

SAE Aerospace An SAE International Group	AEROSPACE STANDARD	SAE AS6003	
		Issued	2011-02
TTP Communication Protocol			

RATIONALE

TTP is used in a variety of aerospace applications (for example, Boeing 787 power generation systems and environmental controls, cabin pressure systems for Airbus A380, Aermacchi M-346 FADEC...) and continues to attract significant cross-industry attention for commercial and defense applications.

The SAE standardization of TTP:

- acts as an integration risk reduction mechanism
- ensures compatible physical implementations
- enables common test/maintenance equipment
- leverages industry investments
- ensures openness and enables multiple component and tool suppliers

and therefore reduces the overall cost and risk of applying this technology.

The SAE standardization based on TTP specification protects long-term system design investments, enables development of the COTS ecosystem, and minimizes sourcing risks for OEMs, integrators and system suppliers.

INTRODUCTION

The Time-Triggered Protocol (TTP) is a real-time communication protocol for the interconnection of electronic modules of distributed fault-tolerant real-time systems. TTP is a core technology for fault-tolerant distributed embedded computing and enables design of deterministic embedded computing platforms for critical systems. As such, it contains communication protocol capability for data exchange among nodes, but also provides higher level services for design of reusable generic platforms with robust partitioning among functions. This document specifies the structure of the TTP protocol, services and mechanisms on an abstract level without implementation-specific details.

TTP's features support the design of systems with a high degree of dependability, safety, availability, reliability, maintainability, and reduced system complexity.

Some methods contained in the document are protected by granted and pending patents, and underlie SAE standard IP policy.

SAE Technical Standards Board Rules provide that: "This report is published by SAE to advance the state of technical and engineering sciences. The use of this report is entirely voluntary, and its applicability and suitability for any particular use, including any patent infringement arising therefrom, is the sole responsibility of the user."

SAE reviews each technical report at least every five years at which time it may be reaffirmed, revised, or cancelled. SAE invites your written comments and suggestions.

Copyright © 2011 SAE International

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of SAE.

TO PLACE A DOCUMENT ORDER: Tel: 877-606-7323 (inside USA and Canada)
 Tel: +1 724-776-4970 (outside USA)
 Fax: 724-776-0790
 Email: CustomerService@sae.org
 http://www.sae.org

SAE WEB ADDRESS:

Full document details: SAE AS6003

**SAE values your input. To provide feedback
 on this Technical Report, please visit
<http://www.sae.org/technical/standards/AS6003>**

Documents
 Document Sharing Hub

TABLE OF CONTENTS

1.	SCOPE.....	4
1.1	Purpose.....	4
1.2	Application.....	4
1.3	Interpretation.....	4
2.	REFERENCES.....	4
2.1	SAE Publications.....	4
2.2	Other Publications.....	5
2.3	Structure of the Document.....	5
2.4	Typographic Conventions.....	5
3.	OVERVIEW OF A TTP-BASED SYSTEM.....	6
3.1	Structure of a TTP Network.....	6
3.2	Structure of an Electronic Module (Node).....	6
3.2.1	Communication Layer.....	7
3.2.1.1	Central Characteristics of the Time-Triggered Protocol.....	7
3.2.1.1.1	Protocol Services.....	7
3.2.1.1.2	Autonomous Operation of the TTP Controller.....	8
3.2.1.1.3	Global Time Base.....	8
3.2.1.1.4	TDMA Bus Access.....	8
3.2.1.1.5	Configuration Data.....	9
3.2.1.2	Overview of the Host Interface.....	9
3.2.1.3	Overview of the Physical Layer Interface.....	9
3.2.2	Application Layer - Time-Triggered Architecture.....	9
3.2.2.1	Application Layer Synchronous to Communication Layer.....	9
3.2.2.2	Application Layer Asynchronous to Communication Layer.....	9
4.	HOST INTERFACE.....	10
5.	DATA LINK LAYER.....	11
5.1	TDMA Scheme.....	11
5.2	TTP Frame Layout.....	12
5.2.1	Frame Header.....	12
5.2.1.1	Frame Type.....	13
5.2.1.2	Mode Change Request.....	13
5.2.2	C-State.....	13
5.2.3	Frame CRC.....	14
5.3	Frame Types.....	15
5.3.1	N-Frame.....	15
5.3.2	I-Frame.....	16
5.3.3	Cold Start Frame.....	17
5.3.4	X-Frame.....	17
5.4	Frame Status.....	18
5.4.1	Null-Frame Reception Status.....	20
5.4.2	Valid Frame Reception Status.....	20
5.4.3	Invalid Frame Reception Status.....	20
5.4.4	Incorrect Frame Reception Status.....	20
5.4.5	Tentative Frame Reception Status.....	20
5.4.6	Correct Frame Reception Status.....	20
5.4.7	Cluster Mode Violation.....	20
5.4.8	Slot Status.....	20
6.	PHYSICAL LAYER – INTERFACE.....	21

7.	FAULT-TOLERANT DISTRIBUTED CLOCK IN TTP	22
7.1	Timing Parameters	22
7.1.1	Microtick	23
7.1.2	Macrotick	23
7.1.3	Precision	23
7.2	Synchronization	24
7.2.1	Action Time and Exchange of Timing Information in a TTP Network	24
7.2.2	Calculation of the Correction Term	26
7.2.3	Correcting the Local Clock	27
8.	PROTOCOL SERVICES	28
8.1	Startup	28
8.1.1	Integration	28
8.1.2	Cold-start	29
8.1.2.1	Timeouts	30
8.1.2.1.1	Startup Timeout	30
8.1.2.1.2	Listen Timeout	30
8.1.2.1.3	Cold Start Timeout	30
8.1.2.2	Big Bang	31
8.2	Membership	32
8.2.1	Membership Vector	33
8.2.2	Multiplexed Slots	33
8.3	Acknowledgment	34
8.3.1	Acknowledgement Algorithm	34
8.3.2	Acknowledgement Algorithm – State 1	34
8.3.3	Acknowledgement Algorithm – State 2	35
8.3.4	Sequence of Checks	36
8.4	Clique Detection	37
8.4.1	Clique Error	37
8.4.2	Communication System Blackout	37
8.5	Host/Controller Life-sign	37
8.6	Cluster Modes	38
8.6.1	Distribution Phase	38
8.6.2	Acceptance Phase	38
8.6.3	Execution Phase	39
9.	PROTOCOL STATES	39
9.1	Protocol States and State Transitions	39
9.1.1	TTP Controller – Protocol State Overview	39
9.1.2	TTP controller – Protocol State Transitions	40
9.2	Operation in TTP Protocol States	42
9.2.1	Protocol Variables	42
9.2.2	Freeze State	43
9.2.3	Init State	44
9.2.4	Listen State	44
9.2.5	Cold Start State	46
9.2.6	Active and Passive State	47
9.2.6.1	Pre-send Phase	47
9.2.6.2	Transmission Phase	50
9.2.6.3	Post-receive Phase	50
9.2.6.4	Idle Phase	51
APPENDIX A	GLOSSARY ACRONYMS	52
APPENDIX B	COMPATIBILITY REQUIREMENTS FOR EXISTING TTP CONTROLLER IMPLEMENTATION	53

1. SCOPE

This SAE Aerospace Standard (AS) establishes the specification for TTP communication protocol functionality as a core networking component for design of synchronous, time-triggered distributed real-time communication networks.

This document is referred to as the “base” specification, containing the generic specification of TTP communication protocol functionality, and frame formats, communication services and protocol state machines with transitions, TTP node operation, TTP network operation, distributed clock synchronization and error detection on the network layer.

The details described in this standard enable interoperable TTP controller implementations.

1.1 Purpose

The purpose of this document is to standardize mechanisms and operation of the TTP protocol for safety-critical and mission-critical applications. The information herein will be used to assist in the design, fabrication, system integration, and obsolescence management of TTP-based systems.

This document is controlled and maintained by the SAE AS-2 committee with technical support from TTA-Group members and TTP users.

1.2 Application

TTP is well-suited for the design of deterministic system architectures with hard real-time behavior in safety and mission-critical applications. The application of TTP as a fieldbus is viable for fault-tolerant distributed systems, deterministic networks, and distributed control system platforms (e.g., flight controls, by-wire steering, environmental controls, smart sensor/actuator networks, distributed power generation, or landing gear). TTP supports design of time-triggered architectures and reusable generic platforms compliant with RTCA DO-297. TTP can also be used for backplane communication and design of modular aerospace system controls.

1.3 Interpretation

The following interpretations shall be placed upon these words, unless stated otherwise, where they are used in this document.

May: An allowed action.

Shall: A mandatory requirement.

Should: A recommended action.

Will: A declaration of intent.

2. REFERENCES

The following publications form a part of this document to the extent specified herein. The latest issue of SAE publications shall apply. The applicable issue of other publications shall be the issue in effect on the date of the purchase order. In the event of conflict between the text of this document and references cited herein, the text of this document takes precedence. Nothing in this document, however, supersedes applicable laws and regulations unless a specific exemption has been obtained.

2.1 SAE Publications

Available from SAE International, 400 Commonwealth Drive, Warrendale, PA 15096-0001, Tel: 877-606-7323 (inside USA and Canada) or 724-776-4970 (outside USA), www.sae.org.

940140 Fault Management in the Time Triggered Protocol (TTP)

2.2 Other Publications

[Kop87] H. Kopetz and W. Ochsenreiter. Clock Synchronization in Distributed Real-Time Systems. *IEEE Transactions on Computers*, 36(8):933–940, Aug. 1987.

[Pal97] R. Pallierer and T. M. Galla. Multiplexed SRUs in TTP – Concepts and Approaches. Research Report 19/97, Institut für Technische Informatik, Technische Universität Wien, Vienna, Austria, October 1997. Confidential.

[Pfe99] Holger Pfeifer, Detlef Schwier, and Friedrich W. Von Henke. Formal Verification for Time-Triggered Clock Synchronization. In Charles B. Weinstock and John Rushby (eds.), Editors, *Dependable Computing for Critical Applications 7*, volume 12 of *Dependable Computing and Fault-Tolerant Systems*, pages 207–226. IEEE Computer Society, January 1999.

[Pfe00] Holger Pfeifer. Formal Verification of the TTP Group Membership Algorithm. In Tommaso Bolognesi and Diego Latella, Editors, Formal Methods for Distributed System Development Proceedings of FORTE XIII / PSTV XX 2000, pages 3-18, Pisa, Italy, October 2000. Kluwer Academic Publishers.

2.3 Structure of the Document

This specification document is self-contained and gives a bottom-up approach to the functionality of the TTP protocol.

It is divided into the following chapters:

- Chapter 3 provides an overview of the TTP network architecture.
- Chapter 4 defines the minimum characteristics and requirements for the host interface.
- Chapter 5 describes and defines the requirements on Data Link Layer level for the frame formats and the bus access scheme.
- Chapter 6 defines the minimum characteristics and requirements for the Physical Layer.
- Chapter 7 describes the timing and synchronization algorithm used by TTP.
- Chapter 8 describes and defines the mechanisms for the protocol service layer.
- Chapter 9 defines the operation of the TTP controller in the context of the TTP state machine model.

2.4 Typographic Conventions

The typographic conventions for this documentation are as follows:

Element	Typographic format
Protocol variables / text emphasis	<i>lower case italic</i>
Protocol states	<i>Upper case italic</i>
Header within a section	bold

3. OVERVIEW OF A TTP-BASED SYSTEM

The TTP controller is the core component of the time-triggered architecture, the conceptual time-triggered-based system design approach. This section describes general aspects of TTP-based network design, as well as system design aspects of an electronic module or *node*.

3.1 Structure of a TTP Network

A TTP-based communication network (TTP network) consists of a set of electronic modules that are typically connected by dual channels – *channel 0* and *channel 1* – which comprise a *TTP-bus*. A TTP network – as shown in Figure 1 – is called a *cluster*. The basic building block of a cluster is the electronic module (node). Depending on the system architecture and depending on the system requirements, e.g. with respect to fault tolerance, communication between nodes can be established on a single channel (channel 0 or channel 1) or redundant channels (channel 0 and channel 1) basis.

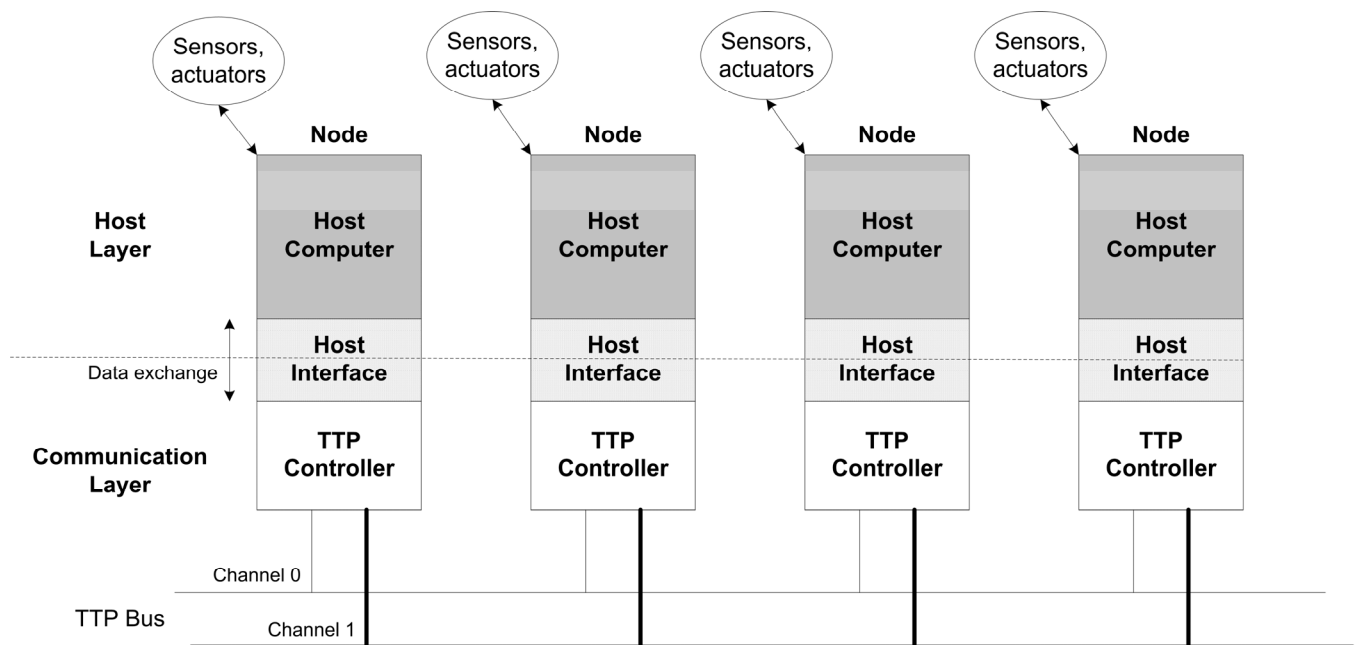


FIGURE 1 - TYPICAL TTP NETWORK

3.2 Structure of an Electronic Module (Node)

An electronic module has the general structure shown in Figure 2. A node comprises of a host (typically a CPU executing an operating system which manages the application software, with memory and the access to the I/O subsystem, or an FPGA) and a TTP controller.

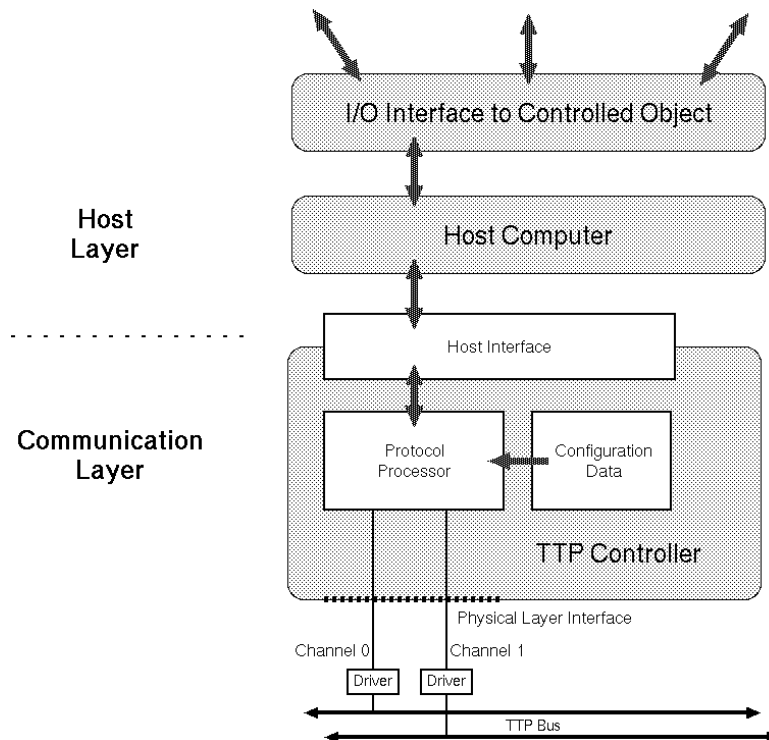


FIGURE 2 - STRUCTURE OF A TTP NODE

A TTP controller shall process the execution of the protocol services based on a statically defined set of communication requirements (shown as configuration data in Figure 2). Furthermore the Protocol Processor shall use a well-defined interface to the host and to the TTP bus to provide reliable operation of the services defined by the TTP (Protocol services).

From an abstract perspective, each node consists of two layers – the *communication layer*, which shall ensure reliable time-triggered communication, and the *host layer*, which provides/consumes application data to/from the communication layer. The host and the communication layer shall run as *autonomous subsystems* on a node, but depending on the chosen system design approach, both subsystems shall be able to run in a synchronous or asynchronous manner to each other.

3.2.1 Communication Layer

The time-triggered protocol defines all the processes in the communication layer. The central characteristics of the time-triggered protocol are the interfaces of the communication layer, which are described in the next section.

3.2.1.1 Central Characteristics of the Time-Triggered Protocol

3.2.1.1.1 Protocol Services

The protocol services are needed to reliably perform and to control the exchange of data between the nodes in a cluster. The protocol services can be divided into the following main categories:

- *Communication services* shall guarantee reliable data transmission, the *startup* of a cluster, the *reintegration* of nodes, the *acknowledgment*, fault-tolerant *clock synchronization* and handling of cluster *mode changes*.
- *Safety services* shall support error detection including *node membership*, the *clique detection* algorithm and *Host/Controller life-sign* service.

The protocol services are defined in Chapter 8.

3.2.1.1.2 Autonomous Operation of the TTP Controller

The TTP controller shall function as an autonomous subsystem. That means that the operations that the TTP controller is able to perform at any given time are defined by the state machine model in chapter 9. But the TTP controller requires a running host for operation and some initialization data and control data as required to perform the individual TTP services (Chapter 8).

3.2.1.1.3 Global Time Base

A TTP Controller shall provide a time base to trigger the protocol events (the execution of the TTP services, frame reception and frame transmission). This time base shall have a granularity in macrotick units (see section 7.1.1). The global time base shall ensure synchronous protocol execution and cluster operation between all nodes in the cluster within a configurable time interval – the so-called *precision*. Synchronous operation ensures that all communication layer-related operations on all nodes happen in a coordinated manner, to enable each node to have bus access without collision, and allow reliable data exchange with predictable latency and minimal jitter. Synchronized operation is achieved (between a minimum of two nodes) immediately after a successful startup – as mentioned in section 8.1.

3.2.1.1.4 TDMA Bus Access

When synchronous cluster operation mode is accomplished, meaning that communication was initiated and synchronization could be achieved between at least two active nodes (nodes that are allowed to transmit data on the Bus), a TDMA-based strategy is used by all nodes in the cluster to access the bus. This means that every active node has a certain amount of reserved bandwidth – a *node slot* – to write data to the bus. Every TTP controller shall receive all data transmissions available on the bus.

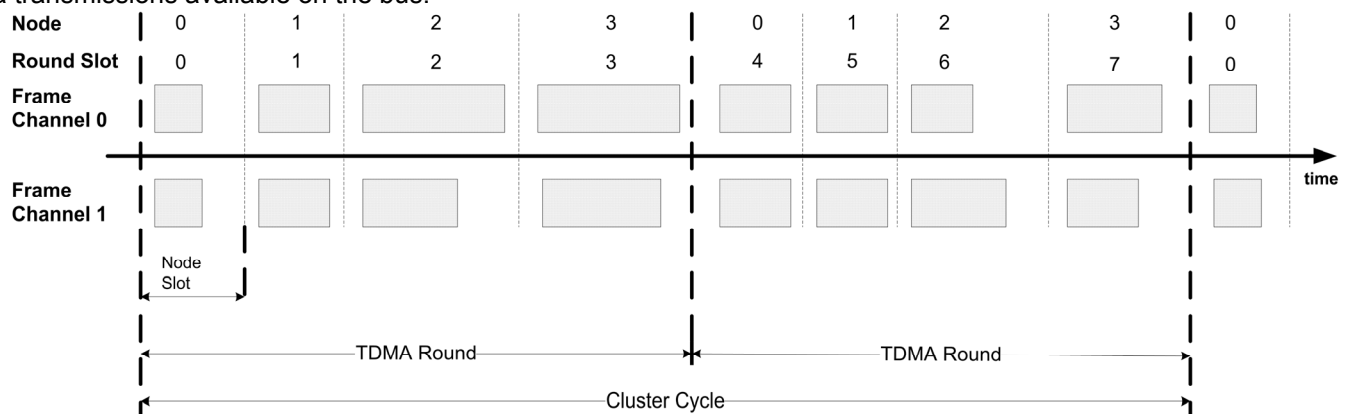


FIGURE 3 - MEDIA ACCESS SCHEME

Each active node shall periodically utilize the full transmission capacity of the bus in its dedicated node slot, by sending *TTP frames* on both channels. The frames on channel 0 and channel 1 may differ in their frame type (as defined in section 5.4), data length and data content.

The periodic sequence of node slots is called a *TDMA round*. The duration of each slot as well as the sending sequence of the different nodes in the cluster shall be equal and fixed in all TDMA rounds and for all cluster modes. The only possible variation between two TDMA rounds is the amount of application data transmitted in the slots (the frame length) and in case of multiplexed slots the assigned sending node in a particular round. The pattern of periodically recurring TDMA rounds is called a *cluster cycle*.

3.2.1.1.5 Configuration Data

A TTP controller shall perform according to application-specific communication and timing requirements which could be defined during a cluster design process. This set of information is referred to as configuration data and should be the output of the cluster design. The configuration data shall define what a TTP controller shall do at each point in time during a cluster cycle. Each node has its own unique slot position in the TDMA round; therefore each node shall be configured with an individual set of communication schedules, respective to its positions in the TDMA round. Some configuration data applies to all nodes in the cluster (e.g., bus speed, TDMA round duration, number of rounds). There is also node-specific configuration data (e.g., the time instant when a frame arrives / is to be sent). A TTP controller shall process the definitions in the configuration data based on the progress of a global time base. Depending on the TTP controller implementation, this set of configuration data may be realized, e.g., directly by the chip design, or made available as a data array in the memory. The set of configuration data is summarized in Table B3.

3.2.1.2 Overview of the Host Interface

The host interface is needed to enable data exchange between the host layer and the communication layer either in synchronous or asynchronous operation mode as described in section 3.2.2. The host interface is required to ensure interoperability – simultaneous random access – between the host and the TTP controller for frame, status and control data. The host interface specification in Chapter 4 specifies the minimum requirements

3.2.1.3 Overview of the Physical Layer Interface

The physical layer interface is needed to guarantee bus-compatibility – interoperability on the network level – between different TTP controllers. Bus-compatibility means that two different TTP controller types – fulfilling the same bus-compatibility specification – may exchange data and are fully compatible in all aspects of TTP communication given that they also share the same physical layer. This physical layer interface is characterized by the connection between a TTP controller and the line driver as described in Chapter 6, as well as on data link level with respect to frame format and bus access scheme as described in Chapter 5. SAE standardization projects AS6003/1, AS6003/2 and other future physical layer standards cover detailed description of physical layers for TTP communication protocol.

3.2.2 Application Layer - Time-Triggered Architecture

The host layer as depicted in Figure 2 is responsible for the execution of local or distributed application algorithms, and handles the interface to its environment (Sensors, Actuators) and to the TTP communication layer. The host layer may perform in a *synchronous* or *asynchronous* manner to the network-wide synchronized communication layer.

3.2.2.1 Application Layer Synchronous to Communication Layer

From the application perspective, if the host subsystem (the algorithms performed on the application layer) is aligned (synchronized) to the global time, data transmission between the host and the TTP controller always occurs at the right time and without data access conflicts, as defined at design time. This approach – *TTA approach* (time-triggered architecture approach) – helps to reduce system complexity, enhances resource use and improves host subsystem abstraction and independence from underlying networking and hardware details.

The TTA approach takes advantage of the availability of the global time to precisely and unambiguously specify the interfaces among the nodes, to simplify communication, to establish system state consistency, to perform continuous and prompt error detection, and to enable seamless system integration for distributed real-time applications.

3.2.2.2 Application Layer Asynchronous to Communication Layer

From the application perspective communication follows a deterministic schedule, but applications can receive data with some latency and delay which is influenced by application startup activity and its alignment with the communication schedule. The application operates in a “local” or event-driven fashion while the communication is strictly deterministic and not influenced by the progress of the application. This means that coordination of data transfers (e.g., prevention of host interface access conflicts) in time should be done by the application.

4. HOST INTERFACE

The host interface shall guarantee collision-free data exchange between the host and the TTP controller. The TTP controller shall make available received application data and continuously updated TTP controller and network related status data. The host shall make available updated message, control, and configuration data to the TTP controller.

Three different categories of data shall be exchanged between the host and the TTP controller:

- a. **Message data** – The host interface shall provide means for the exchange of application data produced by the host (application data for transmission on the TTP bus) or received by the TTP controller (application data acquired from the TTP bus). Furthermore the host interface shall provide means to pass the evaluated frame status after each TTP frame reception or transmission (as described in section 5.4) to the host. The host should deal with application data, only if the status of the received frame is valid.
- b. **Status data** – The TTP controller shall be able to inform the host about the status of the protocol execution by maintaining the status data continuously.
- c. **Control data** – The host shall be able to provide setup information or control and maintenance data to the TTP controller. This is needed to enable protocol execution, valid frame transmission and to initiate operation mode changes.

Table 1 summarizes the different kind of data needed for the execution of the communication and protocol related operations.

TABLE 1 - HOST INTERFACE SPECIFICATION

Host interface – data	Category	Reference
Acknowledgment error	Status	Section 8.3
Clique error	Status	Section 8.4
System blackout error	Status	Section 8.4
Mode violation error	Status	Section 8.6
Initialization error	Status	Section 9.2.3
TTP controller life-sign	Status	Section 8.5
Host life-sign	Control	Section 8.5
Time Startup Field	Control	Section 9.2.5
Protocol version	Status	Section 9.2.3
Mode change request	Control	Section 8.6
Protocol state	Status	Section 9
TTP Controller On	Control / Status	Section 9.2.2
Time Startup Field	Control	Section 9.2.5
Valid C-state	Status	Section 9.2.5

5. DATA LINK LAYER

The data link layer deals with aspects of the TDMA-based bus access scheme, frame format and the calculation of the frame status. Sharing the same data link layer is mandatory for the compatibility of TTP controllers interacting (transmitting/receiving frames) within the same TTP cluster.

5.1 TDMA Scheme

The principle of the bus access strategy in synchronous operation mode – the TDMA scheme – is described in section 3.2.1. Thus, as long as each node uses only its own statically assigned node slot, collision-free access to the bus can be ensured. This can be guaranteed by the existence of a global time base and the static bus access schedule (given by the configuration data) that is known by all nodes. The *round slot* describes a logical slot in the cluster cycle – as illustrated in Figure 3.

A slot shall begin with *the pre-send phase* (PSP) and end at the beginning of the PSP of the next node slot. Figure 4 shows the partitions of a slot. The starting time of the *transmission phase* (TP), meaning the point in time when the frames are planned to be sent or received, is called *action time* (AT). The action time shall be perceived on all synchronized nodes of the cluster as the same point in time (within the precision interval as defined in section 7.1.3). The interval between two successive TPs is called *inter-frame gap* (IFG), consisting of the pre-send phase *post-receive phase* (PRP) and the *idle time*.

The PSP is required to protocol services for preparing the TTP controller to send or receive data (as described in section 9.2.6.1). During the PRP the TTP controller shall process the frame data and shall execute protocol services based on received data (as described in section 9.2.6.3). Since the slot length not only depends on the amount of data sent on the bus and the bus speed, but also on the definition of round length, the number of defined nodes (slots), usually there is a certain amount of time left which shall be called *idle phase*. During the idle phase the protocol operation shall be halted and shall wait for the start of the PSP of the next slot. The number of slots in a cluster cycle, the duration of the node slot, the action time relative to be begin of the cluster cycle and the duration of the TP, shall be defined in the configuration data.

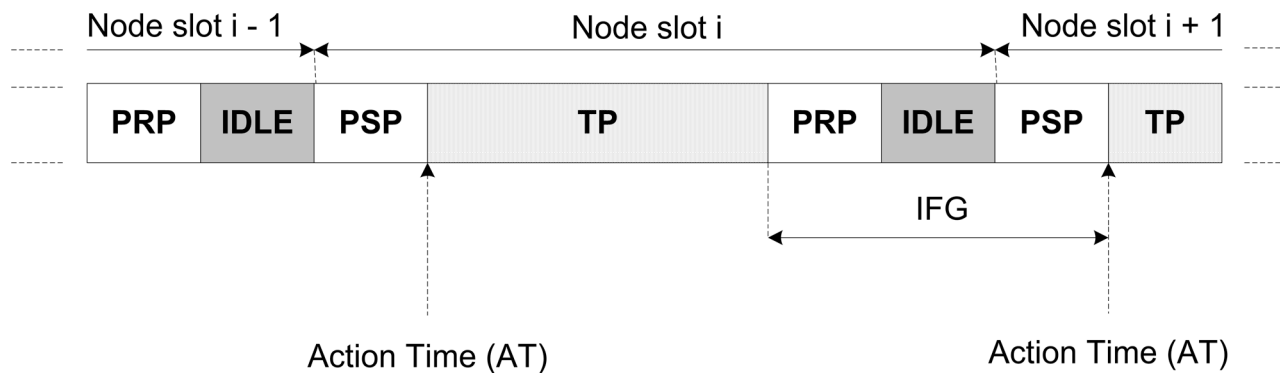


FIGURE 4 - SLOT TIMING

5.2 TTP Frame Layout

A TTP frame is transmitted in the transmission phase and is composed of protocol information and application data. Figure 5 illustrates the general frame layout.

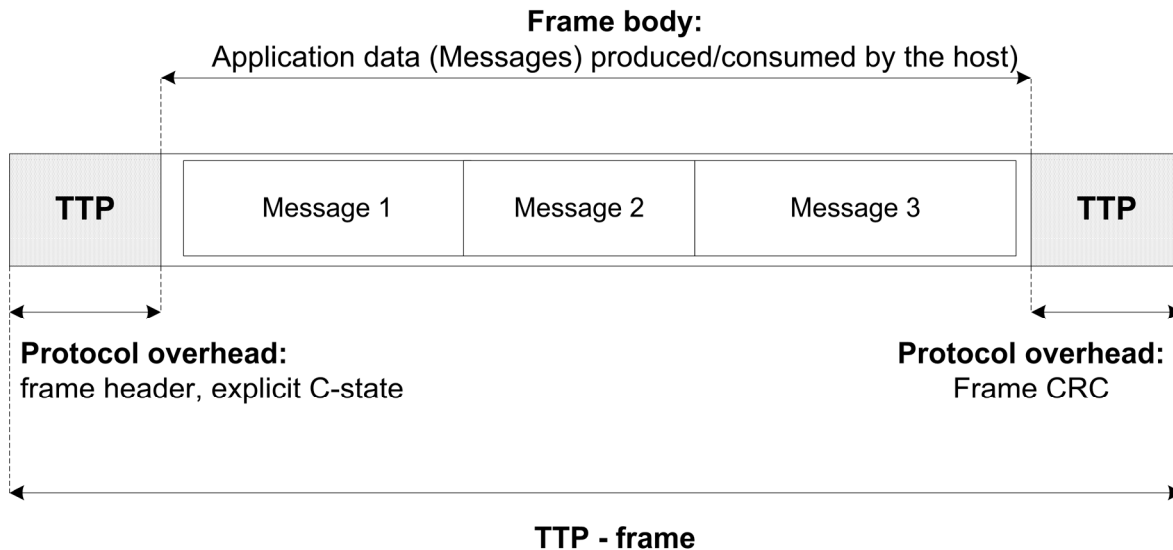


FIGURE 5 - TTP FRAME LAYOUT

TTP Frame: A TTP frame consists of a frame body (containing the application data) and the protocol overhead (necessary for protocol operation).

Frame body: The frame body is the main part of a TTP frame and carries application data produced or consumed by the host; it may be subdivided into parts that are interpreted by the host application called *messages*. Application data shall have no semantics for the TTP controller and therefore no impact on the protocol operation. The maximum size of the frame body is defined in Table B1. In case of a reception, the TTP controller stores the application data in the host interface, and in case of transmission it fetches it from the host interface.

Protocol overhead: The protocol overhead is necessary for the operation of the TTP protocol and shall consist of the frame header (as described in section 5.2.1), the CRC value (as described in section 5.2.3) and depending on the frame type of the explicit C-State (as described in section 5.2.2).

5.2.1 Frame Header

The header is a 4-bit field shall contain the frame type in the LSB and the mode change request field in the three MSBs.

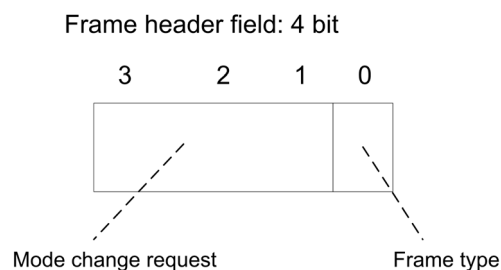


FIGURE 6 - FRAME HEADER

5.2.1.1 Frame Type

The frame type field shall contain information about how the C-state is incorporated in the frame. This information is needed for the interpretation of a received frame by a TTP controller. The following convention shall be provided for the frame type field:

- The frame type shall be set to one (bit set) to indicate the frame has an explicit C-state as described in section 5.3.
- The frame type shall be set to zero (bit reset) to indicate the frame has an implicit C-state as described in section 5.3.

5.2.1.2 Mode Change Request

The deferred mode change request field shall represent the value of a requested mode change as defined in section 8.6. The value range for this bit field is defined in Table B1.

5.2.2 C-State

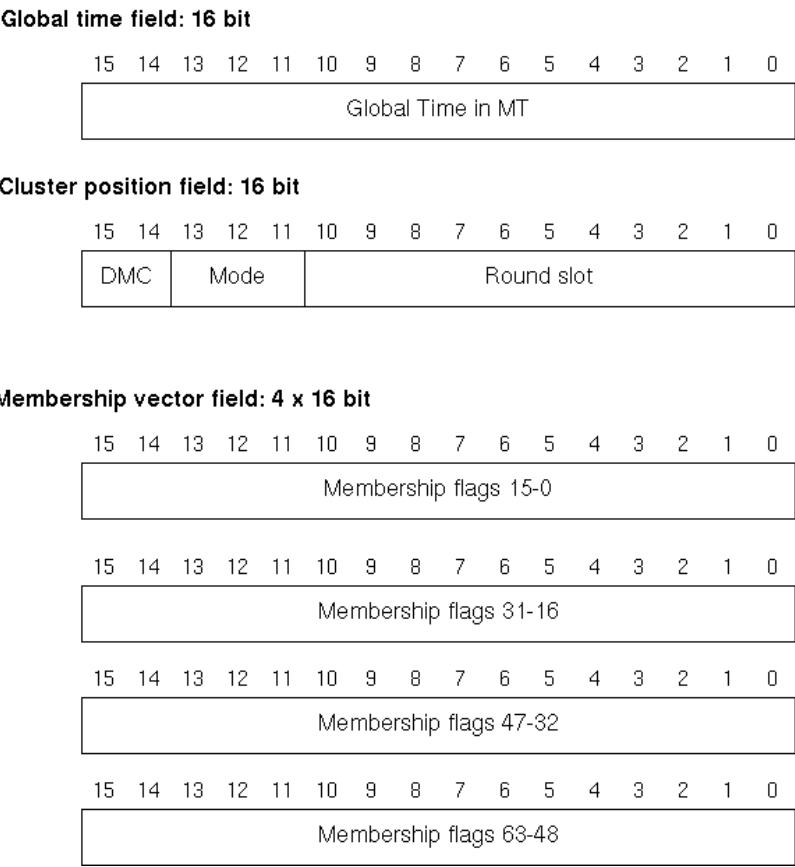


FIGURE 7 - C-STATE LAYOUT

The controller state (C-state) is a collection of state variables describing the TTP controller’s internal view of the cluster state with respect to the global time, cluster position and the membership status. The C-state shall have a length of 12 bytes as depicted in Figure 7. The local C-state shall be updated by each TTP controller in each round slot. The local view of the C-state shall be checked for agreement with the C-state received with each frame transmission. Therefore, the C-state represents a global view of the cluster and shall agree with the calculated C-states of the other nodes in a correctly synchronized TTP cluster. The C-state shall be exchanged by each frame transmission either explicitly or implicitly as described in section 5.2.3. A C-state disagreement between a node and the majority of the other nodes shall mark this node as faulty as described in section 8.2. Thus the exchange of the C-state between the nodes is one of the basic concepts of the protocol to support fault detection and system health monitoring. Furthermore, the C-state contains the minimum amount of data to enable integrating nodes to participate in an already ongoing cluster communication.

The local C-state shall be updated by the TTP controller in each slot in each TDMA round according to the following data definition:

- **Global time** field shall contain the macrotick value of the start time of the transmission phase (action time) in the current slot. The global time field shall be updated during the pre-send phase with respect to the current value of the global time and with respect to the current round slot position.
- **Cluster position** field shall contain information about the current round slot position, the current cluster mode and pending deferred mode change (DMC)
 - The deferred pending mode changes (DMC) field shall contain the value of requested successor cluster mode as defined in section 8.6.2.
 - The cluster mode field shall contain the value of the current active cluster mode the TTP controller is operating in. The TTP controller shall update the Cluster mode field as defined in section 8.6.3.
 - The round slot position field shall contain the current cluster position information. This field identifies the round slot-specific dataset in the configuration data. The round slot position shall be zero-based, meaning that round slot number 0 is the first round slot in the cluster cycle. The TTP controller shall update the round slot position field during the PSP.
- **Membership vector** shall contain a consistent view of the operational status of all nodes in the cluster as described in section 8.2. A TTP controller shall update its membership vector in the post-receive phase based on its perceived status of the received frame. A sending node shall update its membership flag according to the acknowledgement algorithm (defined in section 8.2).

5.2.3 Frame CRC

The CRC check shall be done by a TTP controller to verify the correctness of any received frame. For each frame reception, a TTP controller shall compare the CRC value of the received frame with the computed CRC value over the received frame data (as described in detail in section 5.3). The CRC is used for error detection, not for error correction. CRC, C-state, and frame type are all related to each other as shown in Figure 8. The use of implicit C-States in frames improves data transmission performance, because no bandwidth is used for the C-State transmission. Explicit C-States in frames are required for the integration process and for cluster startup, because a TTP controller can only integrate on a received frame if it can extract the C-State directly. A TTP controller shall send one of these frames in its sending slot, which is defined in the configuration data. Depending on the explicit or implicit availability of the C-state in a transmitted frame, the receivers of the frame shall perform the CRC check differently as described in the following:

- **Explicit C-state:** The TTP controller shall compare its local C-state with the C-state of the received frame. This enables the TTP controller to check in case of an erroneous received frame, whether its cause is related to a transmission error or a C-state disagreement.
- **Implicit C-state:** In the case of an implicit C-state, the C-state is not part of the frame's data. The TTP controller shall use its local C-state together with received frame data for the CRC calculation. In case of an erroneous received frame, the TTP controller cannot distinguish between a transmission error and the occurrence of a C-state disagreement.

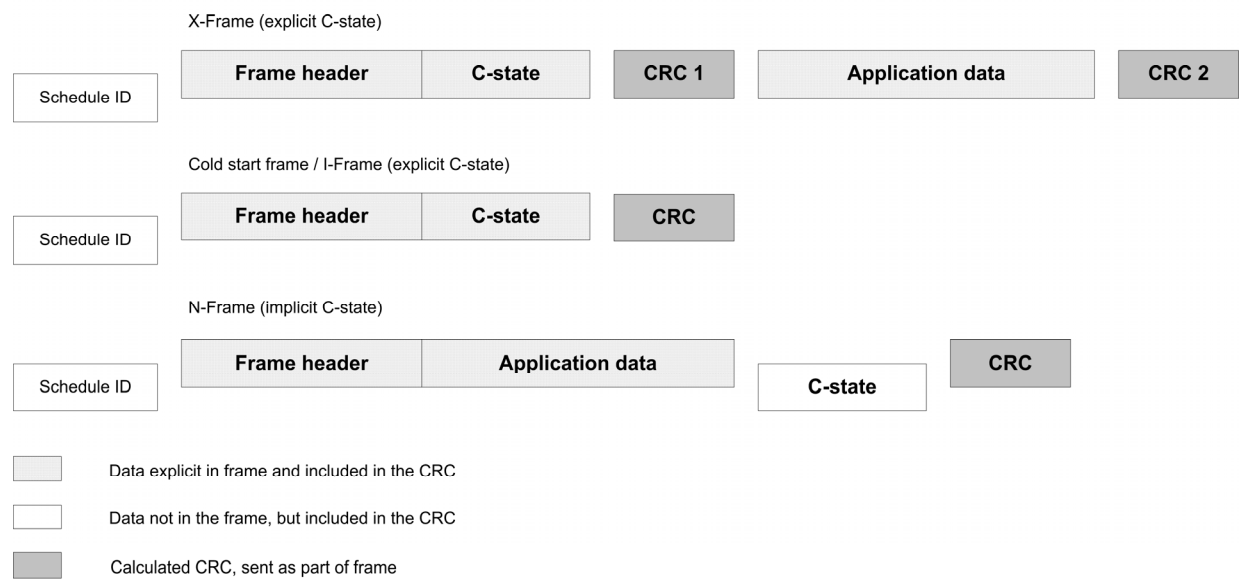


FIGURE 8 - IMPLICIT VERSUS EXPLICIT C-STATE

The CRC calculation on each node shall be initialized with the same CRC seed value, which shall be contained in the configuration data as defined in Table B1. This seed value shall relate to the specific set configuration data (cluster design) and shall be different for the CRC calculation for each channel. This will prevent the TTP controller from integrating on TTP communication based on a different cluster design which contains incompatible configuration data and crossed-over channels are detected immediately

The TTP controller shall calculate the CRC based on a generator polynomial defined in Table B1 that ensures that the defined Hamming distance is fulfilled for the maximum frame length. Table B1 specifies the parameters needed for CRC calculation which shall be fulfilled by any TTP controller.

5.3 Frame Types

TTP frames are coherent sequences of data, transmitted as broadcasts from a TTP controller on both channels within their slot.

TTP frames can be classified into:

- Frames with implicit C-State (N-frames)
- Frames with explicit C-State (I-frames, X-frames)
- Cold start frames

The frame type for all frames expected by a TTP controller to be sent or received shall be defined in configuration data, for each channel and for each slot.

5.3.1 N-Frame

The *N-frame (normal frame)* shall contain only application data. The C-state of the sender shall be implicitly included in the CRC value. Figure 9 specifies the frame layout.

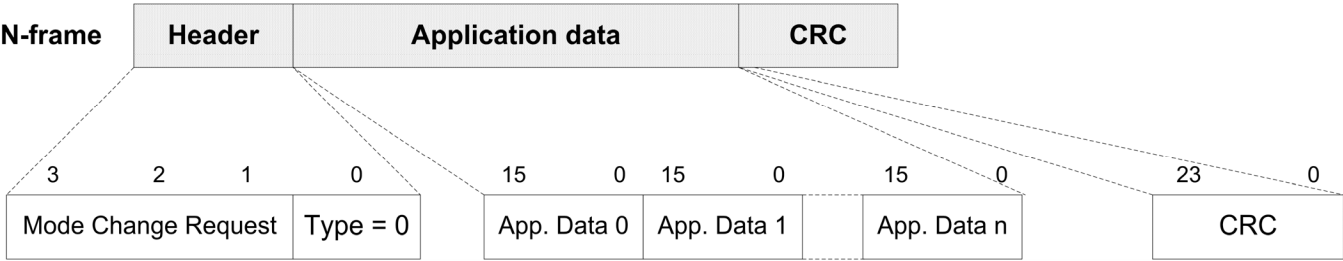


FIGURE 9 - N-FRAME

Because of the absence of an explicit C-state, nodes cannot integrate on an N-frame. N-frames have the least transmission overhead and therefore the best data efficiency.

An N-frame shall contain the following parts:

- Frame type identifier set to 0 (implicit C-state)
- Mode change request according to the mode change request field in the host interface (see section 8.6)
- Application data frame size: 1 to maximum data length (see Table B1), configurable per round slot
- CRC calculated over the frame header, the application data and the C-state

5.3.2 I-Frame

An *I-frame (initialization frame)* shall contain the explicit C-state of the sender, shall not contain application data, and is used for integration purposes only. Figure 10 specifies the frame layout of an I-frame.

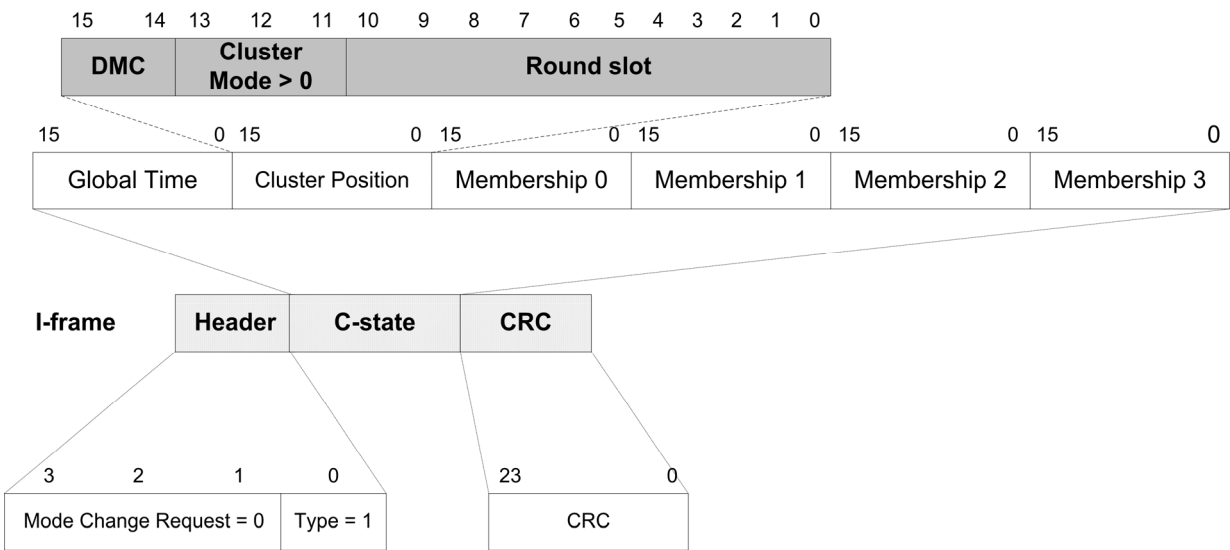


FIGURE 10 - I-FRAME

An I-frame shall contain the following parts:

- Frame type identifier set to 1 (explicit C-State)
- Mode change request according to the mode change request field in the host interface (see section 8.6)
- C-state

- CRC calculated over the frame header and the C-state.

The TTP controller shall transmit an I-frame in its sending slot whenever no application is scheduled for a round slot.

5.3.3 Cold Start Frame

The cold start frame is needed to initiate cluster communication. It has the format of an I-frame but with special value settings. Cold start frames shall be sent only by nodes that are allowed to initiate cluster communication. Figure 11 specifies the frame layout of a cold start frame.

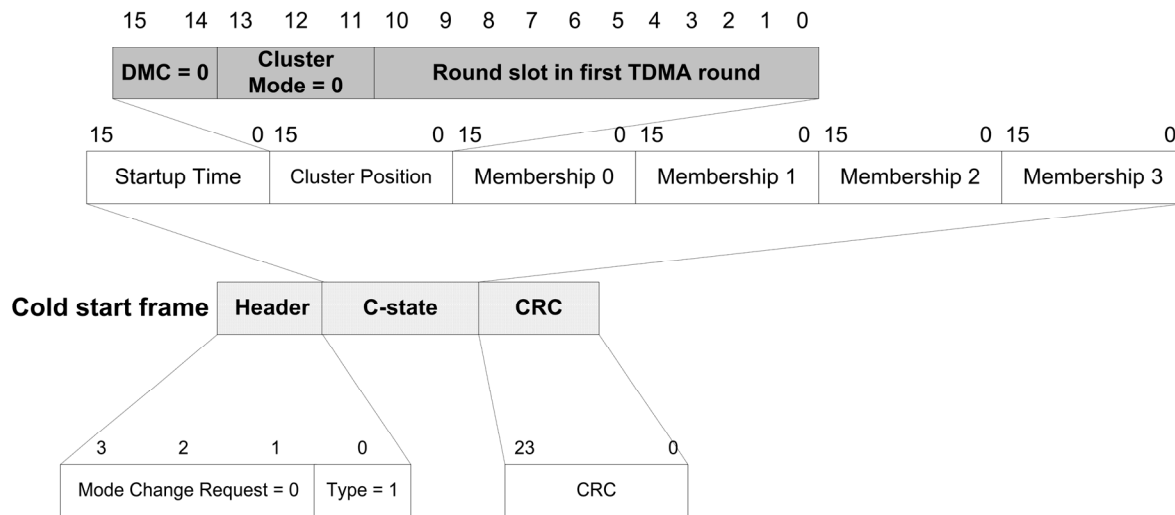


FIGURE 11 - COLD START FRAME

A cold start frame shall contain the following parts:

- Frame type identifier set to explicit C-State
- Mode change request set to 'no request' (see Table B1)
- The C-State's Global Time field set to the contents of the *Time Startup* field of the host interface.
- The Round slot position field set to the round slot number of the nodes sending slot (the first TDMA round).
- The C-State's Cluster Mode field set to the *cold start ID*. The cold start ID is a dedicated and unique cluster mode that is different from any cluster mode identifiers that may be used during schedule-synchronous operation as well as from startup mode.
- The C-State's Deferred Pending Mode Change (DMC) field set to 0x0.
- Membership flags except that of the cold start node shall be set to 0 in the C-State's membership vector field.
- CRC calculated over the frame header and the C-state.

5.3.4 X-Frame

An *X-frame (extended frame)* shall contain both the explicit C-state of the sender and application data.

Figure 12 specifies the frame layout of an X-frame.