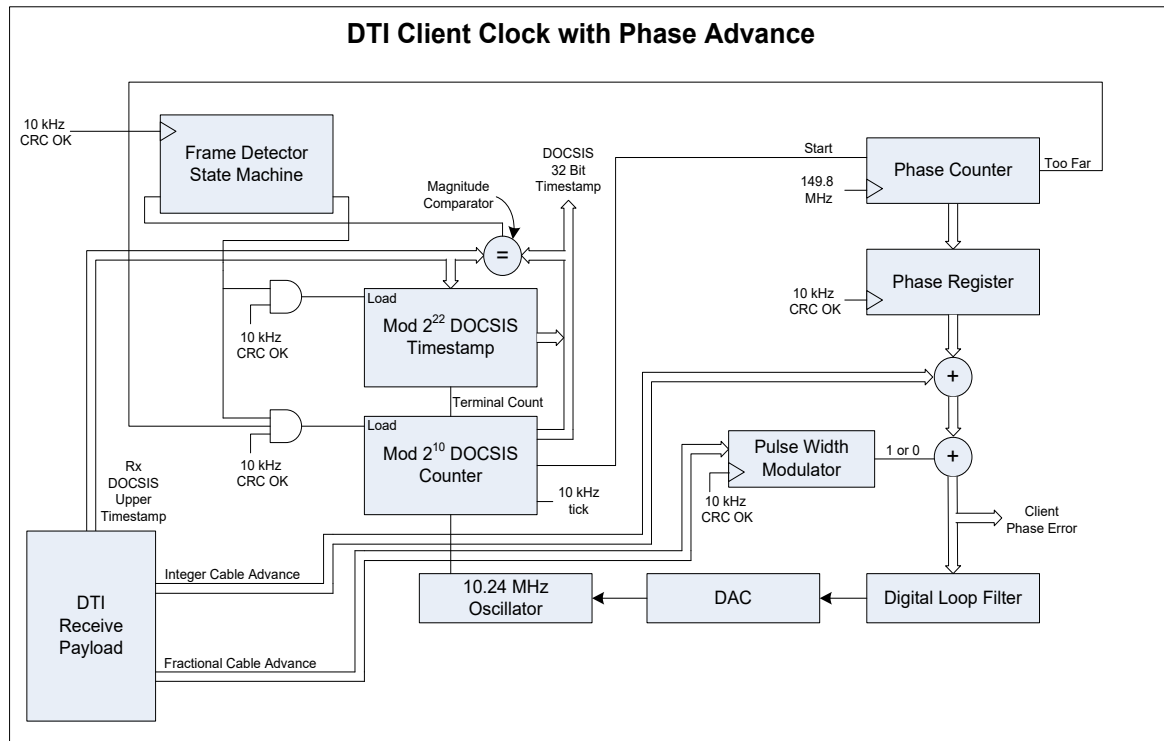


Figure II-4 - DTI Clock Processor

The client clock is built around a type II digital phase locked loop. Its frequency is steered to achieve phase alignment of its local 10 kHz frame clock with respect to the 10 kHz CRC OK signal from the DTI client frame processor. The phase alignment is advanced by the cable advance value. The phase locked loop needs to be tolerant of missing clocks since any data error will prevent the 10 kHz CRC OK from being asserted in the DTI frame processor.

The client clock should use a local mod 1024 DOCSIS Counter to produce the lower part of the DOCSIS timestamp (the local 10 kHz frame clock), to create a DTI clock-based 10 kHz to run the DPLL phase comparator. This counter utilizes the 10.24 MHz local oscillator and the phase of the counter is controlled by the clock loop. The count value of this mod 1024 counter can be loaded if excessive phase errors exist in the clock loop to accelerate the initial lock time of the loop.

The phase counter generates the digital phase word that controls the loop. The phase can be a signed straight binary counter that is loaded with a negative value by a decoded value off of the mod 1024 counter. The preload value should be such that the phase counter passes through 0 when mod 1024 counter rolls over to 0. If the phase counter saturates, it generates the "too far" signal that is used to adjust the mod 1024 counter.

The 10 kHz CRC OK signal from the DTI frame processor is used to strobe the phase counter value into the phase register. This will cause the phase register to hold the uncorrected phase error in the loop. To correct for the cable

delay the integer portion of the server cable advance, received from the payload in the frame receiver, is added to the phase counter register to create a corrected phase error value. This will cause the DPLL to lock with a phase register value that is the negative cable advance value and thus will advance the clock by that amount.

The fractional cable advance, which is a fraction of one of the 149.8 MHz clock cycles, is achieved by pulse-width modulating a 1-bit value that is added to the phase register. The pulse-width modulator creates a pulse-width that is the fractional cable advance/256 duty-cycle. The pulse-width modulator is clocked at 10 kHz since this is the phase update rate of the DPLL.

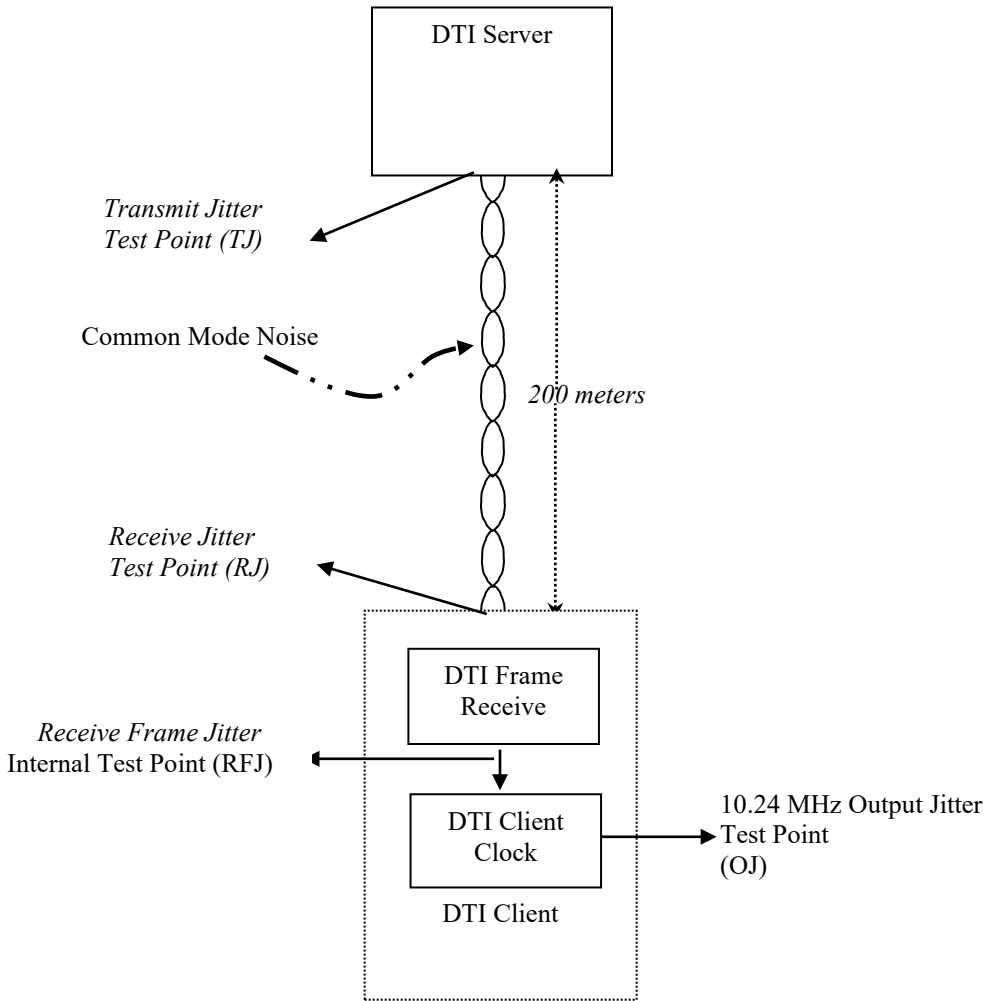
The corrected phase error is filtered by the digital loop filter and sent to an A to D converter to steer the voltage controlled 10.24 MHz oscillator. The corrected phase error is also sent to the DTI Client Frame processor block to be added to the payload to be sent back to the server.

The upper 22 bit DOCSIS timestamp value, received in the DTI server payload, is compared against the mod  $2^{22}$  DOCSIS timestamp counter and is checked at the 10 kHz CRC OK rate. A frame detector state machine checks this to determine if the two are in agreement, and if not, the local DOCSIS timestamp counter will be synchronized to the value received from the DTI server. The mod  $2^{22}$  DOCSIS timestamp counter is only loaded if the frame detector state machine indicates an out of frame condition. The upper  $2^{22}$  DOCSIS timestamp is concatenated with the mod 1024 counter to produce the complete  $2^{32}$  DOCSIS timestamp.



## Appendix III DTI Jitter Budget

The diagram below shows a reference model for the jitter budget analysis:



**Figure III-1 - DTI Jitter Budget Reference Model**

### III.1 Model Description

The model characterized the accumulated jitter in the forward path (from the DTI server to the DTI client). The reverse path is not included as its impact is limited to establishing cable advance.

The output of the DTI server is a Manchester encoded frame. The frame includes the preamble, payload and CRC and is transmitted at a 10 kHz rate. The physical layer characteristics are specified in section 5 of the proposed standard. The principle specifications are:

- Peak differential symbol voltage: 2.2 to 2.6 V
- Common mode source noise: <50 mV
- 100 Ohm differential impedance

The DTI standard supports an all ones test signal mode at the output. The Transmit Jitter (TJ) test point is at the DTI server RJ-45 connector.

The next component is the transport. Section 5 stipulates that the transport is UTP category 5E cable (or better) with a maximum distance limit of 200 meters. The DTI transmission is half duplex (ping-pong) and is the only wire pair active in the cable to minimize interference and optimize delay compensation.

Section 5.4.5 specifies the common mode rejection requirements. The induced common mode noise level is 15 V (the same as 10BT in 802.3). Both the differential noise level in voltage and edge jitter is specified.

The mapping from common mode noise to edge jitter is modeled in two steps. First the differential noise level is established based on the following:

- The model assumes a white noise common mode source at the 15 V peak level.
- The minimum Common Mode Rejection Ratio (CMRR) is specified for both the cable and the terminating transformers (35 dB).

The second step is to map the amplitude noise level to the jitter level based on the minimum slew rate of the receive signal over the cable.

This jitter performance is testable at the input DTI client RJ45 connector. The Receive Jitter (RJ) is modeled as the power sum of the transmit jitter and common mode induced jitter in the transport.

The DTI frame receive process recovers the receive 10 kHz frame rate from the incoming bursts. The recovery process utilizes the preamble for alignment. The vendor is free to implement the frame recovery process parameters in any method consisting with meeting the output performance objectives. The model assumes a digital PLL based on the high frequency ~149 MHz local clock used for cable advance. The bandwidth of this recovery PLL is assumed to be a maximum of one full frame or less. The Receive Frame Jitter (RFJ) is therefore an internal test point that is included in the model to capture the jitter filtering aspects of the frame recovery process.

Finally, the DTI client clock is modeled in steady state as PLL. The reference input to the PLL is the 10 kHz receive frame process included the 149 MHz digital quantization noise.

The local oscillator is assumed to be the DTI minimum client oscillator. The loop bandwidth is assumed to be 1 Hz for this analysis.

## **III.2 Analysis**

### **III.2.1 Transmit Jitter Specification**

The high frequency jitter >10 Hz is specified to be less than 50 ps RMS. This jitter allowance accommodates the expected noise in digital drive circuitry required in a DTI server. The transmit jitter is modeled as a white noise source.

### **III.2.2 Receive Jitter Analysis**

The receive jitter is the power sum of the transmit jitter and the common mode noise related jitter. It can be viewed in the time domain as the delay variation of the receive eye-pattern. The common mode component is bounded by physical layer requirements.

The common mode induced edge jitter bound over 200 meters of cable is less than 2.005 ns RMS.

The total receive jitter is the power sum which is dominated by the common mode effect (2.01 ns RMS).

**III.2.3 Receive Frame Jitter Analysis**

The edge jitter on the eye-pattern is filtered in the frame recovery process. Assuming a preamble based filtering approach the effective noise reduction factor prior to aliasing into the 10 kHz frame rate is 11.3<sup>14</sup>. The resulting Receive Frame Jitter is then 177 ps RMS.

**III.2.4 10.24 MHz Output Jitter**

The output jitter is the power sum of two components:

The jitter power of the DTI minimum oscillator in the band of interest.

The residual jitter power of the quantized 10 kHz frame signal in the band of interest

The output jitter given a 1 Hz loop filter bandwidth with no enhanced jitter suppression techniques is:

**Table III-1 - DTI 10.24 MHz Output Jitter Performance**

HPF Jitter Cutoff Freq Hz <sup>15</sup>	Integrate Jitter in Band ps RMS			
	Residual 10 kHz input jitter <sup>16</sup>	DTI Minimum Oscillator <sup>17</sup>	DTI 10.24 MHz Jitter	DOCSIS 2.0 Spec <sup>18</sup>
10	6.8	4.02	7.9	88
100	3.44	1.55	3.8	73
1000	0.72	0.46	0.86	70

The broadband jitter (10 Hz to ½ master clock) is shown in the first row. The 88 ps RMS budget is calculated from the DOCSIS 2.0 (6.3.8) by power summing the individual bands. In contrast, the DTI 10.24 MHz broadband jitter is 7.9 ps RMS. Recall that this jitter is under the conditions of maximum cable distance, maximum common mode noise and minimum DTI client oscillator. There is a better than 20 dB margin compared to the required master clock performance.

If we consider the direct use of the 10.24 MHz DTI signal to support RF carrier operation (with minimal additional jitter filtering), the more important aspect may be the high frequency jitter, as the carrier recovery will track the lower frequency components.

The third row shows the integrated jitter above 1 kHz. The 10.24 MHz output directly from the client has less than 1 ps (0.86) of jitter in this band. This is better than a 38 dB margin compared to the required master clock performance in this band.

<sup>14</sup> Based on a particular implementation of frame clock recovery other methods may have more or less noise suppression.

<sup>15</sup> High pass from cutoff to ½ master clock rate (5.12MHz)

<sup>16</sup> Assuming worst case common mode noise over maximum 200 meter cable distance

<sup>17</sup> Typical measured performance of a compliant VCTCXO

<sup>18</sup> The numbers derived from the 50ps, 20ps, 50ps, 50ps integrated phase noise requirements for the master clock in DOCSIS 2.0.

## Appendix IV Symbol Clock Synchronization

In synchronous (S-CDMA) operation, the Downstream Symbol Clock is locked to the 10.24 MHz Master Clock using a 16 bit integer M/N ratio.

For the standard DOCSIS and EuroDOCSIS symbol rates, the recommended M/N ratios are shown in Table IV-1.

**Table IV-1 - Symbol Clock Synchronization**

Modulation	Symbol Rate	M/N
EuroDOCSIS	6.952	869/1280
DOCSIS 64-QAM	5.056941	401/812
DOCSIS 256-QAM	5.360537	78/149

In an M-CMTS installation, the DTI Server-Client distributes a phase aligned Master Clock across multiple EQAMs. It is also desirable to control the phase of the Symbol Clocks generated by each EQAM. This can be achieved without additional signaling across the DTI interface.

The DOCSIS Timestamp Counter is a 32 bit counter clocked by the 10.24 Master Clock, which rolls over every 4294967296 clock cycles. Unfortunately, none of the required ‘n’ values divide evenly into this number.

The DTI Client also provides a ‘gpssec’ value. ‘gpssec’ is a 32-bit timestamp that is incremented every second, or more accurately, every 10240000 cycles of the 10.24 MHz Master Clock.

It is convenient to declare that a positive zero-crossing of all symbol clocks occurred at the Master Clock edge where the ‘gpssec’ counter was set to zero (January 6, 1980). Given this, it is possible, given the ‘gpssec’ counter value, to determine how many Master Clock cycles remain before the next symbol clock positive zero-crossing, as follows:

$$\text{Master Clock cycles remaining} = (\text{‘gpssec’} \times 10240000) \text{ MOD ‘N’}$$

For example, if the ‘gpssec’ value has just updated to 123456, then the number of Master Clock cycles remaining before a positive zero-crossing of the DOCSIS 256-QAM Symbol Clock (M/N=78/149) is given by:

$$\text{Master Clock cycles remaining} = (123456 \times 10240000) \% 149 = 135.$$

This means that the DOCSIS 256-QAM Symbol will experience a positive zero-crossing in exactly 135 Master Clock cycles. This remainder value (135) can be used to ‘count-down’ to a reset pulse, or alternatively, can be forced directly into the divisor register of the NCO used to generate the symbol clock.

Given that the set of N values is limited, specific formulas can be used which simplify the math. Specifically for the three values of N in the table:

$$N=1280 \quad \text{Master Clock cycles remaining} = 0.$$

$$N=812 \quad \text{Master Clock cycles remaining} = ((\text{gpssec} + \text{rollover}) \text{ MOD } 203) * 680 \text{ MOD } 812$$

$$N=149 \quad \text{Master Clock cycles remaining} = ((\text{gpssec} + \text{rollover}) \text{ MOD } 149) * 124 \text{ MOD } 149$$

Note that for N=1280 there are exactly 8000 divide by N cycles in a second so that the cycle remaining is constant and zero.

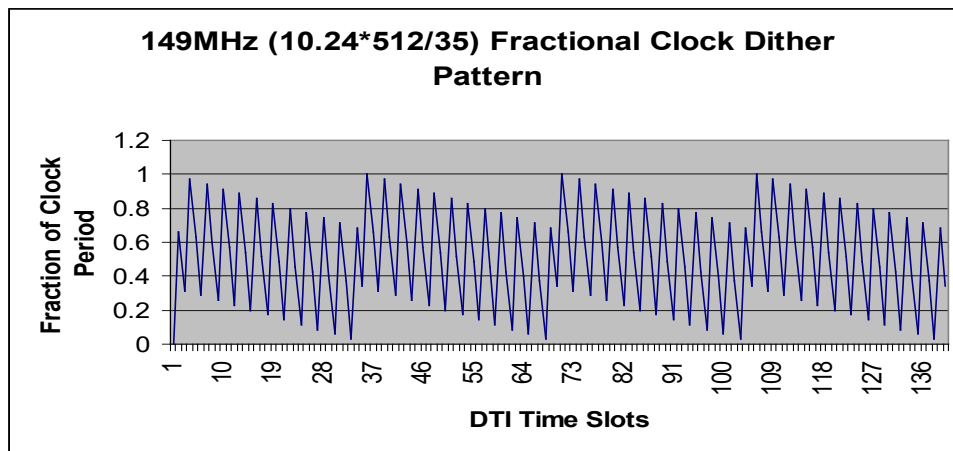
Since this ‘n’ value does not divide evenly into  $2^{32} \times 10240000$ , this mechanism could cause a single cycle glitch when ‘gpssec’ rolls over. This will occur once every 136 years, and will occur first in the year 2116. To prevent this effect a rollover term is included. The rollover should be added starting at the rollover event in the year 2116. The rollover term is 74 for N=812 and 129 for N=149.

## Appendix V DTI High Speed Clock Considerations

One key aspect of the DTI timing protocol is the adoption of the nominal 149.8 MHz DTI high-speed clock. The precise definition of the DTI high-speed clock is:  $10.24 \text{ MHz} * 512/35$ . The DTI high-speed clock permits an all-digital integrated implementation of phase measurement functions with low residual jitter and phase bias. Phase measurement is required in the client to precisely lock the client 10 kHz clock to the receive clock with cable advance applied. The server requires precise phase measurement to perform calculation of cable advance. Also, digital receive clock recovery may utilize the high-speed clock to support all digital implementations. The selection of the DTI high-speed clock is a trade-off between output jitter and bias.

For example, one simple approach is to select a direct multiple of the master clock. A high-speed clock of 153.6 MHz is exactly  $10.24 \text{ MHz} * 15$ . One could consider a counter measuring the delta phase between the 10 kHz receive clock and the local client 10 kHz clock.<sup>19</sup> In the absence of noise, the measurement result would be static for delay changes up to the nominal 6.5 ns of the high-speed clock. While the resulting jitter would be low the alignment error could be the full 6.5 ns. Real world noise and bias variation would yield 6.5 ns transient activity and degrade operation, especially S-CDMA precise ranging.

The selection of the non-integer ( $512/35 = 14.62857\dots$ ) multiple generates a fractional clock dither pattern that repeats every 35 DTI timeslots as shown in Figure V-1.



**Figure V-1 - Fractional Clock Dither Pattern**

This dither pattern permits discrimination of phase shifts error to 190 ps resolution while constraining pattern jitter so that the client PLL can effectively filter it. Consider the same counter as above measuring the delta phase between the receive signal and local 10kHz signal, but now using the 149.8 MHz DTI high-speed clock. If the phase offset is 191 ps, the least significant bit of the counter will toggle as shown in Figure V-2. The least significant bit toggles to "one" once every 35 DTI timeslots. This pattern will be averaged over the client PLL bandwidth, and for a 10 Hz PLL, the average phase is 187 ps with less than 5 ps residual jitter. If we increase the phase offset to 381 ps, the least significant bit will now toggle as shown in Figure V-3. Now the pattern is two "one" pulses every 35 DTI timeslots with the average output phase measurement of 374 ps and less than 5 ps of residual jitter.

<sup>19</sup> This is just one example where jitter and bias issues could arise. The same issues need to be consider for digital clock recovery of the DTI receive signal as well as cable delay measurement.

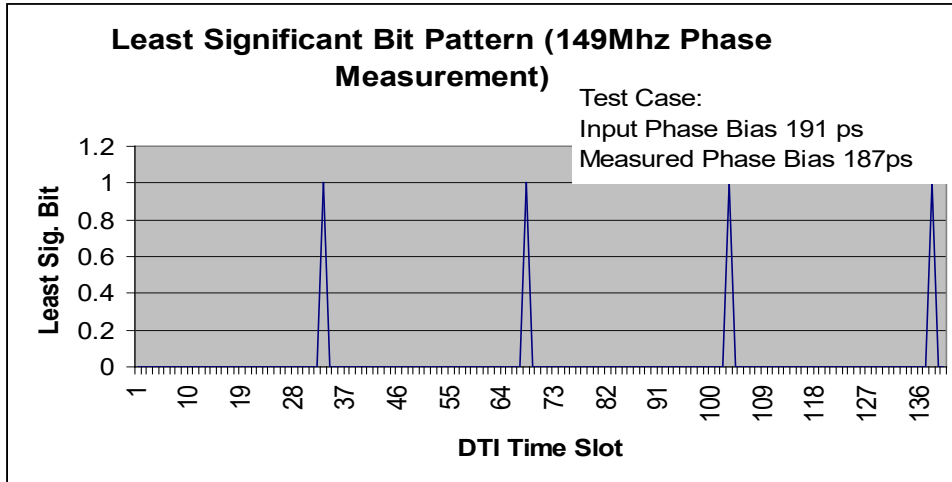


Figure V-2 - Phase Measurement Dither Pattern 191 ps Offset

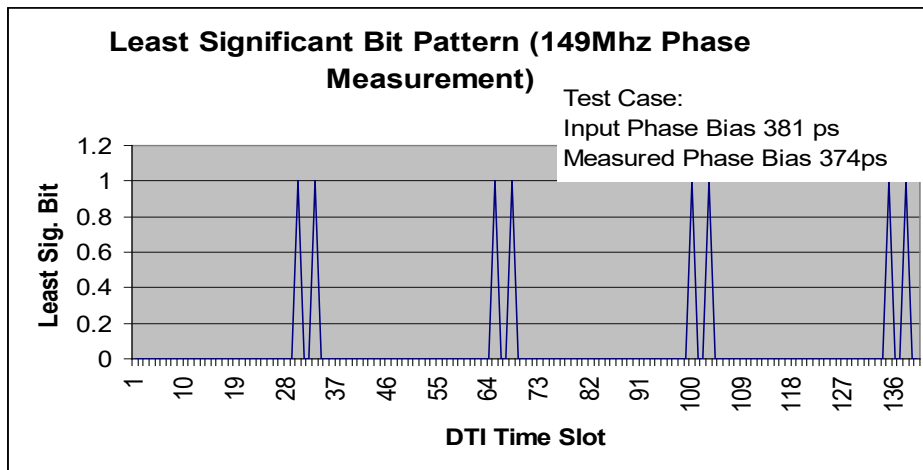


Figure V-3 - Phase Measurement Dither Pattern 381 ps Offset

## Appendix VI M-CMTS System Issues Regarding Use of DTI Conveyed Timebase by an M-CMTS Entity Incorporating a DTI Client

The DTI specification specifies all that is required for the DTI interface between a DTI server and a DTI client. A DTI client resides in an M-CMTS entity, such as the M-CMTS Core, the EQAM, and the future consideration of an Upstream Receiver. The functionality about how the master clock timebase is delivered from the DTI server to DTI clients and the timing accuracy performance of this delivery is carefully specified in this document. However, there then is the topic of how an M-CMTS entity incorporating a DTI client will make use of this DTI delivered timebase and what delay is allowed in the use of this delivered timebase.

For example, the replicated master timebase in the EQAM is used for timestamps placed into inserted or corrected SYNC messages. For example, at the M-CMTS Core, a replicated master timebase is needed for scheduling and creating MAP messages and knowing when to send MAP messages in the downstream and when to send MAP information to the Upstream Receiver. At an Upstream Receiver, a replicated master timebase is needed to correctly receive upstream burst transmissions and to perform the timing adjustment operation in Initial or Station Maintenance.

Specific requirements regarding how readily an M-CMTS Core or EQAM utilize the DTI conveyed time are covered in [DEPI]. In this Appendix, the discussion focuses on some of the system issues related to how readily the DTI conveyed time is used by an M-CMTS entity.

### VI.1 Example Issues

Consider if a M-CMTS Core, EQAM, or the future Upstream Receiver, introduces unreasonable delays in how it applies the DTI conveyed timebase such that there is a significant difference in applied time at each of these M-CMTS entities. Various issues could arise, such as:

1. The size of Initial Maintenance opportunities provided by the M-CMTS Core may not be sufficient such that some CMs may have trouble ranging or be prevented completely from successfully ranging.

Today, an Initial Maintenance opportunity is typically sized to accommodate the maximum round trip propagation delay on the HFC plant, the transmission time of the Ranging Request message itself, and other latencies incurred in the HFC round trip path that a CM does not offset prior to Initial Maintenance. Any delay that a M-CMTS Core applies to the DTI conveyed time that is not compensated in some other means manifests as additional delay in the round trip for the ranging process. If this lag of the applied time by the M-CMTS Core compared to the DTI conveyed time is not accounted for in the sizing of the Initial Maintenance opportunities, these opportunities may not be large enough. However, if there are significant differences in notion of utilized time at the Burst Receiver compared to the M-CMTS Core and Initial Maintenance opportunities are made large enough, there is lost bandwidth due to the oversizing of the Initial Maintenance opportunities.

2. The M-CMTS Core may have to employ much larger MAP advance times (the time in advance that the M-CMTS Core sends a MAP message downstream such that it can arrive at any CM in sufficient time such that the MAP can be used). This could significantly reduce throughput performance by enlarging the round trip delay.

This was not a concern with an integrated CMTS since all the functions that required consultation of the master timebase were integrated on one line card or within a single chassis. That is not the case with an M-CMTS architecture.

3. The EQAM may timestamp SYNC messages with a greatly delayed version which may artificially enlarge the round trip time and thereby reduce throughput performance.
4. Large differences in notions of time between M-CMTS Core and a separate Upstream Receiver unit may be unworkable from a system standpoint.

For example, if the MAP information being sent from the M-CMTS Core to the Upstream Receiver is past the point of time-relevance, i.e., the MAP information is late, the burst demodulator in the Upstream Receiver will not be able to properly receive any upstream burst.

On the other hand, if the MAP information being sent from the M-CMTS Core to the Upstream Receiver is very early, the Upstream Receiver is required to have the capability to buffer a very significant amount of MAP information. This would have to be specified in a specification for the Upstream Receiver. Otherwise, significantly different notions of time at the M-CMTS Core and Upstream Receiver could result in the required amount of buffering for MAP information to exceed the amount available at the Upstream Receiver.

## VI.2 Solution

The goal of traditional DOCSIS system synchronization was to convey the CMTS master timebase to CMs. The components of the CMTS, all on one linecard or in one chassis, shared a common notion of time or notions of time that were offset by extremely small amounts of time. The goal of DTI system synchronization is to provide different M-CMTS entities the same notion of master timebase, which then can be conveyed to CMs through the mechanisms that have been defined in traditional DOCSIS (timestamps in SYNC messages for TDMA and lock to the downstream symbol rate clock in concert with SYNC messages and the S-CDMA Timestamp Snapshot for S-CDMA).

Each M-CMTS entity receives the master timebase through its DTI client. However, how readily that timebase is utilized is the key issue discussed here. The main issue is the amount of divergence in the notion of utilized time at M-CMTS Core versus that at the EQAM versus that at the Upstream Receiver. Hypothetically, if each M-CMTS entity took the DTI conveyed timebase and used a zero-delayed version of it, all M-CMTS entities synchronized by the same DTI server have identical notions of time. Likewise, if each M-CMTS entity took the DTI conveyed timebase and essentially used a delayed version of it by exactly 500 ms, all M-CMTS entities synchronized by the same DTI server would have identical notions of time. Problems arise when different M-CMTS entities apply different latencies with regards to applying the DTI conveyed time.

The analogy of the "clock on the wall" can be used to illustrate the issue. The DTI interface operates to convey the master clock timebase to a DTI client in an M-CMTS entity. The M-CMTS entity will replicate the master clock in some way such that there is now a "clock on the wall" that is the reference for when to do certain tasks. The issue is to appropriately bound how much later the "clock on the wall" is relative to the master clock time conveyed by DTI.

So, in the M-CMTS specifications, the selected solution is to bound the amount of time in which the timebase utilized by an M-CMTS entity lags the DTI conveyed time. For the EQAM and M-CMTS Core, these bounds are found in [DEPI]. As mentioned before, the Upstream Receiver, when it is completely specified as an M-CMTS entity, would have to have a similar bound applied to it.

