Byzantine Fault Containment in TTP/C

Günther Bauer Hermann Kopetz Wilfried Steiner

Institut für Technische Informatik
Vienna University of Technology
Treitlstr. 3/3/182.1
A-1040 Vienna, AUSTRIA
E-mail: {sc,hk}@vmars.tuwien.ac.at

Abstract

The TTP/C protocol is a communication protocol for safety-critical real-time applications. It is designed to meet both the cost constraints of the automotive industry and the stringent safety constraints of the aeronautics industry. This is achieved by using the static nature of the TTP/C communication pattern to build relatively cheap communication controllers being supervised by guardians that protect correct nodes from faulty ones. The complexity and, thus, the costs of these guardians determine the type of node failures a TTP/C-based network can tolerate. In this paper, we will give a short overview of the TTP/C protocol and discuss its fault hypothesis. We will then introduce a general guardian that enables a TTP/C-based network to tolerate arbitrary node failures.

1. Introduction

The Time-Triggered Architecture (TTA) is a distributed computer architecture for highly dependable real-time systems. The core building block of the TTA is the communication protocol TTP/C. This protocol has been designed to provide non-faulty nodes with consistent data despite the presence of faulty nodes or a faulty interconnection network channel. To achieve consistency the protocol algorithms assume that a fault is either a reception fault or a consistent send fault of some node. Although the protocol uses this rather optimistic failure mode assumption, the TTA can isolate and tolerate a broader class of faults. This is possible by making intensive use of the static knowledge present in a TTP/C-based distributed computer system. This off-line available knowledge allows to build interconnection networks which transform arbitrary failure modes of nodes into failure modes the communication protocol can deal with.

This paper discusses a promising new approach to transform failure modes in TTP/C-based systems utilizing a star topology interconnection network: the central guardian [3, 1]. In a star network architecture all TTP/C nodes can share guardians that are physically located at the star coupler of the network. This setup requires only a single guardian per replicated communication channel rather than a guardian for each node as needed in a bus setup [6]. Thus, the guardian may implement sophisticated algorithms while keeping overall system costs low. In fact, a central guardian may even be designed as smart as to isolate arbitrary node failures.

The remainder of the paper is organized as follows: we start with a short introduction to the TTP/C communication protocol and will present its fault hypothesis. We will then discuss the requirements imposed on a guardian for TTP/C that enables isolation of arbitrary node failures.

2. The TTP/C Communication Protocol

A TTP/C network consists of a set of communicating nodes connected by a replicated interconnection network. A node computer comprises a host computer and a TTP/C communication controller with two bi-directional communication ports. Each of these ports is connected to an independent channel of a dual-channel interconnection network. Via these broadcast channels the nodes communicate using the service of the communication controller.

The TTP/C protocol implements broadcast communication that proceeds according to an *a priori* established time-division multiple access (TDMA) scheme. This TDMA scheme divides time into slots each being statically assigned to a particular node. During its slots the node has exclusive write permission to the interconnection network. The slots are grouped into rounds: in the course of a (TDMA) round every node is granted write permission in exactly one slot.

This work has been supported by the European IST project *Next TTA* under project No IST-2001-32111.

Furthermore, nodes always send in slots having the same relative position within a round; finally, the slots assigned to a particular node have the same length in each round. A distributed fault-tolerant clock synchronization algorithm establishes the global time base needed for the distributed execution of the TDMA scheme.

A cluster cycle comprises several TDMA rounds and multiplexes the slots assigned to a node in succeeding TDMA rounds between different messages produced by the node (this is similar to the TDMA round, which multiplexes the communication channels between several nodes). Every node has knowledge – stored in read-only memory – of the complete communication pattern (and not only of the slots assigned to itself). These data are called message descriptor list (MEDL) and allow nodes to know *a priori* the types of messages being sent or received. Thus, there is no need for transmitting the sender IDs or message IDs explicitly.

TTP/C messages are called frames and the protocol defines three types of frames: normal frames (N-frames) carry user data. Initialization frames (I-frames) carry protocol-specific state information that allows nodes to integrate into an operational cluster. Finally, extended frames (X-frames) contain both user data and protocol state information. The type of a frame to be transmitted in a particular slot of the TDMA round is also stored in the MEDL. In addition – to allow for node integration – frames carry an identifier bit in a frame header.

By periodic examination of frame states the protocol establishes a membership service: if a node receives a correct frame [4] on either of the communication channels, it considers the respective sender correct. A correct receiver will consider a frame correct if it meets all of the following requirements:

- transmission of the frame starts and ends within the temporal boundaries of its TDMA slot
- the signal constituting the frame on the physical layer obeys the line encoding rules
- the received frame passes a CRC check
- sender and receiver agree on the distributed state of the TTP/C protocol (i.e., the C-state)

It does not matter if the sender is in fact correct (as judged by an omniscient observer) or what faulty receivers conclude. If a node receives a correct frame, it assumes that the contents of the frame are authentic and that sender and receiver agree on the distributed state of the communication system, i.e., the controller state (C-state). The C-state consists of the membership, the global time the frame broadcast was started at, and the number of the current TDMA slot. To test C-state agreement when an N-frame (which contains solely user data) is received, the CRC check mentioned above is performed on the frame data concatenated with the local C-state (extended CRC check [4]). If the resulting CRC checksums are identical at sender (i.e., the checksum transmitted with the frame) and receiver, the receiver assumes that it maintains the same C-state as the sender. Alternatively, when an I-frame or an X-frame is received, the C-state data transmitted with the frame are compared to the receiver-local view of the C-state. In any case the membership service of the protocol ensures that a node can only succeed in broadcasting frames if it maintains a correct C-state (i.e., the same C-state as the receivers).

To allow for integration of nodes into an active cluster, some nodes of the cluster periodically broadcast their respective C-state in I-frames or X-frames. Nodes willing to integrate can learn membership, global time, and the actual position within the global communication pattern from the C-state. Thus, the node is enabled to participate in communication after having received an I-frame or an X-frame.

3. TTP/C Protocol Fault Hypothesis

In the following paragraphs we will introduce the fault containment regions [2] of the TTP/C protocol. Further, we will provide definitions of types and frequency of faults that can be withstood by the protocol. Finally, we will define a minimum configuration needed to tolerate these faults.

3.1. Fault Containment

The TTP/C protocol distinguishes between two types of fault containment regions:

- node computers (comprising a host computer and a communication controller part) and
- channels of the interconnection network.

A fault containment regions is supposed to fail as a unit. Distinct fault containment regions will fail statistically independently if the respective faults are covered by the fault hypothesis.

3.2. Node Faults

As for the frequency of node faults, the fault hypothesis of the protocol claims that

1. only one faulty node exists within the duration of a TDMA round
2. a node may become faulty only after any previously faulty node either has shut down or operates correctly again.

With respect to the types of node faults, the TTP/C protocol assumes that

3. a transmission fault is consistent (i.e., if a faulty node broadcasts a frame on a correct channel, all receiving nodes will consistently consider the respective frame faulty or correct)

4. a node does not send data outside its assigned sending slots on both channels of the interconnection network

5. a node will never send a correct frame outside its assigned sending slots

6. a node will never hide its identity when sending frames.

The fault hypothesis does not state anything about faults other than communication faults. Any fault of a node (even a reception fault) will either become manifest by a transmission fault of the affected node or will never be perceived by other nodes of the cluster.

3.3. Network Faults

With respect to the frequency of faults of a channel, the fault hypothesis states that

7. only one channel is faulty during a TDMA slot.

As to the types of interconnection network faults it must be guaranteed that

8. a channel does not spontaneously create correct frames

9. a channel will deliver a frame either within some known maximum delay or never.

3.4. Single Faults & Minimum Configuration

The TTP/C protocol promises to provide its consistent frame delivery and membership service even in the presence of faults provided that at most one component happens to be faulty in a particular slot. To achieve fault-tolerance, however, a minimum configuration must be ensured.

To tolerate a faulty node the minimum configuration in TTP/C requires in general that, in every slot, there exist at least three correct nodes which need to be correct for the whole duration of the slot. In particular, if the cluster operates in synchronous protocol mode, three correct nodes, which must actively participate in clock synchronization and are synchronized to each other, are required in a minimum setup. Further, to allow for integration of a correct node despite a faulty active node, an I-frame must be transmitted every TDMA round and there must be at least one correct node that sends I-frames.

4. The Tasks of the Guardian

The purpose of the guardian is to increase the probability that TTP/C nodes of a cluster will face only faults covered by the fault hypothesis as presented in Section 3. In principle, this is achieved by placing a guardian at the interface(s) of a component and let it control the appearance of the respective component at its interface(s) to other components and, thus, act as a failure mode converter. Consequently the failure modes of the component are – at the interface to other components – replaced by the failure modes of the guardian.

The central guardian discussed in this paper checks (at the operational level of the interface specification [5]) for the expected syntax and the timing at the interface of the nodes it supervises. It is thus able to transform types of faults. At its output interface the guardian will mirror the input received from the attached sender node if this input complies to some specified rules. Otherwise the guardian will exhibit a predefined behavior (that complies to the fault hypothesis).

To guarantee compliance to the types of node faults the guardian needs to transform the following failures of TTP/C communication controllers:

1. SOS failures in the line encoding of frames at the physical layer

2. SOS failures with respect to the timing of frame transmission

3. transmission of any data outside the assigned sending slot (both in synchronized cluster operation and during startup)

4. masquerading of nodes during the startup phase of the protocol.

Additionally, to provide fault isolation to integrating nodes:

5. transmission of invalid (i.e., non-agreed) C-state data.

Transformation of failure modes one and two ensures that transmission faults will be consistent. Supervision of failure mode three will guarantee that a node can never send anything outside its assigned sending slots. Finally, hiding its identity becomes impossible to a node if the guardian checks for cheaters. Thus, all assumptions regarding the types of faults as discussed in Section 3.2 are covered.

4.1. The Central Guardian Approach

Figure 1 provides the (logical) top level architecture of a TTA cluster utilizing a star topology network. The cluster comprises four regular nodes, two dedicated nodes, and two star couplers. The regular nodes are connected to each of the replicated channels of the (star topology) interconnection network via bi-directional links. Two independent central guardians are located at the center of each communication channel, i.e., at the star coupler. The guardian of a channel controls all the (frame) traffic at the respective channel. To achieve this, the guardian needs to be provided with the TTP/C clock synchronization service and needs to have access to C-state data. A dedicated node consisting of a TTP/C protocol controller provides these services (by providing the central guardian with a regular TTP/C protocol interface, i.e., the CNI). This controller is logically (as depicted in Figure 1) a regular TTP/C controller that does not send any frames and whose existence is thus transparent to other nodes in the cluster. Physically, the controller is located at the star coupler and is part of the guardian itself.

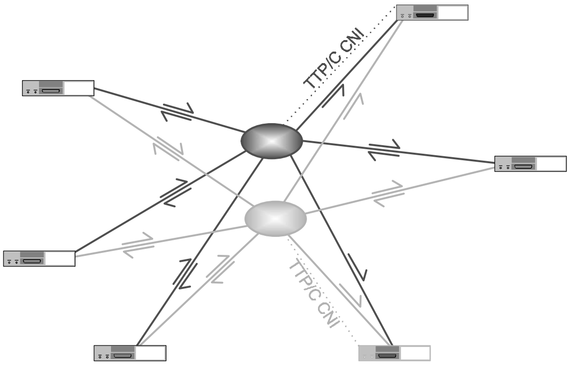

Figure 1. Star Topology Cluster

This approach provides both cost efficiency and a low statistical dependency of node and guardian faults. Cost efficiency is a consequence of needing only two guardians (one per channel) irrespective of the actual number of nodes in the cluster. Because of the strict "one-at-a-time" communication pattern of TDMA-based communication and the fact that a guardian protects receivers from faulty senders, it suffices to have, for all nodes, a single common guardian that is – at a particular point in time – logically assigned to the sender of the respective slot. Write access of a node is prohibited outside its respective sending slot.

The actual value of statistical dependency of node and guardian faults basically depends on the particular implementation. Influencing parameters are the type of physical connection between nodes and the star coupler, independence of power supplies, physical vicinity of the devices, and others. At the logical level nodes and guardians do not have any common mode failure modes.

Further, integrating a central guardian into the star coupler of a star network has the following advantages:

- The algorithms in the guardians can be extended to provide additional monitoring services, such as condition-based maintenance.
- If the guardians reshape the physical signals, the architecture becomes resilient to arbitrary node faults.
- Point-to-point links have better EMI characteristics than a bus and can easily be implemented on fiber optics.

5. Outlook

In this paper we have presented the TTP/C protocol, its fault hypothesis and minimum configuration requirements and the principles of a central guardian for this protocol. The central guardian is a natural yet powerful choice in star network topologies. Because a whole TTP/C cluster needs only two of these central guardians, its design is less constrained by cost arguments than a local guardian needed once (or even twice) for every node. In fact, a central guardian may contain algorithms so sophisticated that arbitrary node failures can be tolerated.

Currently, we are designing the algorithms for a smart central guardian to be applied in a star network topology. This guardian will be able to isolate arbitrary node failures, thus, allowing to waive sophisticated self-checking mechanisms when needing to ensure fail-silence failure semantics. Test series with first prototypes of this central guardian both in VHDL simulation and on an FPGA-based hardware prototype implementation provided promising results.

References

[1] G. Bauer, H. Kopetz, and P. Puschner. Assumption Coverage under Different Failure Modes in the Time-Triggered Architecture. *8th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2001), Antibes Juan-les-pins, France*, pages 333–341, Oct. 2001.

[2] C. Jones, M.-O. Killijian, H. Kopetz, E. Marsden, N. Moffat, M. Paulitsch, D. Powell, B. Randell, A. Romanovsky, and R. Stroud. Revised Version of DSoS Conceptual Model. Project Deliverable for DSoS (Dependable Systems of Systems), Research Report 35/2001, Technische Universität Wien, Institut für Technische Informatik, Treitlstr. 1-3/182-1, 1040 Vienna, Austria, 2001.

[3] H. Kopetz, G. Bauer, and S. Poledna. Tolerating Arbitrary Node Failures in the Time-Triggered Architecture. *SAE 2001 World Congress, March 2001, Detroit, MI, USA*, Mar. 2001.

[4] H. Kopetz. *TTP/C Protocol – Version 0.5*. TTTech Computertechnik AG, Schönbrunner Straße 7, A-1040 Vienna, July 1999. Available at http://www.ttpforum.org.

[5] H. Kopetz. On the Specification of Linking Interfaces in Distributed Real-Time Systems. Unpublished draft, Technische Universität Wien, Institut für Technische Informatik, Treitlstr. 1-3/182-1, 1040 Vienna, Austria, 2002.

[6] C. Temple. Avoiding the Babbling-Idiot Failure in a Time-Triggered Communication System. In *Proceedings of the 28th Annual International Symposium on Fault-Tolerant Computing (FTCS-28)*, pages 218–227, June 1998.