Chapter 10
Real-Time Operating Systems

Overview

- Introduction to TTP/C
- Protocol objectives
- Protocol layers
- Smallest replaceable units and fault tolerant units
- Structure of the communication node interface (CNI)
- Membership service and CRC computation
- Introduction to TTP/A for field bus applications

10.1 Task Management

Time-Triggered Protocols (TTP) are designed for hard real-time systems.
There are two versions:

- TTP/C for fault-tolerant hard real-time systems
- TTP/A for low cost applications (e.g. field bus applications)

Example application: Pinoth snow groomer

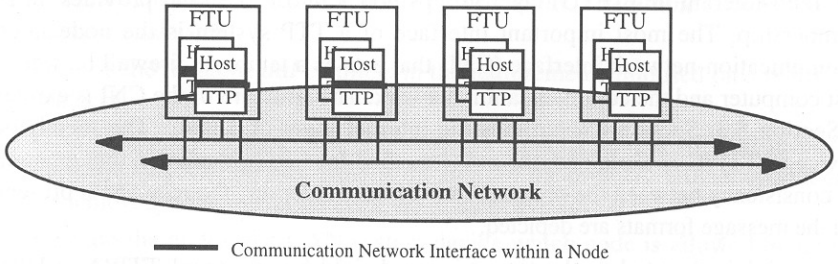
Example application: Boeing 787 Dreamliner

Protocol Objectives

1. Message transport with low latency and minimal jitter
2. Support of Composability
3. Provision of a fault-tolerant membership service
4. Fault-tolerant clock synchronization
5. Distributed redundancy management
6. Minimal overhead, both in message length and in number of messages
7. Scalability to high data rates, efficient operation on twisted-pair and optical fiber

Structure of a TTP System

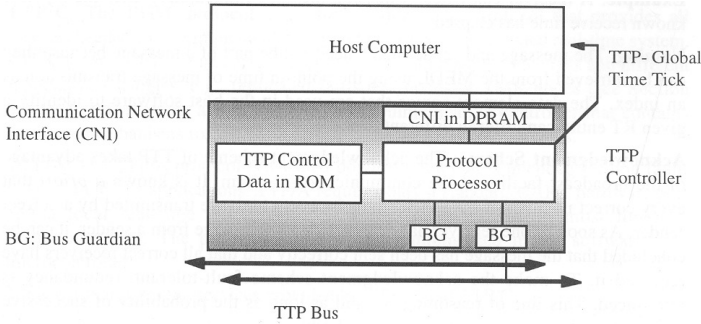
A TTP system consists of a cluster of fault-tolerant units (FTU's), each comprised of one or more nodes, and interconnected by a communication network.



A node is the smallest replaceable unit (SRU).

A node consists of two subsystems:

- the host computer
- the communication controller



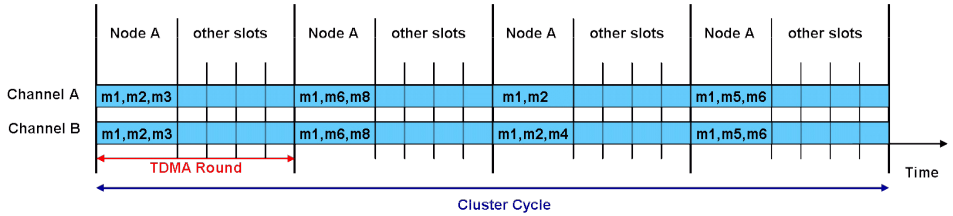
The communication network interface (CNI) is the node-internal interface between the communication controller and the host. The CNI is formed by a dual-ported RAM.
Data integrity passed between host and communication controller is ensured by a non-blocking write protocol.

The communication controller has a local memory to hold the message descriptor list (MEDL).

The MEDL determines

- at what point in time a node is allowed to send a message
- when a node can expect to receive a message

The MEDL has the size of one cluster cycle.



The Bus Guardians (BGs) are independent hardware devices. They monitor temporal access pattern of the controller to the replicated busses, and terminate controller operation in case of timing violations.

Design Rationale

TTP is a TDMA protocol. Every node sends a message on a shared communication channel during a predetermined, statically assigned time slot.

Composability:

The TTP controller operates autonomously. It is controlled by the MEDL inside the controller, and the fault-tolerant global time.

The CNI between the TTP controller and the host computer is fully specified in the value and temporal domain, which supports composability of an architecture.

An error in any one of the hosts cannot interfere with proper operation of the communication system, since no control signals cross the CNI. The MEDLs are inaccessible to the hosts.

Best use of a priori knowledge:

In a time-triggered architecture it is known at design time which node must send which message at what time. Thus, a receiver can detect missing messages, or bus guardians can prevent the effect of babbling clients.

Naming:

In a time-triggered architecture, messages can be uniquely identified by their time of transmission. Thus there is no need to transport message names along with the message. This increases data efficiency, particularly with short messages. It also improves composability, since there is no need to agree on message names across different hosts.

Acknowledgement scheme:

Every operational member of an ensemble hears every message transmitted on the communication channel. As soon as one receive acknowledges the message, it can be concluded that the message has been sent correctly, and that all correct receivers will have received the message.

Fail silence in the temporal domain:

TTP assumes that nodes support the file-silent abstraction in the temporal domain. This means a node either delivers a message at the correct time, or not at all.
This supports error confinement at the system level. The bus guardians implement the fail-silent mode for nodes by monitoring channel access and disconnection the channel from the controller in case of temporally erroneous access.
A membership service can detect the failure of a node with small latency.

Fail silence in the value domain:

Fail silence in the value domain must be implemented by the host. For this, the host must ensure by space and/or time redundancy that all internal failures of a host are detected before a non-detectable erroneous output message is transmitted.
Value failures at the communication level are detected by a CRC mechanism.

Design tradeoffs:

TTP has decided to limit bandwidth requirements on cost of processing requirement at the nodes. The rationale behind this decision is that bandwidth is more difficult to increase than processing power, which has steadily increased over the past years with advancements in VLSI technology.

Protocol Variants

TTP comes in two variants, a full version called TTP/C and a scaled-down version called TTP/A. At the CNI, the structure is compatible for both protocol versions.

TTP/C:

TTP/C provides all services needed for the implementation of a fault-tolerant distributed real-time system. TTP/C supports FTUs that comprise replicated communication channels and different replication strategies, e.g., replicated fail-silent nodes or TMR nodes.

TTP/C requires a specially designed communication controller containing hardware mechanisms for the implementation of the protocol functions.

TTP/A:

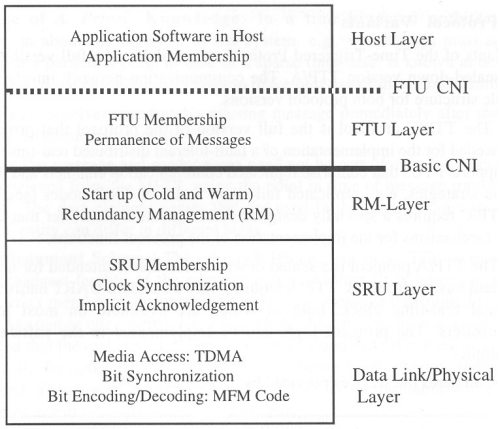
TTP/A is intended for non fault-tolerant field bus applications. It requires just a standard UART hardware port and a local real-time clock, typically available on most low-cost microcontrollers. The protocol logic is implemented in the software of a microcontroller.

The following table compares features for the two protocol variants.

Service	TTP/A	TTP/C
Clock synchronization	Central multimaster	Distributed, fault-tolerant
Mode switches	Yes	Yes
Communication error detection	Parity	16/24 bit CRC
Membership service	simple	full
External clock synchronization	yes	yes
Time-redundant transmission	yes	yes
Duplex nodes	no	yes
Duplex channels	no	yes
Redundancy management	no	yes
Shadow node	no	yes

10.2 Interprocess Communication

The TTP/C is organized in a set of conceptual protocol layers, which do not correspond to the OSI protocol layers. The Basic Communication Network Interface (CNI) typically marks the interface between purely software implemented protocol functionality and software/hardware related functionality.



Data Link/Physical Layer

This layer serves to exchange frames between nodes. It provides

- media access control
- bit synchronization
- bit encoding and bit decoding

Access scheme is time-division-multiple access (TDMA). Access is controlled by data stored in the MEDL of the TTP controller.

Bit synchronization and bit encoding/decoding uses Modified Frequency Modulation (MFM) code. In MFM, data bits are separated by clock bits; (x,y) encodes to (x, x NOR y, y).

Requirements on physical layer on twisted pair are less demanding than those of a CAN system (no bit arbitration required). Thus, a CAN twisted pair network can be used.

Today, data rates specified go up to 25 MBit/s.

SRU Layer

The single replaceable unit (SRU) layer provides the following services:

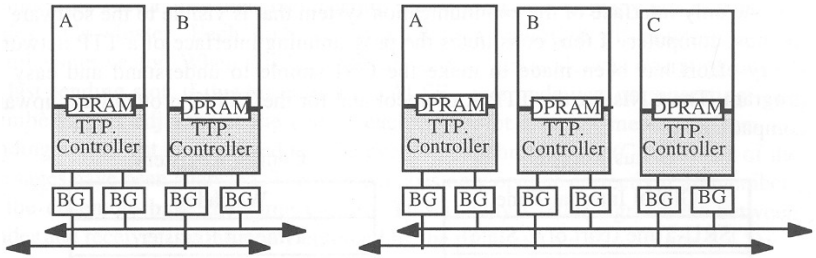
- Stores data fields of received frames in the memory area of the CNI DPRAM
- Establishes node membership
- Performs Byzantine-resilient clock synchronization by the fault-tolerant average algorithm
- Provides immediate and deferred mode change service to higher layers

Redundancy Management Layer (RM Layer)

The redundancy management layer provides the following services:

- Cold start of a TTP/C cluster. For this it uses the mode-change service of the SRU layer.
- Reintegration of a repaired node
- Dynamic redundancy management, i.e. replacement of a failed node by a shadow node

The CNI contains a node reconfiguration field. In case a host decides to assume a new role, the name of the requested role is written into the reconfiguration field. The TTP controller checks if that role is permitted. If that is the case, it performs a node role change, and reconfigures the bus guardians to protect bus access in the new role.



FTU Layer

The FTU layer groups two or more nodes to form a fault-tolerant unit.
The FTU layer ensures that data are only visible in the FTU CNI after they have become permanent.

Examples for different FTU layer implementations:

1. Two fail-silent nodes can be grouped into an FTU that provides the specified service as long as one of the two nodes is operational. Fail-silence in the value domain is ensured by the host.
2. Three nodes can be grouped into a Triple Mode Redundancy (TMR) FTU. A TMR FTU can tolerate a single value failure in any of its nodes. Synchronization of the three nodes is implemented by the lower layers.
3. A FTU can comprise of software subsystems executing on different nodes.

The FTU membership service is provided by the FTU layer.
The FTU layer can be implemented in the host computer or in the TTP/C controller.
A basic TTP/C controller, implemented in hardware, does not include the FTU layer, but rather provides the Basic CNI interface to the software in the host computer.

10.3 Time Management

The CNI is the only interface of the communication system that is visible to the software of the host computer. It is the programming interface of the TTP network.
The CNIs for the TTP/C and TTP/A protocols are upward compatible.

Status Registers
Global Internal Time
SRU-Time (part of C State)
MEDL (part of C State)
Membership (part of C State)
Status Information

Control Registers
Watchdog
Timeout Register
Mode Change Request
Reconfiguration Request
External Rate Correction

Structure of the CNI

The CNI is a data-sharing interface between the RM layer and the FTU layer. It consists of

- a status/control area, containing system information. It serves to facilitate communication between the host computer and the TTP controller via dedicated data fields
- a message area, containing the messages sent or received by the node, including a control byte for each message
- a control line from the TTP controller to the host computer, signalling the tick of the global clock

Optionally, there may be another control line from the TTP controller to the host to signal that a certain time has been reached.

Status/Control Area

The status/control area of the CNI is a memory area of the DPRAM containing the control and status information shared between the TTP controller and the host computer.

Status registers updated by the TTP controller:

The following status registers are updated by the TTP controller:

- Global time (two byte) of the cluster, established by the mutual internal synchronization of the TTP controllers.
- SRU-time, containing the current global time in SRU slot granularity. This time stays constant during a complete SRU slot and increases at the beginning of the next SRU slot.
- MEDL position, denotes the current operating mode of the cluster and the current position in the message descriptor list.
- Node membership vector, comprises as many bits as there are nodes in the cluster. Each node is assigned to a specified bit position of the membership vector. When the bit is set to “TRUE”, the node was operational during its last sending slot. The membership is adjusted at the end of each SRU slot after all messages from the sending node must have arrived and the CRC fields have been analyzed.
- Status information, diverse status and diagnosis information regarding operation of the protocol

The SRU-time, MEDL position, and node membership vector form the h-state of the communication controller (called C-state). The protocol only operates correctly if all members of the ensemble have the same C-state.

The protocol continually enforces synchronization of the C-state between sender and receiver.

Control registers written by the host:

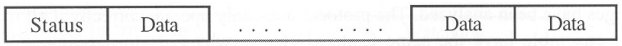
The following control registers are written by the host CPU:

- Watchdog field, must be periodically updated by the host CPU. The TTP controller checks this field to determine if the host CPU is alive. If the CPU does not update this field anymore, the controller assumes that the CPU has a failure and stops sending messages on the network.
- Time-out register, permits the host computer to request a temporal control signal (an interrupt) at a specific future point of the global time. This permits the host to synchronize activities with the global time in the cluster.
- Mode-change request, can be used to request a mode change (e.g., airplane on the ground or in the air) to a new schedule in all nodes of a cluster. This mode change request is transmitted to all other nodes at the next sending point of this node. TTP permits immediate mode changes and deferred mode changes. A deferred mode change is delayed until the start of the next cluster cycle. A mode change is potentially dangerous. The MEDL of the communication controller contains a static lock that can be turned on before system start so that mode changes are disabled.
- Reconfiguration request, permits the host CPU to request a role change of the node. In case a host detects that an important node has failed the host can execute a role change request to perform the function of the failed node. To avoid erroneous role

changes, the role-change mechanism is protected by special permission fields in the MEDL.
- External rate correction is provided for external clock synchronization. This permits a time gateway to request a bounded common-mode drift of all nodes in a cluster to synchronize with an external time source, such as a GPS time receiver.

Message Area

The message area is application specific and determined by the MEDL of the TTP controller. Beyond the actual message data, a message entry contains a status byte indicating potential error conditions.



Consistent Data Transfer

Since access to the CNI can be concurrent, inconsistencies could result when at the same time data is being written and read.
Consistency of a single-word data transfers across the CNI is guaranteed by the hardware arbitration of the DPRAM. Multi-word data-transfer is realized as follows:

Controller to host:

When a message arrives, it is copied from the receive buffer of the TTP controller into the message area of the CNI. The time when this happens is known a priori.
Together with the message data field, a status byte is written by the TTP controller. All this occurs before the end of each SRU slot.

Access conflicts can be avoided in two ways:

1. The host avoids reading data from the CNI during the times it knows the TTP controller writes data to the CNI. This is possible if the host can derive its read-access intervals from the global time base.
2. A non-blocking write protocol is being used. When the TTP controller writes to the common data area, a flag is set indicating that a write is in progress. When the host computer tries to read data from the shared area, and detects during any single read operation that the write flag has been set, it stops the read operation and retries with a small delay. This protocol ensures that the TTP controller is never delayed while accessing the CNI.

Host to controller:

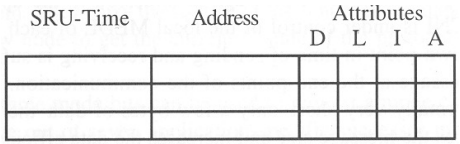
The host needs to be aware of the current time and knows a priori when the TTP controller reads from the CNI. The host operating system must synchronize its output action such that it does not write into the CNI when the TTP controller performs a read operation.

10.4 Error Detection

The Message Descriptor List (MEDL)

The Message Descriptor List (MEDL) is a static data structure within each TTP controller that

- controls when a message must be sent on or received from the communication channel
- contains the position of the data in the CNI



The MEDL serves as a dispatching table for the TTP controller. The length of the MEDL is determined by the length of the cluster cycle, i.e., the sequence of TDMA rounds after which the operation of the cluster repeats itself.

MEDL entry:

An entry in the MEDL comprises three fields:

- a time field (SRU-time), with the point in global time with SRU granularity when the message specified in the address field must be communicated
- an address field, points to the CNI memory cells where the data items must be stored to or retrieved from
- an attribute field with four subfields:

10.4 Error Detection

- a direction subfield (D), specifying if the message is an input or output message
- a length subfield (L), denoting the length of the message that must be communicated
- an initialization subfield (I) specifying whether the message is an initialization message or a normal message
- a parameter subfield (A) containing additional protective information concerning mode changes and node role changes

The host can only execute mode-changes permitted by the attribute field of the MEDL. In safety-critical systems, all mode changes requested by a host can be blocked by the MEDL.

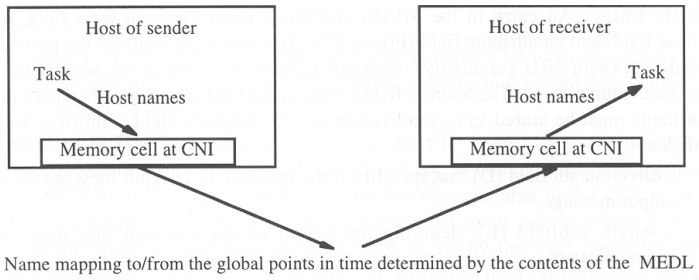
Each node must have its personal MEDL. The set of all personal MEDLs of a cluster must be consistent.

MEDLs are generated during design time by cluster compilers.

Name mapping:

Name mapping is performed under the control of the MEDL in each controller. Unambiguous identification of each message is given by its global transmission time. Thus, name mapping from the sending host to the receiving host is performed

- from host to CNI memory cell at sender and unique global transmission time
- from CNI memory cell to host at receiver and unique global transmission time

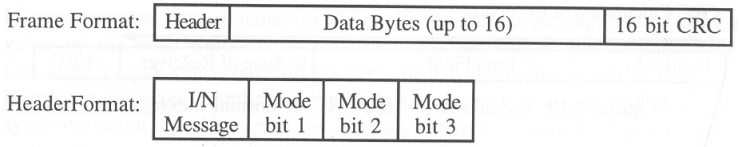


10.4 Error Detection

Frame Format

A TTP/C frame consists of three fields:

- a four-bit header
- a variable length data field of up to sixteen bytes (length is defined in MEDL)
- a two or three byte CRC field



During normal operation, a node transmits one such frame during an SRU slot on each of the two replicated channels.

First bit of the header:

This bit determines if the message is a normal (N) message or an initialization (I) message. I-messages are used to initialize the system. They carry the C-state of the sender in the data field and make it possible for a new node to get the current C-state of the protocol when joining the ensemble.

Mode bits:

These bits can be used to request a mode change in all nodes of the cluster. The change is differential; one out of seven successor modes to any given mode can be selected. The mode change mechanism can be disabled by setting parameters in the MEDL.

Data field:

This field contains up to sixteen data bytes. The actual length is given by an entry in the MEDL.

CRC field:

This field contains the CRC check sum for communication error detection.

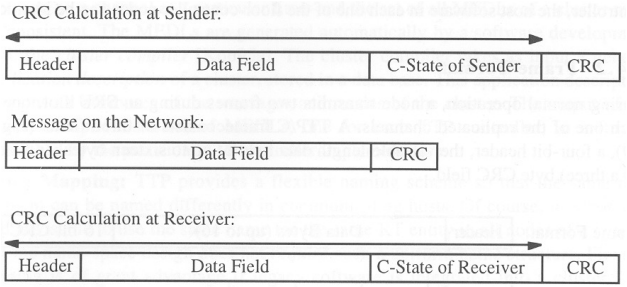
CRC Calculation

The CRC of an I-message is calculated of the concatenation of the header and the data bytes.

For N-messages the procedure is slightly different.

At the sender, the CRC is calculated over the message contents concatenated with the sender's C-state.

At the receiver, the CRC is calculated over the received message contents concatenated with the receivers C-state.
If the result of the CRC at the receiver is negative, then either the message has been corrupted during transmission, or there is a disagreement between the C-states of the sender and the receiver. In both cases, the message must be discarded.



The Membership Service

The membership field of a node contains one bit for each cluster node. If one out of the redundant messages is received by a receiving node, the receiving node considers the sender node operational at this membership point.
The node is considered operational until its next membership point in the following TDMA cycle. If a node fails within this interval, the failure is only recognized at the next membership point. Thus, the delay of the membership information is at most one TDMA cycle.
If none of the expected messages arrives with a correct CRC, then a receiver considers the sending node as failed and clears the membership bit of the sender at the end of the current SRU slot.

If a particular node did not receive any correct message from a sending node, e.g., because the incoming link of the receiver has failed, it assumes that this sending node has crashed, and it eliminates it from its membership vector at the end of the current SRU slot.

If, however, all other nodes received at least one these messages they come to a different conclusion about the membership. This creates two cliques that cannot communicate with each other because they contain a different C-state (the membership vector is part of the C-state).

TTP contains a mechanism which let the majority view in such a conflict situation, i.e., the node with the failed input port, which is in the majority, is removed from the membership. Before sending a message, a node counts its negative CRC results during the last TDMA round. If more than half of the messages received have been discarded because of a failed CRC, the node assumes that its C-state differs from the majority, terminates its operation and thus leaves membership.

This mechanism avoids clique formation among the nodes of the ensemble. Agreement on membership is thus bound to indirect acknowledgement of message reception by the majority.

Clock Synchronization

Clock synchronization within TTP does not require any special synchronization messages or send time information as part of a message.

Each node knows from its MEDL the expected time of arrival of each message. The deviation between the observed time of arrival to the expected time of arrival is thus a measure for the clock difference between the receiver's clock and the sender's clock.

A fault-tolerant clock synchronization algorithm is applied periodically to synchronize the global time of each node in a cluster and keep it within a predefined precision.

10.5 Case Study: OSEK and AUTOSAR

Deadlines can only be guaranteed if worst case execution (WCET) times of all application tasks are known a priori.

In addition, the worst case delays caused by administrative functions (e.g. operating system services, context switches, scheduling) need to be known. These delays we call the worst case administrative overhead (WCAO).

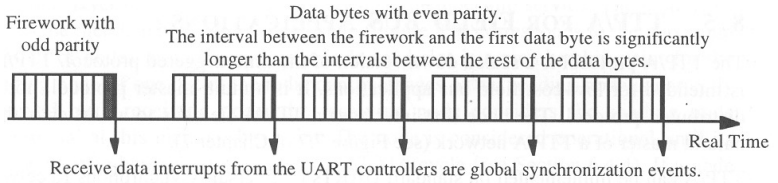
Principles of Operation

WCET depends on

- source code of task
- properties of object code generated by compiler
- characteristics of compiler

Round:

Problem is to find the longest path through the code (critical path). Number of paths increases exponentially with size of program. Annotations provided by the programmer can assist a tool to extract the critical path, by providing additional semantic information.



Modes:

The execution time of source code statements depends heavily on the object code generated by the compiler. The compiler generated object code timing analysis must be related to the source program by means of statement-level annotations.

Time-outs:

The execution time of source code statements depends heavily on the object code generated by the compiler. The compiler generated object code timing analysis must be related to the source program by means of statement-level annotations.

Error Detection and Error Handling

WCET depends on

Error detection in the time domain:

The execution time of source code statements depends heavily on the object code generated by the compiler. The compiler g

Error detection in the value domain:

The execution time of source code statements depends heavily on the object code generated by the compiler. The compiler g

Response Time of a TTP/A System

WCET depends on

Points to Remember