Time Triggered Protocol

Seminar paper for Special Emphasis A SS/22

1st Aditya Kumar
dept. of Electronic Engineering
Hochschule Hamm Lippstadt
Lippstadt, Germany
aditya.kumar@stud.hshl.de

***Abstract*—The time triggered protocol (TTP) is a distributed fault-tolerant real-time communication protocol. It consists of two real time protocols TTP/C and TTP/A that follow the defined Time-Triggered Architecture. To provide collision-free bus allocation, both protocols use a Time-Division Multiple Access (TDMA) technique. The main focus of TTP/C is to connect the components to create a highly reliable real time system capable of crucial applications such as X-by-wire in the automotive and avionics domains. TTP/C employs replication by using a duplicated bus system as well as a guardian to keep babbling idiot failures at bay. TTP/A is intended to integrate the sensors and actuators into a simple network and reduce cost.**

I. Introduction

Distributed electronics is increasingly becoming popular as they offer improved safety, reduced weight and cost of systems, and maintainability. This has lead to a shift from traditional electro-mechanical solutions to to distributed electronic control systems. The avionics's industry demands a more reliable communication systems. The requirements for this application requires a dependable real-time communication system that interconnect control nodes. In addition such a communication system would require a transducer bus that would provide the interface for various actuators and sensors of the given system. In this paper we will be talking about the time triggered protocol that targets the required needs mentioned above. We will be discussing the time triggered architecture and its TTP/C protocol in detail followed with a brief discussion of TTP/A protocol. The protocol has played a crucial role in the industry with over 500 million flight hours and over 1 billion operational hours have been accumulated by TPP controllers for aviation and railway applications.

The TTP/C is a highly reliable real-time communication service with clock synchronization, fault-tolerant and membership service which is suitable for x-by-wire systems in the automation and aviation domain. On the other hand the time triggered field-bus TTP/A integrates smart transducer in all types of distributed systems.

II. Time Triggered Architecture

The Time-Triggered Architecture (TTA) creates a framework for the domain of large distributed embedded real-time systems in high-dependability environments.[1] A node which is the basic building block in the TTA is made up of three components:

- Host Computer : Software for the relevant system application runs on the Host Computer
- Controller Network Interface (CNI) : The CNI is situated between the Host and the Controller that separates the network from the application-level software. A Message Descriptor List(MeDL) excites within the CNI that contains a data sharing interface allowing Controller and the Host to access the shared memory.
- Time-Triggered Controller : The Controller serves as the connection between the Node and the shared network and provides with the essential services for the protocol. "...the TTP/C controller provides guaranteed transmission times with minimal latency, jitters, fault-tolerant clock synchronization, and fast error detection." [2]

These nodes behave as a region to contain faults. Two replicated channels are connected to these nodes to tolerate the loss of communication thus forming a cluster for fault tolerance.

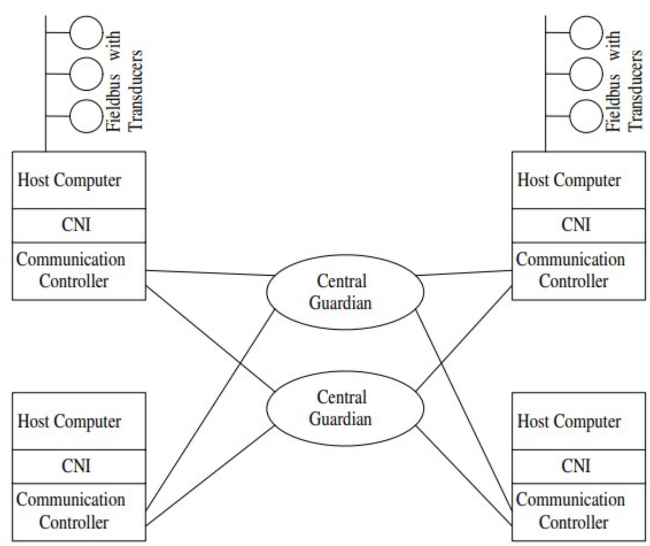

Fig. 1. Star Network Topology

The TTA supports two network topologies: the star and the bus. Each node in the bus topology has a bus guardian component(see figure 2) that regulates access to the communication channel by the node's communication controller. The bus guardian is constructed as a separate component to prevent a node from sending data outside of its designated time slot, Thus in the event of a component failure this prevents a babbling idiot failure of a node. Two star couplers serve as central bus guards(see figure 1) in the star topology making it a better approach. The following are some of the reasons why:

- The star topology is robust to spatial proximity faults because the star coupler is separated from the nodes physically.
- Since only one bus guardian per channel is necessary, the star approach is economically attractive [3].
- Irregular node failures are isolated by central guardians. Signal reconfiguration of the sender's message is then used to deal with off-specification defects that aren't caught in the bus topology.

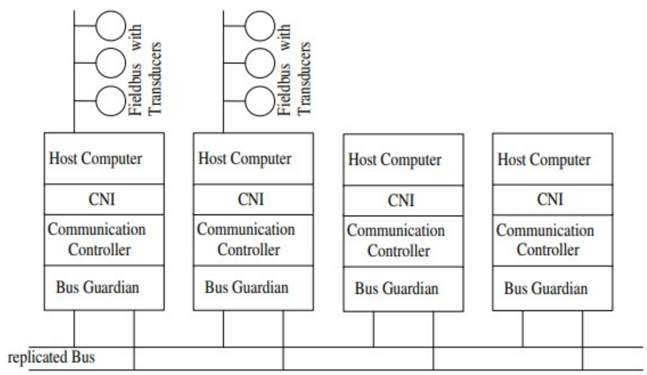

Fig. 2. Bus Network Topology

III. THE TTP/C PROTOCOL

The TTP/C is a safety-critical time triggered protocol that Uses a TDMA approach over the replicated communication channels to provide an autonomous fault-tolerant message transmission at known times and with little jitters between the CNIs of cluster nodes. It also uses fault-tolerant clock synchronization that does not rely on a central time server to build the global time base and imparts a membership service that notifies every appropriate node of data transport consistency. Clique avoidance is used to discover faults that aren't part of the fault hypothesis, which isn't allowed at the protocol level.

the following are the key characteristics of the ttp/c protocol:

A. Scheduling

The scheduling in TTP/C systems is static which means that the decision about when a node can send a message to the system bus is decided at design time as it shares it roots with the TDMA approach. The lattice points of a TTP/C called action lattice, specifies the timings when the different nodes in a system are enabled to communicate. The difference in time between two adjacent points in the action lattice represents the system's "basic cycle time" and serves as a lower constraint on its reaction time. To identify which tasks must be completed at run time, the Operating system in a TTP/C system employs table look-ups from a table which is produced at compile time and the schedule is stored in the message descriptor list(MeDL) situated in the communication controller.[4]

B. Use of State Message

Nodes create a state message in each TDMA round under TTP/C. These rounds are divided into slots where each system has its own sending slot. The TTP/C Host sends state messages to the CNI (Controller Network Interface) for transmission via the system network. Messages are not queued; instead, they are broadcast in each round and subsequently overwritten by the following message. It is important to send frames in each round.

C. Clock Synchronisation

As previously stated, the communication controllers in the node of a TTP/C system, have access to the bus based on a prior allotment of time slot.Due to this, keeping track of global time is essential to the protocol's successful operation. Given a prior scheduling allocation, all nodes in a cluster of TTP/C node is aware of which node has access to the bus during a certain time period. A node can then calculate the difference between the time when the messages are received from other nodes. this is because TTP/C is a broadcast protocol, so all nodes hear all messages,thus knowing the schedule by noting the time when messages are received by other nodes.[5]

D. Reliability and Fault-Tolerance

The necessity for extremely dependable operation is obvious since TTP/C targets embedded devices that emphasis on the safety-critical needs. Stated below are the several characteristics of the TTP/C protocol that contributes to its fault-tolerance and reliability.

1) Fail-Silence -TTP/C nodes are meant to accept a lot of the accountability for detecting flaws in their own operations."Each and every node must give results that are right in both the value and time domain, or no results at all." [6] When a node identifies a problem with its operation, it shuts down.
2) Replicating the Components -The TTP/C protocol allows for fault tolerance by replicating the components of a real-time system. A node that consist of the host, CNI, and TTP/C controller – is replicated to provide redundancy in the event that one of them fails. The redundant components are then bundled together to form a Fault Tolerant Unit (FTU). [7] TTP/C also allows twin system buses, which transport a second set of signals.
3) Bus Guardian The Bus Guardian (BG) is a hardware component of the TTP/C Controller that acts as the gateway for the system bus. The Bus Guardian's primary responsibility is to enable the bus driver only during the transmission slot for its node, and to ensure that the bus driver is deactivated at all other times. This helps to avoid "babbling fool" mistakes and is critical to the system

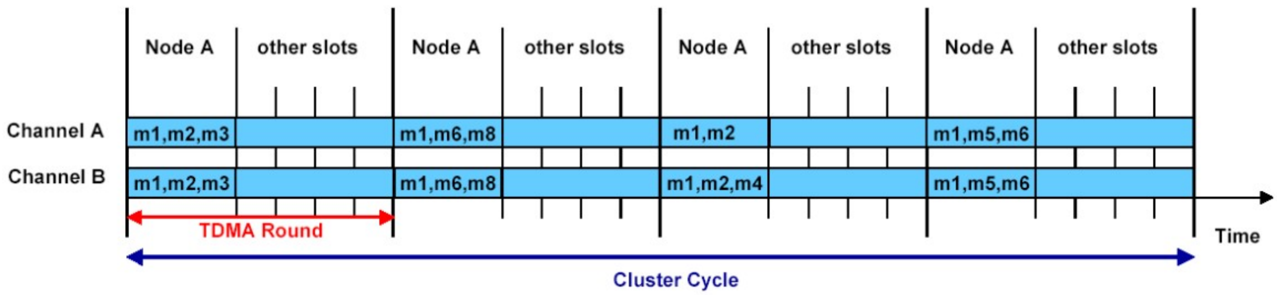

Fig. 3. Frames, messages, slots, TDMA round, and cluster cycle in TTP/C communication

E. Support for Consistency

It is advantageous to have data consistency for complicated distributed systems. Consistency in single-node systems can be assumed because data transferred to memory is available to all software subsystems at the same time, and all subsystems read the same value. This kind of consistency cannot be assumed in distributed systems due to the following two factors: message transmission delays that affect the present state must be considered. It is not necessary that a message always arrive at all recipients at the same time. Individual nodes may also fail or communications may become lost.

If state consistency (irrespective of the application) is not maintained at the communication platform level the design and programming in a distributed system becomes arduous and challenging, especially for sophisticated applications. Application-independent state consistency support is capable of not only freeing the application CPU of the burden of executing consistency protocols, but also of checking the validity of complicated consistency algorithms for all applications. As a result, data consistency must be implemented on the communication controller inside of the CNI.

The application subsystem's logic would be extensive and complex due to the enormous number of probable node failures, communication failures, or changes in message arrival timing and sequence. As a result, state consistency should be maintained as a basic communication subsystem service that meets the following requirements:

- All nodes that are correct, agree on the same data
- All given nodes agree on the data that is sent by a correct sender
- All correct subsystems deliver the received value at the same point in time.

F. Membership and Acknowledgement

TTP/C was designed with the notion that the protocol should provide data consistently to all correct nodes in the distributed system and that, in the event of a failure, the communication system should determine which node is problematic on its own. The membership protocol and an acknowledgment mechanism achieve these responsibilities.

In a TTP/C-based cluster, each node keeps a membership list consisting of all the correct nodes. This data is updated locally in response to successful (or unsuccessful) data transmissions, reflecting the receiving node's local view on all the other nodes. Each receiver see's and checks the sender's membership, which is included in the sender's message or disguised in the Cyclic Redundancy Check (CRC) calculation, within each transmission.

1) Acknowledgment: a sender node looks for acknowledgement from the other nodes after transmission to check whether the signal was received by the receiver (on the communication level). This is accomplished by examining the first as well as the second sender's membership list. These nodes indicate when transmission was successfully received if the sender node appears on their membership list. Otherwise, the sender node will be alerted that the communication failed. The state message is re-transmitted in the next cycle according to the time-triggered principle.
2) Membership Consistency Check: In a TDMA round, each node sees and checks the membership lists of all the other nodes due to the rigorous round-robin scheme of the TDMA round. Each sender with a unique membership list is believed to be in error. This ensures successful communication between nodes with consistent view of all nodes that accept each other as member.
3) Clique Avoidance The clique avoidance mechanism is active in order to detect multiple component flaws and inconsistencies. The algorithm examines if a node is a part of the majority clique before each transmit operation. If the node is in a minority clique, it signifies that an extremely rare fault event has occurred outside of the fault-tolerance, causing inconsistency. This condition is sent to the application software, which can then determines whether to perform a fail-safe operation.

Communication consistency is achieved by combining these techniques with the common time foundation established by clock synchronisation. This ensures that all correct nodes receive the same information at the same time.

IV. The TTP/A Protocol

TTP/A is a low-cost fieldbus protocol used for smart transducer interconnection. The Object Management Group (OMG) established the protocol as a smart transducer interface standard in 2002[8]. TTP/A is a time-triggered transport service that defines a interface between transducers and a CORBA network within a distributed smart transducer subsystem. The smart sensor technology offers a number of advantages from the points of view of technology, cost, and complexity management [9]:

- Non-linear sensor signals that are electrically weak can be conditioned, calibrated, and converted to digital form on a single silicon chip without picking up noise from long external signal transmission lines. [10]
- The smart sensor has a well-defined digital communication interface to a sensor bus, allowing for ”plug-and-play” functionality if the sensor has a reference to its documentation in the form of an electronic data sheet, as described in IEEE 1451.2 Standard [11].
- It is possible to use the network to monitor all the local operation of the sensing device.

V. Conclusion

Despite the fact that TTP/C and TTP/A are both time-triggered real-time communication protocols, they do not compete with each other. TTP/C focuses on the connectivity of components to create a highly reliable Real-time system suitable for critical applications such as X-by-wire in the automotive and avionics industries. TTP/A, on the other hand, allows for simple network integration of sensors and actuators.

References

[1] C. Scheidler, G. Heiner, R. Sasse, E. Fuchs, H. Kopetz, and C. Temple. Time-Triggered Architecture (TTA). Advances in Information Technologies: The Business Challenge, IOS Press, 1997.

[2] Ross Bannatyne, “Time Triggered Protocol-Fault Tolerant Serial Communications for RealTime Embedded Systems,” Wescon 1998, p. 88.

[3] TTTech Computertechnik, Sch¨onbrunnerstraße 7, Vienna, Austria. The Time-Triggered Architecture – A Platform for Safety-Critical Applications in the Automotive Industry, 2002. Available at http://www.tttech.com.

[4] SAE, “Glossary of Vehicle Networks for Multiplexing and Data Communications,” SAE Recommended Practice, J1213/1, SAE, Sept. 1997.

[5] Georg Kroiss, “The Time Triggered Communication Protocol TTP/C,” Real Time Magazine, No.4, 1998, p. 101.

[6] Elmar Dilger, Thomas Fuhrer, Bernd Muller, and Stefan Poledna, “The X-by-Wire Concept: Time-Triggered Information Exchange and Fail Silence Support by New System Services,” SAE 1998 Conference, p. 145.

[7] Hermann Kopetz, Real-Time Systems: Design Principles for Distributed Embedded Applications,p. 177.

[8] Object Management Group (OMG). Smart Transducers Interface V1.0, January 2003. Specification available at http://doc.omg.org/formal/2003-01-01 as document ptc/2002-10-02.

[9] H. Kopetz. Software engineering for real-time: A roadmap. In Proceedings of the IEEE Software Engineering Conference, Limmerick, Ireland, 2000.

[10] P. Dierauer and B. Woolever. Understanding smart devices. Industrial Computing, pages 47–50, 1998.

[11] L. H. Eccles. A brief description of IEEE P1451.2. Sensors Expo, May 1998.