IST-2001-37652

Hard Real-time CORBA

Title

HRTP PCT Implementation

Authors

Manuel Rodríguez (UPM)
Thomas Losert (TUV)

Reference	IST37652/076 Deliverable 4.5
Date	2003-10-11
Release	1.0
Status	Final
Clearance	Consortium

Partners

Universidad Politécnica de Madrid
Lunds Tekniska Högskola
Technische Universität Wien
SCILabs Ingenieros

Summary Sheet

IST Project 2001-37652
HRTC
Hard Real-time CORBA

HRTP PCT Implementation

Abstract:

The present document describes the TTPIOP prototype implementation.

Copyright

HRTC Partners:

Universidad Politécnica de Madrid
Lunds Tekniska Högskola
Technische Universität Wien
SCILabs Ingenieros.

Release Sheet (1)

Release:	**0.1 Draft**
Date:	2003/10/09
Scope	Initial version
Sheets	All

Release:	**1.0 Final**
Date:	2003/10/11
Scope	Revision and corrections.
Sheets	All

Table of Contents

1. TTPIOP Prototype Implementation

At the present time and due to the late arrival (October 8th) of the TTP protocol the HRTP implementation of the PCT has not been done. The present document describes briefly the implementation of the prorotype. Anyway a HRTP implementation is being developed and at least the CCS control loop test will be performed.[1]

1.1. Description of Hardware

As hardware for the prototype a cluster consisting of 4 TTP-Monitoring Nodes has been used.

The node itself consists of the following components:

- CPU MPC855T (80MHz) containing a 32-bit PowerPC core
- with builtin fast ethernet controller but without an FPU,
- 16 MByte of SDRAM,
- 8 MByte of FLASH ROM,
- PCMCIA slot, and
- TTP/C-C2 communication controller.

The PCMCIA slot has not been used in this prototype as well as the fast ethernet (100Base-TX) connection which has been very useful for development but is not necessary for the demonstration of the prototype.

A more detailed documentation of the relevant parts of this node has already been part of document [HRT02].

[1] The code implementation can be reported as an addendum to this document if so considered by the reviewers.

1.2. Description of Operating-System

The Operating System (OS) of the prototype can be subdivided into two parts: the non-real-time part and the real-time scheduler that allows execution of real-time tasks with low latency.

The following is just a brief summary. More details about the software of the development-environment has already been part of document [HRT02].

Non-Real-Time Operating System

As Operating System Linux 2.2.4 is used. This is a standard time-sharing OS which provides good average performance and highly sophisticated services. Like other OSs, it offers several services to the applications like hardware management of devices, scheduling, and interprocess communication. But it suffers from a lack of real time support.

Real-Time Scheduler

In order to establish real-time capabilities the Real-Time Application Interface (RTAI) extension has been used that is not a real time operating system in the original sense. It is based on the Linux kernel, providing the ability to make it fully preemptable.

1.3. CORBA - ORB

Layering of protocols is an approved method for providing features (e.g. ISO OSI 7-layer reference model) since *decomposition* is a key element of scalability and extendability.

The OMG has begun standardization of the interface between ORB and transport in order to allow choosing ORBs and transport-plugin from different vendors. Although at the time of writing the standardization of the Extensible Transport Framework (ETF), former called pluggable transport, has not been finished, there are already interim submissions and it is unlikely that major changes will be applied on its way to the final document. In order to be able to validate the results in a prototype the ORB of the Integrated Control Architecture (ICa) has been used because of its support of an older submission of the ETF called Open Communication Interface (OCI). Compared with the more recent submissions there are no major conceptual differences and thus the results gained with this prototype also should be valid for a plugin according to the final specification.

TTP/C (see [Kop02]) has been chosen as underlying deterministic protocol for a prototype implementation of the transport-plugin.

This *transport-plugin* is based on the deterministic transport TTP/C. It can be subdivided into a part running in kernel-mode of (RTAI-)Linux and another part running in user-mode (see Figure 1: Overview of the OCI-Plugin).

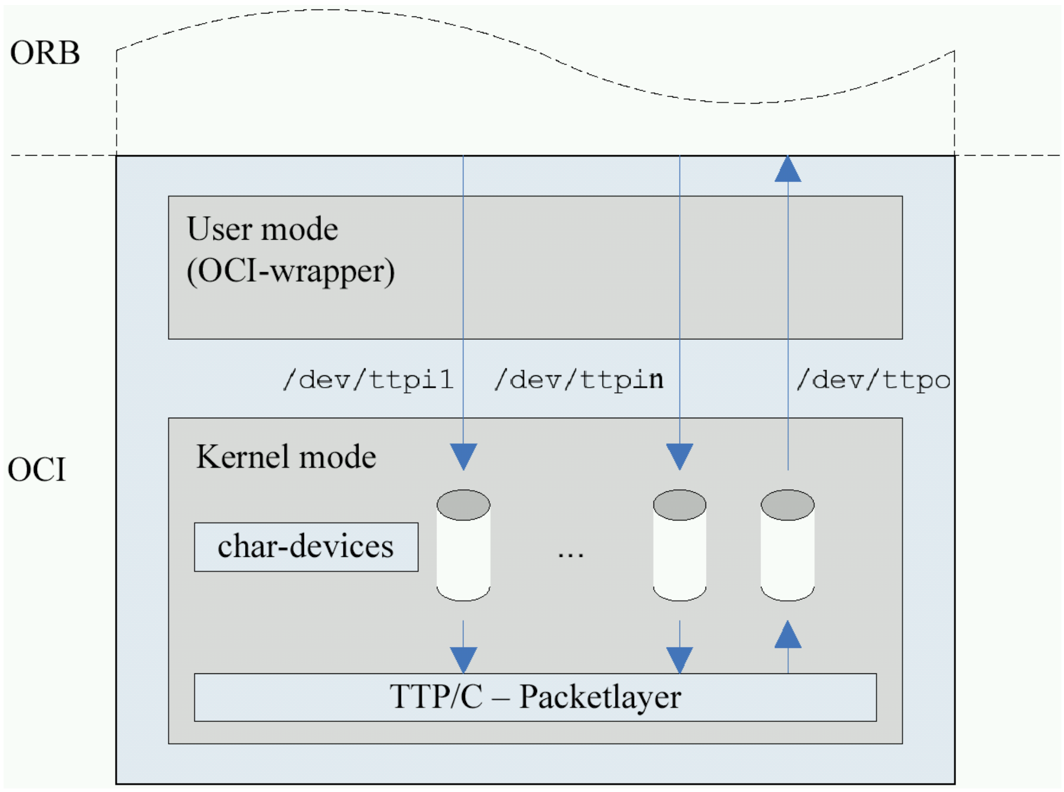

Figure 1: Overview of the OCI-Plugin

The user-mode part is a wrapper for the respective device-files in order to provide the interface according to the OCI-submission (see above) while the kernel-mode part provides the functions behind these device-files.

Roughly the kernel-mode part can be split into a packet-service and a device-driver for several character devices.

Packet-Service

The packet service is implemented as a time-triggered task derived from work described in [NEX03] that fragments outgoing messages into packets and reassembles incoming packets back into messages.

In order to fulfill its mission this task runs under control of RTAI and puts packets of outgoing data into the Communication Network Interface (CNI) thereby arranging the dissemination of the packets through the time-triggered communication service. Furthermore, the time-triggered communication controller places received packets in the CNI. The packet layer task retrieves these incoming packets and fuses them to messages.

As interface to the adjacent layer the packet-service provides a set of message queues in a shared memory. For preventing faulty nodes interfering with other nodes there are individual incoming queues for each of the n nodes as well as an outgoing queue for messages that are broadcast to all other nodes.

Device-Driver

The device-driver runs in kernel-mode of the non-real-time OS and allows to fetch and put messages into the message queues described above by means of reading/writing from/to read/write-only files in the filesystem with regular I/O-functions.

For performing this task the device-driver maintains the message queues and - in blocking-mode - releases the message to the calling read-operation as soon as the virtual TDMA-round[2] has been finished. This allows the calling process to be synchronized with the communication schedule of the underlying protocol.

Further this driver provides support for some extra char-devices that allow accessing, e.g., the global time (allowing the creation of timestamps) or the membership-vector of TTP/C (for detection of faulty nodes).

[2] In order to overcome the limitation of TTP/C regarding the maximum payload per slot several TDMA-rounds have been combined to a virtual TDMA-round.

2. Validation

For validation a cluster consisting of 4 nodes has been used with a TDMA-round length of 320 µs. The central guardian (see [BKS03]) has been replaced by a regular ethernet hub. Although this means that the possibility of some special failure modes, e.g., SOS-failures, has been neglected it is possible to reintroduce it later without any further changes since the central guardian is transparent for the nodes if it is necessary to consider these failure modes.

In the configuration used for the prototype implementation TTP/C allows establishing a global time with a precision below 1 µs. Since the interrupt for the packet-layer is generated by TTP/C and the periodic task of the packet-layer runs at highest priority under direct control of RTAI the jitter measured at the packet-layer is also about 1 µs.

Since the ORB is running in user-mode and the thin wrapper for providing the OCI interface can be neglected a test-program has been written that directly accesses the device-files. This program used – in the same manner as the ORB – blocking read-operations from these device-files for synchronization with the underlying transport and generated a timestamp upon the reception of each of the 1000 packets. The intervals between two packets have been visualized in Figure 2: Measurement with a virtual TDMA-round length of 3.2 ms.

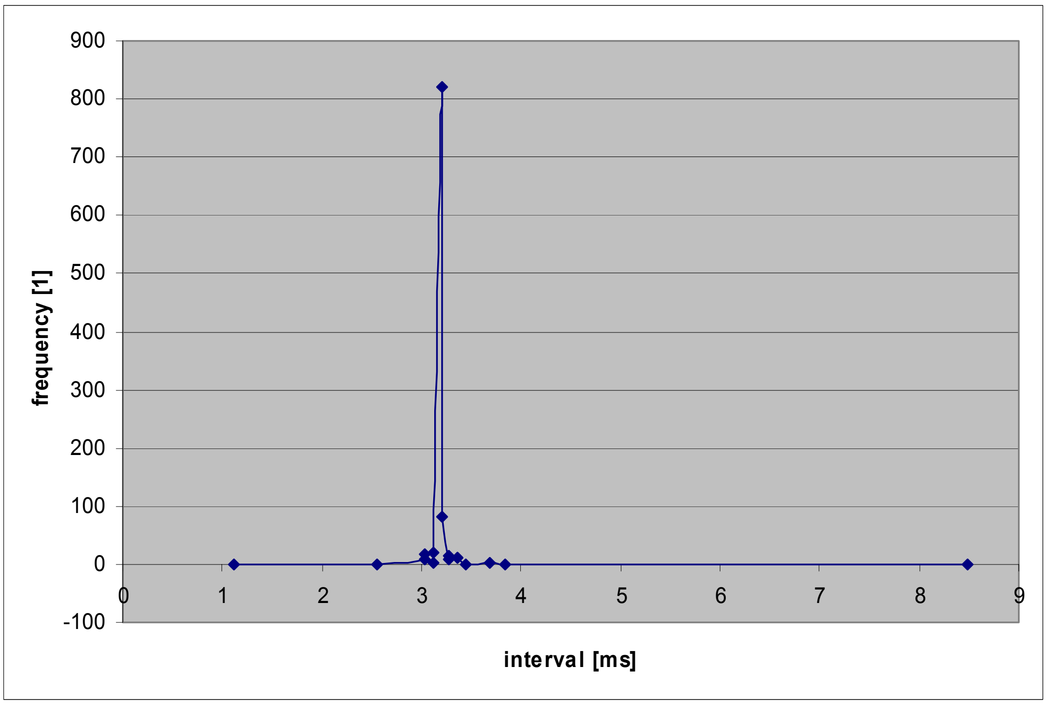

Figure 2: Measurement with a virtual TDMA-round length of 3.2 ms

About 90% of all measurements are in the expected interval of 3.2 ms ± 1 µs. Within the interval of 3.2 ms ± 160 µs have been more than 99% of all measurements. The remaining 1% has been distributed around milliseconds of the expected value.

Although the prototype implementation has behaved as expected for most of the measurements, it cannot be used in hard real-time applications because of the worst-case performance.

Although tracking down these runaways is very difficult since they do occur rarely and in a sporadic way and it would require deep insight into the internal mechanisms of the kernel as well as some modifications for testing (that could themselve influence the temporal behavior), we suppose that most of the "slight" deviations (within 160 µs of the expected value) are caused by kernel-internal operations (scheduling, RTAI) and most of the "heavy" deviations (up to milliseconds) are the result of blocking during I/O-operations or similar.

Since these unexpected deviations from the desired behavior could not be detected at the packet-layer (measured by switching I/O-pins and monitoring with an oscilloscope) but have been measured with the test-software running in usermode our conclusion is that running the ORB in kernel-mode (or under direct control of RTAI) should be a major concern when dealing with this problem.

3. References

[BKS03] Günther Bauer, Hermann Kopetz, and Wilfried Steiner. The Central Guardian Approach to Enforce Fault Isolation in the Time-Triggered Architecture. In Proceedings of the Sixth International Symposium on Autonomous Decentralized Systems (ISADS 03), pages 37–44, April 2003.

[HRT02] Thomas Losert. HW/SW-Platforms. HRTC Project (IST-2001-37652) Document 031, October 2002.

[Kop02] Hermann Kopetz. TTP/C Protocol – Version 1.0.0. Technical report, TTTech Computertechnik AG, Vienna, Austria, 2002. Available at http://www.ttpforum.org/.

[NEX03] NEXT TTA. Event-Triggered Services Prototype Implementation. NEXT TTA Project (IST-2001-32111) Deliverable D2.3, March 2003.