A reconfigurable data communication system for on-orbit serviceable, highly modularized satellites

vorgelegt von
Dipl.- Ing. Dung Tham

an der Fakultät V – Verkehrs- und Maschinensysteme
der Technischen Universität Berlin

zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften
- Dr. Ing. -
genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Dr.-Ing. Flávio Silvestre, Technische Universität Berlin

Gutachter: Prof. Dr.-Ing. Klaus Brieß, Technische Universität Berlin

Gutachter: Prof. Dr.-Ing. Sergio Montenegro, Julius-Maximilians-Universität Würzburg

Tag der wissenschaftlichen Aussprache: 30. April 2021

Berlin, 2021

Abstract

The concept of highly modularized satellites with the capability for On-Orbit Servicing is a new and innovative approach in space engineering promising high potential and great benefits in mid-term and long-term future. Highly modularized satellites represent the next generation of economically and ecologically sustainable spacecraft, which provide the abilities to be scalable, extendable, maintainable and serviceable on orbit.

However, the development of highly modularized satellites with on-orbit serviceability and reconfigurability is posing a great challenge to space engineers and researchers. The spatial distribution of spacecraft components in different building blocks results in the need for a system intelligence in every module for locally monitoring and controlling of the integrated components and on the same time, for globally coordinating with other building blocks to perform the critical functions and applications of the satellite. On top of this, the desired capability of On-Orbit Servicing extends the already very challenging requirements to the on-board communication system of modular satellites with further demands for flexibility and reconfigurability. Therefore, spacecraft engineers designing a highly modularized satellite need to face the challenge of developing an appropriate on-board data communication system which must provide on the one side, high reliability and fault tolerance and on the other side, the needed flexibility and reconfigurability of the network.

At present, there is no known concept in the state of the art for an on-board data communication system that provides the combination of the required reliability for space applications and reconfigurability needed for modular satellites in one communication system.

The objective of this thesis is to design and develop a concept of an on-board data communication system including methods and mechanisms, which are needed to allow its deployment on highly modularized reconfigurable satellites with the capability of on orbit servicing while the dependability is still kept on the required high level.

In order to achieve the objective, firstly, requirements to the data communication system aboard highly modularized satellites are derived from the high-level system requirements of the modularized spacecraft designs and sharpened by several conceptual considerations, which are necessary for modular systems. A further contribution to the concept is given by a detailed analysis in which several communication standards are assessed to their applicability on modular satellites regarding to the resulted requirements to find the most promising candidate. The next step is to elaborate the necessary mechanisms and methods to fulfil all the requirements of a highly modularized satellite with the capability of On-Orbit Servicing to the on-board data communication system. Lastly, the concept is evaluated and verified experimentally by using the hardware of the modular In-Orbit-Demonstration satellite in the German iBOSS project.

Kurzfassung

Hochmodulare, im Orbit wartbare Satelliten bilden einen innovativen und zukunftsorientierten Designansatz im Bereich der Raumfahrttechnik, welcher hohe Potentiale und viele Vorteile gegenüber klassische monolithische Satellitendesigns aufweist. Modulare Satelliten repräsentieren die nächste Generation der ökonomisch und ökologisch nachhaltigen Raumfahrtsysteme, die neben der Wartbarkeit im Orbit auch skalierbar, erweiterbar und rekonfigurierbar sind.

Jedoch treten durch die modulare Satellitenstruktur eine Reihe neuer Herausforderungen für Raumfahrtingenieure auf. Durch die Verteilung der Satellitenkomponenten in verschiedenen Bausteinen ist eine dezentrale Steuerungsarchitektur des Satelliten notwendig. Die verteilt integrierten Komponenten in den einzelnen Modulen müssen von einer lokalen Systemintelligenz überwacht und kommandiert werden. Gleichzeitig muss eine bausteinübergreifende Koordination zur Ausführung der kritischen Funktionen des Satelliten durch eine zuverlässige, fehlertolerante on-board Datenkommunikation gewährleistet werden. Zudem benötigt die Wartbarkeit und Rekonfigurierbarkeit der modularen Satelliten im Orbit ein flexibles sowie rekonfigurierbares Kommunikationsnetzwerk. Die Vereinigung dieser beiden Eigenschaften, Rekonfigurierbarkeit und Zuverlässigkeit, in einem Kommunikationssystem stellt eine besonders große Herausforderung in der Entwicklung dar.

Nach dem aktuellen Stand der Technik ist kein Lösungskonzept für dieses Problem bekannt, das sowohl die hohe Zuverlässigkeit für Raumfahrtanwendungen aufweist als auch eine flexible und rekonfigurierbare Datenkommunikation erlaubt.

Gegenstand dieser Dissertation ist es daher, ein Konzept für ein Datenkommunikationssystem auf hochmodularen rekonfigurierbaren Satelliten zu erarbeiten. Dieses Konzept vereint in sich die notwendige Flexibilität für die Rekonfiguration im Orbit und die notwendige Zuverlässigkeit sowie Fehlertoleranz für komplexe, verteilte, sicherheitskritische Systeme in Raumfahrtanwendungen. Hierzu werden zunächst anhand konzeptioneller Überlegungen die Systemanforderungen eines modularen Satellitendesigns auf funktionale und nicht funktionale Anforderungen an das Datenkommunikationssystem heruntergebrochen, gefolgt von einer detaillierten Untersuchung der bekannten Busarchitekturen auf ihre Anwendbarkeit in modularen Satelliten mit dem Ziel, einen Kommunikationsstandard als Absprungsbasis zur Konzepterstellung zu bestimmen. Zusätzlich werden erforderliche Mechanismen und Methoden erarbeitet, um die Anforderungen eines hochmodularen rekonfigurierbaren Satelliten an das Datenkommunikationssystem zu erfüllen und das Konzept damit zu komplettieren. Das erarbeitete Konzept wird abschließend mit Hardware des IOD-Satelliten (In Orbit Demonstration) des deutschen modularen Satellitenprojekts iBOSS experimentell evaluiert und verifiziert.

Acknowledgement

I would sincerely like to thank all persons who in some way contributed and helped me to finish this thesis.

First of all, I would like to thank Professor Klaus Brieß for accepting me into his group and for giving me the opportunity to work in the motivating environment of the Chair of Space Technology, Institute of Aeronautics and Astronautics, TU Berlin and the freedom during my research. I am very thankful for his support, guidance, critical feedbacks and last but not least the motivation to finish this thesis. It was really an honour to work under his supervising.

I would like to thank all of my ex-colleagues in the iBOSS team at the TU Berlin, especially my ex-colleagues and friends Lars, Anja and her friend Benno, Thomas and Jens for all the support, the helpful discussions and the motivation for finishing this work.

My special thanks to my ex-colleague and friend Frank, for his countless helpful and valuable discussions and supports in all the years at the Chair of Space Technology.

Further, I would like to thank my friends Benno, David, Jana and Marcus for reading and commenting on this thesis.

I am thankful to my parents, who have never lost their trust they put in me. Dear Dad, you have always been my greatest motivation to take this important step in my life. I thank you, my brother for reading and commenting the thesis. Your helpful comments contributed a lot to my work.

Finally, my special thanks to my wife and my children for all the support and encouragement during this exciting and sometimes difficult time. Thank you, my beloved Thuy Ly, for taking care of everything, especially our two wonderful children, Kien and My, and for allowing me to focus on my thesis. I greatly appreciate your support and patience, especially during the final phase of this thesis. This thesis would not have been possible without you. Thank you!

TABLE OF CONTENTS

1 INTRODUCTION 1

1.1 Motivation 3

1.2 Objectives and contributions of this work 4

1.3 Thesis outline 5

2 FUNDAMENTALS AND A STATE-OF-THE-ART ANALYSIS 7

2.1 Data Communication aboard spacecraft 7

2.1.1 On-board data networks 9

2.1.2 Communication Reliability 11

2.1.3 Deterministic and real-time communication 12

2.2 Modular spacecraft projects 14

2.2.1 MMS 15

2.2.2 ROSE 16

2.2.3 ARAMIS 18

2.2.4 PHOENIX 20

2.2.5 IBOSS – A modular satellite design for OOS 23

2.2.5.1 The main idea of iBOSS 24

2.2.5.2 iBLOCK - the building block of iBOSS 25

2.2.5.3 iSSI - The multifunctional interface 27

2.3 Data communication systems in modular robots 27

2.4 Dependability of the data communication network aboard modular spacecraft 29

2.5 Concluding remarks 32

3 DATA COMMUNICATION CONCEPT FOR MODULAR SATELLITES 34

3.1 Conceptual Considerations for the On-board Data Bus System in modular satellites 34

3.1.1 Analysis of physical network topology for modular satellites 34

3.1.2 Logical communication schemes for modular satellites 36

3.1.3 Separation of P/L data traffic and HK and Satellite control data traffic 39

3.1.3.1 On-board data communication on reference satellite missions 40
3.1.3.2 Physical separation of data traffic types 42
3.1.3.3 Logical separation of data traffic types 43
3.2 Requirements to the data communication architecture of modular satellites 43
3.3 Comparison of data communication systems 46
3.4 Application of TTEthernet as the global data bus on modular spacecraft 53
3.5 The scheduling problem of the on-board data network of modular spacecraft 54
3.6 Concluding remarks 56

4 HARDWARE CONCEPTS FOR THE DATA COMMUNICATION SYSTEM 58
4.1 Network hardware in the module 58
4.2 The Optical Data Interface 59
4.2.1 Data transmission method between building blocks 59
4.2.2 Requirements of the Data Interface 60
4.2.3 Hardware design of the Optical Data Interface 61
4.2.3.1 Data Interface Backend 63
4.2.3.2 Data Interface Frontend 64
4.2.4 Functional tests of the Data Interface 65
4.2.5 Characteristics of the Data Interface 67
4.2.6 Testing of the Data Interface toward radiation hardness 68
4.3 The OBC-Switch-Board – A multifunctional component 69
4.3.1 Hardware design of the OBC-Switch-Board 70
4.3.2 Design aspects to enhance the dependability of the OBC-Switch-Board .. 72
4.4 Concluding remarks 74

5 NETWORK CONCEPTS FOR THE ON-BOARD DATA COMMUNICATION OF HIGHLY MODULARIZED SATELLITES 76
5.1 Engineering of the reconfigurable data network 76
5.1.1 Scheduling of the on-board data communication 77
5.1.1.1 Physical and logical network model 78
5.1.1.2 Decision variables and constraints of the network schedule 79
5.1.1.3 Network configuration using the tool package ^{TTE}Tools of TTTECH 81

5.1.1.4 Network configuration for a 2x2 iBOSS modular satellite 83
5.1.2 Loop-handling concept of the communication data network 88
5.2 Reliability enhancement of the on-board data communication system 89
5.2.1 Analysis of failures 90
5.2.2 Detection of failures 94
5.2.3 Network recovery in case of failures 98
5.2.3.1 Network recovery in case of a disruptive data link between two building blocks 98
5.2.3.2 Network recovery in case of a link failure inside a building block 99
5.2.3.3 Network recovery in case of an OBC failure inside a building block 99
5.2.3.4 Network recovery in case of an TTE-Switch failure inside a building block 100
5.3 Reconfigurability of the on-board data networks during OOS scenarios 100
5.3.1 OOS-Reconfiguration 103
5.3.2 OOS-Exchange and OOS-Removal 105
5.3.3 OOS-Extension 106
5.4 Concluding remarks 107

6 EVALUATION OF THE PROPOSED CONCEPTS 108
6.1 Experimental model of the data network aboard a small modular satellite 108
6.2 Evaluation of the communication scheduling approach 110
6.3 Evaluation of the concept for network recovery in case of failures 116
6.3.1 Network recovery in case of a link failure between two building blocks 116
6.3.2 Network recovery in case of a link failure between two building blocks 118
6.4 Network reconfiguration for OOS use cases 119
6.4.1 OOS-Reconfiguration 119
6.4.2 OOS-Extension 120
6.4.3 OOS-Exchange and OOS-Remove 121
6.5 Concluding remarks 121

7 CONCLUSIONS AND OUTLOOK 123
7.1 Conclusions regarding the objectives of the thesis 124
7.2 Contributions to iBOSS 126
7.3 Contribution outside of iBOSS 126

7.3.1 Application of the proposed data communication system in terrestrial industry e.g. the automotive industry 126
7.3.2 Application of the data communication system in modular space robots and space rovers 126
7.4 Outlook and Future works 127

APPENDIX **129**

Comparison of communication architectures to their applicability on modular satellites 129

REFERENCES **136**

LIST OF FIGURES

Figure 1.1: Computer generated model of space debris in GEO [6] ... 1
Figure 1.2: Artist's illustration of a OOS scenario in the German modular satellite project iBOSS [10] ... 2
Figure 2.1: An overview of hard- and software architecture of a spacecraft [12] ... 8
Figure 2.2: Different network topologies ... 10
Figure 2.3: Categories of MAC protocols and their application in bus standards [21] ... 13
Figure 2.4: Modularity Spectrum ... 15
Figure 2.5: MMS component structure [26] ... 16
Figure 2.6: ROSE Spacecraft overview [24] ... 17
Figure 2.7: ROSE Module Unit Interface [24] ... 17
Figure 2.8: ROSE avionics architecture – including two ACS modules, a RWA module, a C&DH module and a power module [24] ... 18
Figure 2.9: AraMIS 2x2x2 satellite concept [28] ... 19
Figure 2.10: AraMIS on-board data bus concept [28] ... 20
Figure 2.11: HISat – Hyper-Intergrated Satlet [38, 41] ... 21
Figure 2.12: Novawurks' vision of modular satellites built up by HISats [41, 42] ... 22
Figure 2.13: Launched satellites based on the HISats [45–47]: a) SIMPL (launched in 2015), b) eXCITe (launched in 2018), c) PODSAT 1 (launched in 2018) ... 23
Figure 2.14: Classification of components to subsystems for the modular design in iBOSS [50] ... 24
Figure 2.15: iBOSS modular spacecraft concepts and cube-shaped building blocks [10] ... 25
Figure 2.16: Standard components of an iBLOCK [49] ... 26
Figure 2.17: Software stack of iBOSS [48] ... 26
Figure 2.18: The multifunctional interface iSSI [7, 8] ... 27
Figure 2.19: Systematic exposition of the dependability concept [55] ... 30
Figure 3.1: Illustration of an iBOSS modular satellite with 2x2x2 configuration ... 35
Figure 3.2: Different possible network topologies for the exemplary modular satellite configuration ... 35
Figure 3.3: Different logical communication architectures within a modular system ... 37
Figure 3.4: Typical satellite system production flow [58] ... 39
Figure 3.5: Functional block diagram of the EnMAP satellite bus [59] ... 40
Figure 3.6: Conceptual view of SGEO subsystem electrical interfaces and network protocol stack [60] ... 41
Figure 3.12: Exemplary transmission of the 3 data types in TTEthernet ... 53
Figure 3.13: Acyclic directed graph of the ACS service example ... 55

Figure 4.1: Block diagram of the network hardware and the components inside of a building block 58
Figure 4.2: Overview of data transmission technologies known in terrestrial applications [13, 49] 60
Figure 4.3: Components of the data interface 62
Figure 4.4: Block diagram of the connection between the data interface and the network switch [48] 63
Figure 4.5: Connectors of the DIF backend PCB 63
Figure 4.6: Thermal concept of the DIF backend PCB 64
Figure 4.7: Front side and back side of the DIF frontend PCB 64
Figure 4.8: Dimensions of the DIF frontend [48] 65
Figure 4.9: Coordinate system of the test setup for the data interface [48] 65
Figure 4.10: Test setup of the data interface [48] 66
Figure 4.11: Measured data rate of rotated data interface as the proof for rotation symmetry [48] 66
Figure 4.12: Measured data rate of the data interface by alignment in x- and y-direction [48] 67
Figure 4.13 Measured data rate over z-translation [48] 67
Figure 4.14: The radiation test results of the data interface [48] 69
Figure 4.15: The OBC-Switch-Board [96] 69
Figure 4.16: Block diagram of the OBC-Switch-Board [48] 71
Figure 4.17:Maximum expected power dissipation of the OBC-Switch-Board [48] 71
Figure 4.18: Bottom and upper side of the OBC-Switch-Board with references for the mounted components [48, 96] 72
Figure 4.19: Electrical block diagram of the architecture of the OBC-Switch-Board [48, 62] 73
Figure 4.20: Grounding concept of the OBC-Switch-Board [48] 74
Figure 4.21: Thermal concept of the OBC-Switch-Board [48] 74
Figure 5.1: Block diagram of a small 2x2 modular satellite based on the iBOSS concept [7–9] 77
Figure 5.2: Scheduling process of a reconfigurable modular satellite 79
Figure 5.3: Overview of the network configuration process using ^{TTE}Tools 82
Figure 5.4: The physical network topology of the exemplary modular satellite 83
Figure 5.5: Communication schedule of the exemplary network on the 2x2 modular satellite 87
Figure 5.6: Network loops on a 2x2x2 iBOSS modular satellite configuration 88
Figure 5.7: Network components and devices of a module regarded as funtional elements 90

Figure 5.8: Exemplary modular satellite configurations with weak designs: a) A configuration with a hermit module; b) a configuration with a hermit cluster and several hermit modules 92

Figure 5.9: The critical TTE-Switch failure in the on-board data network of a modular satellite 95

Figure 5.10: Algorithm to check link health to neighbouring modules implemented on the OBCs of the modular satellite 96

Figure 5.11: Algorithm for failure detection and classification implemented in the health masters 97

Figure 5.12: Recovery of the communication network with a link failure by rescheduling (Dashed lines – Physical links; full lines – scheduled BE and CT data paths; red line - defective link. 98

Figure 5.13: Recovery of a network with a TTE-Switch failure 100

Figure 5.14: Artist's illustration of an on-orbit servicing mission in iBOSS project [10] 101

Figure 5.15: Flow diagram to check for rescheduling necessity of OOS use cases 102

Figure 5.16: Exemplary reconfiguration of a modular satellite 103

Figure 5.17: Sequence diagram of the OOS-reconfiguration 105

Figure 5.18: Exemplary scenario of an OOS-Extension of a modular satellite 107

Figure 6.1: Planned iBOSS IOD mission with 4 iBLOCKs (Artist's depiction) [10] 108

Figure 6.2: CAD model of the EQM and block diagram of the on-board data network 109

Figure 6.3: Experimental setup of the data communication network on the iBOSS IOD satellite 110

Figure 6.4: Bandwidth utilization for each physical link of the on-board data network of the iBOSS IOD satellite 114

Figure 6.5:Network schedule for the on-board data communication of the iBOSS IOD satellite 115

Figure 6.6: Simulation of a link failure in the experimental setup by removing a connecting cable 116

Figure 6.7: Network recovery after a link failure between BB1 and BB4 117

Figure.6.8: Experimental simulation of a failure on BB4 118

Figure 6.9: Network recovery after a node failure on BB4 118

Figure 6.10: Ethernet link to the PC via RJ45 interface of the OBC on BB2 119

Figure 6.11: Reconfiguration of the exemplary network configuration of the IOD satellite 120

Figure 6.12: Extension of the exemplary network of the IOD satellite with a further building block 120

Figure 6.13: Experimental setup with 5 OBC-Switch-Boards 121

Figure 7.1: Artist's conception of space rover built of iBOSS modules [10] 127

Figure 7.2: Artist's conception of iBOSS' future vision [10] 127

LIST OF TABLES

Table 2.1: Properties and requirements of on-board data traffic [13] 8
Table 2.2: Criticality range of functions aboard spacecraft [14] 9
Table 2.3: Summary of intramodular and intermodular communication in modular robots [53] 28
Table 3.1: Comparison of local and global data bus scheme 38
Table 3.2: Comparison matrix of the communication architectures [8, 9, 62] 51
Table 4.1: Characteristics of the DIF [48] 68
Table 4.2: Specifications of the OBC-Switch-Board [96] 71
Table 4.3: Components of the OBC-Switch-Board [48, 96] 72
Table 5.1: Physical and logical model of the exemplary communication network 84
Table 5.2: Scheduling results of the exemplary communication network for a 2x2 iBOSS modular satellite 85
Table 5.3: Categorization of failures of network elements in one module 92
Table 6.1: CT jobs of the four building blocks of the modular IOD satellite 111
Table 6.2: File sizes of the binary files for network devices 112
Table 6.3: Results of the network scheduling for the IOD satellite 113

ABBREVIATIONS

ACS	Attitude Control System
AIAA	American Institute of Aeronautics and Astronautics
AIT	Assembly, Integration and Test
AOCS	Attitude and Orbit Control System
ARAMIS	Italian acronym for modular architecture for small satellites
ASIC	Application-Specific Integrated Circuit
A/D	Analog/Digital
BAG	Bandwidth Allocation Gap
BB	Building Block
BE	Best-Effort
CA	Collision Avoidance
CAN	Control Area Network
CD	Collision Detection
CMG	Control Moment Gyroscope
COTS	Components off the Shelf
CR	Collision Resolution
CSMA	Carrier Sense Multiple Access
CT	Critical Traffic
C&DH	Command and Data Handling
DAG	Directed Acyclic Graph
DARPA	Defense Advanced Research Projects Agency
DIF	Data Interface
DLR	Deutsches Zentrum für Luft- und Raumfahrt (German Aerospace Center)
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
EPS	Electrical Power System
ES	Endsystem
GEO	Geosynchronous Orbit
GPIO	General Purpose Input/Output
HISat	Hyper-Integrated Satlets
HK	Housekeeping
HST	Hubble Space Telescope
iBOSS	intelligent Building Blocks for on-Orbit Satellite Servicing and Assembly
ICMP	Internet Message Control Protocol
ICU	Interface Control Unit
IMU	Inertial Measurement Unit
IOD	In Orbit Demonstration
IR	Infrared
ISRO	Indian Space Research Organisation

ISS	International Space Station
iSSI	intelligent Space System Interface
I/F	Interface
I&T	Integration and Test
I^2C	Inter-Integrated Circuit Bus
JAXA	Japan Aerospace Exploration Agency
JTAG	Joint Test Action Group
LEO	Low Earth Orbit
LEOP	Launch and Early Orbit Phase
LVDS	Low Voltage Differential Signalling
MAC	Media Access Control
MIU	Module Interface Unit
MMS	Multimission Modular Spacecraft
NASA	National Aeronautics and Space Administration
OBC	On-Board Computer
OOA	On-Orbit Assembly
OOS	On-Orbit Servicing
OSI	Open Systems Interconnections
PAC	Package of Aggregated Cells
PCB	Printed Circuit board
PCDU	Power Control and Distribution Unit
PCI	Peripheral Component Interconnect
PCU	Power Control Unit
PDU	Power Distribution Unit
P/L	Payload
QSPI	Quad Serial Peripheral Interface
RC	Rate-constrained
RF	Radio Frequencies
ROS	Robot Operating System
ROSE	Reconfigurable Operational Spacecraft for Science and Exploration
RTU	Remote Terminal Units
RVD	Rendezvous and Docking
RWA	Reaction Wheel Assembly
R&D	Research and Development
SNMP	Simple Network Management Protocol
SMM	Solar Maximum Mission
SoM	System on Module
SPI	Serial Peripheral Interface
SPOF	Single Point of Failure
SpW	SpaceWire
SSN	Space Surveillance Network
SW	Switch
S/C	Spacecraft

TCS	Thermal Control System
TDMA	Time Division Multiple Access
TFTP	Trivial File Transfer Protocol
TM/TC	Telemetry/Telecommand
TT	Time-Triggered
TTA	Time-triggered Architecture
TTCAN	Time-triggered CAN
TTE	TTEthernet
TTEthernet	Time-triggered Ethernet
TT&C	Telemetry Tracking and Command
UART	Universal Asynchronous Receiver Transmitter
VL	Virtual Link
WCET	Worst-Case Execution Time
XML	Extensible Markup Language file

1 INTRODUCTION

In the last decades, several trends have essentially shaped the development in the space engineering. Cheaper and faster access to space and spaceflight technologies has been strongly driven by the post-millennial global NewSpace movement. Along to that, increasing commercialization and governmental using of private space launcher systems accompanied by the dynamic development in the computer chip technology and the miniaturizing trend in the space technology have pushed up the amount of the launched satellites for the last two decades to a recordable number. Exemplary for that, in 2017, the Indian space agency ISRO successfully launched 104 satellites in one single mission [1, 2]. In September 2020, the US company SpaceX launched 60 satellites on a Falcon 9 rocket and filings for 30,000 additional satellites have been submitted beside of the 12,000 approved satellites for establishing the Starlink, a satellite internet constellation to enable a global broadcast service [3, 4].

This rapid development intensifies the problem of space debris, which has become one of the central issues of the space industry since the nineties. The steadily increased number of space debris in the low earth orbit (LEO) and the geostationary earth orbit (GEO) poses a growing threat for the operational satellites, spacecraft and space stations. At present, roughly 22,300 of these objects are regularly tracked and catalogued by the U.S. Department of Defense's Space Surveillance Network (SSN). According to estimations using statistical models, there are around 34,000 objects greater than 10cm and up to 900,000 objects with size between 1cm and 10cm [5, 6].

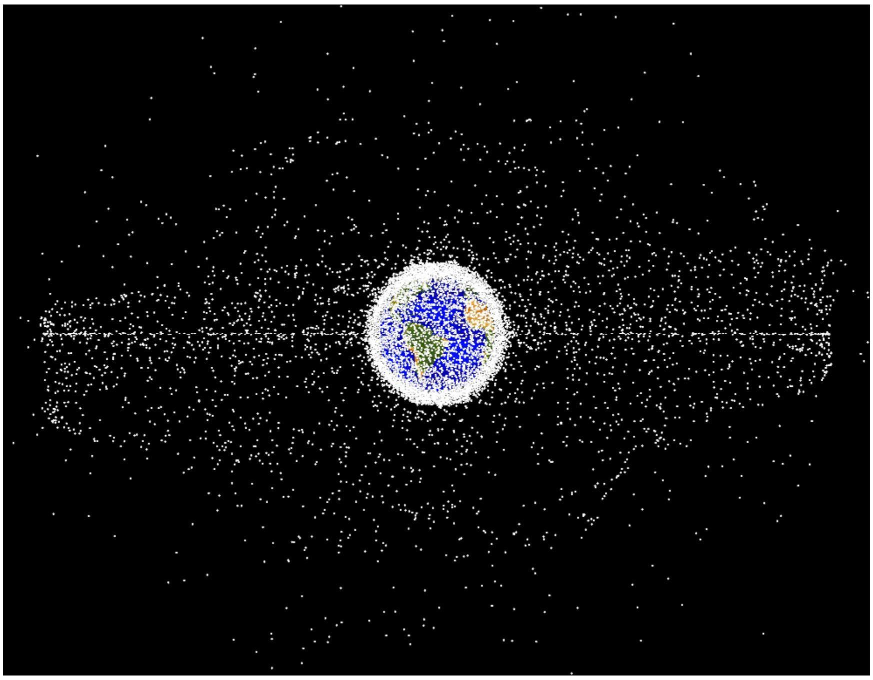


Figure 1.1: Computer generated model of space debris in GEO [6]

To address this growing problem of space debris, several approaches for reducing further space debris are discussed by international researcher groups. One of the promising approaches can be found in the sustainable measurements in designing and operation of satellites and spacecraft. However, the sustainability of the traditional spacecraft and satellites are limited. Despite of the newest technological achievements, their traditional monolithic architecture reduces satellites to highly expensive, single-use products built around one or more dedicated payload systems designed to fulfil specified mission objectives.

In the last few years, modular satellite designs with the capability of on-orbit servicing have been getting one of the high promising approaches in space engineering with high potential and great benefits in mid-term and long-term future. Especially, highly modularized and reconfigurable satellites comprising of building blocks with standardized form factor but individual functionality opens new opportunities to a sustainable spacecraft development and operation. Such satellite designs provide features like the ability to be reconfigured for new missions after a successful mission accomplishment, to be maintained by replacing faulty components by new ones, to be upgraded with newer generations of hardware or even to be refuelled to extend the lifetime on orbit. Additional to that, the standardization of the building blocks and interfaces between them is an integral part in the development, production and qualifying processes of the modular satellites and implicates significant time and cost saving as well as the rapid-production-on-demand capabilities. These economic advantages are also accompanied by ecological benefits regarding to the mentioned problem of space debris and increase substantially the sustainability of the future spacecraft and space missions [7–9].

A highly modularized satellite consists of several standardized modules connected by standardized interfaces with the capabilities of mechanical coupling, transfer of electrical, thermal energy and data transfer between the modules. Such highly modularized satellite designs are proposed in the German project iBOSS or in the PHOENIX program of the US Defense Advanced Research Projects Agency (DARPA). They form the basis for On-Orbit Servicing (OOS) and On-Orbit Assembly (OOA) missions, respectively reconfiguration, maintaining, upgrading and mission redefinition on orbit.

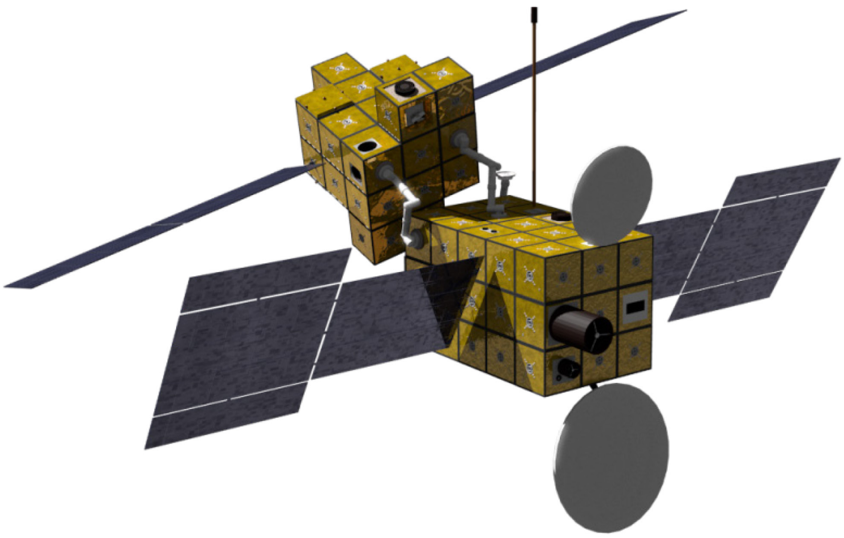


Figure 1.2: Artist's illustration of a OOS scenario in the German modular satellite project iBOSS [10]

The approach of modularizing of satellites begins with the decomposition of a traditional satellite system on component level or on subsystem level and the distribution of the components of a satellite in different modules. Thus, a modularized satellite bus with all necessary subsystems like Attitude and Orbit Control (AOCS), Thermal Control (TCS), Electrical Power (EPS), etc., is a composition of several modules connected with each other. In order to monitor and control the locally integrated components, each module needs to provide a system intelligence for local data handling activities like sensor reading, controlling the actuators, routing the data and information needed for satellite operating if necessary, from one end of the satellite through several modules to the opposite end. From the point of view of communication and data handling, the modular satellite can be regarded as a highly complex, distributed and embedded system with decentral control architecture, whose system intelligence is partitioned in several locally distributed processors and whose performance is indicated by the inter-process and inter-processor communication of information [7, 9].

1.1 Motivation

Due to the reconfigurability and serviceability in operation on-orbit, the compliance of the requirements of reliability, fault tolerance, composability in space systems are additionally extended by demands for high reconfigurability and flexibility of the data network. Comparing to the classical monolithic satellite designs, where the data handling for the operating of a satellite system is centralized in the On-board Computer (OBC) and the global data bus, the communication architecture of modularized spacecraft systems must provide mechanisms and functions for the coordination of locally distributed devices and components (e.g. distributing sensor reading and actuator commands) and for the prevention of fault propagation from one application to another one. Thus, the requirements to a fault tolerant and reliable communication of data from one distributed component to another is an issue of major importance in the design of a modular spacecraft and even harder than on monolithic satellites.

For that reason, several well-known control architectures integrated in classical monolithic satellite systems cannot be applied without extensive adaption efforts. Especially with respect to the on-orbit reconfigurability, extensibility and maintainability, the formerly proven bus communication technologies in monolithic satellite systems are not adaptable on modular satellites. At present, no proven concept of an appropriate on-board data communication system for highly modularized satellites with the capability of on-Orbit Servicing is known in the state of the art.

In summary, development of an appropriate on-board data communication system for highly modularized satellites with capability of OOS must regard the following aspects beside of the standard design aspects for space systems:

- Highly modularized spacecraft are regarded as a highly complex, distributed system with mixed criticality requirements to the data traffic. The on-board data communication architecture shall provide mechanisms and functions to support the on-board data traffic with different criticality levels.

- The on-board data communication system of modularized spacecraft shall provide the needed reliability for space applications and services to support fault containment for the prevention of fault propagation from module to module of the spacecraft.
- The on-board data communication system shall provide high flexibility and reconfigurability of the data network to meet the requirements of OOS.

In the field of modular spacecraft, many research programs have been presented especially in the last two decades. The most known among them is surely the DARPA's PHOENIX program with three launched modular satellites up to date. Beside of that, several concepts and research works have been proposed, e.g. ROSE - Reconfigurable Operational Spacecraft for Science and Exploration, AraMIS - acronym for Modular Architecture for Satellites in Italian, the German iBOSS project and others. However, according to the author's best knowledge, no contribution exists that provides a concept for an on-board data communication system for highly modularized satellites, which meets the mentioned requirements above.

1.2 Objectives and contributions of this work

The main goal of this thesis is to develop and apply a concept of an appropriate on-board data communication system for highly modularized and reconfigurable satellites with the capability of OOS. Following objectives are set for this thesis:

- Analysis of the requirements of a highly modularized spacecraft architecture and break them down to dedicated requirements and system design aspects for the on-board data communication system beside of the standard requirements for reliability and fault tolerance to the on-board communication of space systems.
- Studying the proven or potential communication standards and comparing them with each other towards their applicability on a highly modularized satellite with OOS capability.
- Development of a concept for the on-board data communication system of highly modularized spacecraft including the needed network hardware and software.
- Verification of the proposed concept and all provided mechanisms and functions to fulfil the specific system and architecture requirements of modular spacecraft (e.g. reconfigurability, reliability, loop handling, etc.).

In order to achieve the objectives of this dissertation, following research questions must be answered:

1. Which dedicated features and characteristics are needed for the data communication system of highly modularized satellites with OOS capability beside of the standard requirements to data communication system on-board spacecraft and which constraints are given by the modular architecture?
2. Which communication architectures can be applied on highly modularized satellites with OOS capabilities?
3. What are the major challenges of the development of an appropriate on-board data communication system for modular reconfigurable satellites and how can the identified system level design challenges be solved?

For answering the research questions, firstly, requirements to the on-board data communication system on modular satellites will be broken down from the high-level system requirements of the highly modularized satellite designs. An overview of the bus standards applied on spacecraft as well as the data communication concepts of modular robots will be described out and compared. Based on the conceptual considerations derived from the requirements, the concept for the on-board data communication system including needed network hardware and network management software development is elaborated. The concept shall provide communication protocols containing defined mechanisms and functions to coordinate the data communication of locally distributed devices and components allocated in several building blocks of a modular satellite, to transfer satellite control and housekeeping data traffic with different criticality levels on a reliable manner and, on same time, to have enough bandwidth and throughput to deal with large payload data amount. Additionally, the proposed on-board data communication system shall be designed to meet the challenges of the different scenarios of on-orbit satellite servicing like the reconfiguration of the building blocks of the modular satellite, the exchange of building blocks with new ones and the adding or removing of building blocks without neglecting the network stability required for a spacecraft in operation. Lastly, the concept of the on-board data communication system for modular satellites with the presented methods and mechanisms will be evaluated and verified by an experimental setup of the data network on the iBOSS In-Orbit-Demonstration (IOD) satellite. Different OOS operations of an iBOSS modular satellite will be simulated from the point of view of the data communication.

Since the idea for this thesis has been raised during the iBOSS project, knowledge gained during the project have been used in this work and the proposed concept is verified by using iBOSS hardware.

1.3 Thesis outline

The thesis is structured as follows. Chapter 2 encompasses the relevant fundamentals of the data communication on spacecraft and provides an overview of the known modular concepts for satellites and of different concepts for modular robots as an analysis of the state of the arts. In Chapter 3, an extensive comparison of communication bus architectures is accompanied by several conceptual considerations for designing of a data network on modular spacecraft. Chapter 4 describes the design and development of network hardware regarding the dependability needed for modular satellite designs. In Chapter 5, the elaborated final concept for the on-board data communication system of highly modularized satellites with OOS capabilities is presented and complemented with the needed methods to enhance dependability and procedures for OOS scenarios. Chapter 6 contains the description of the experimental setup and the evaluation of the proposed concept and accompanying methods and approaches. The results and contributions of the thesis are summarized and discussed in Chapter 7 with the outlook on further works.

2 FUNDAMENTALS AND A STATE-OF-THE-ART ANALYSIS

This chapter provides the fundamentals relevant in this work and an analysis of state-of-the-art. Beginning with Section 2.1, the basics of on-board data communication on spacecraft with the focus on deterministic and reliable communication are described and the relevant terminology explained. Following to that, known concepts of modular satellites are presented and discussed with special regard to their deployed on-board communication systems in Section 2.2. Since the approaches proposed in this thesis are elaborated during the project iBOSS, the iBOSS concept is presented with more details in Section 2.2.5. Due to their conceptual similarity to modular spacecraft, in Section 2.3, an overview of the applied data communication systems for modular robots is presented and discussed. The term dependability is discussed extensively with the focus on applying methods to enhance dependability of the on-board communication system of modular satellites in Section 2.4 and the chapter is concluded with the concluding remarks in Section 2.5.

2.1 Data Communication aboard spacecraft

On a spacecraft, the on-board Command and Data Handling (C&DH) subsystem can be regarded as the “brain” of the spacecraft whose main tasks are defined as follow:

- Reception, execution and sending of the telecommands from the ground station operator to one or more subsystems on-board and storage of time-tagged or position-tagged telecommands and execution or forwarding them to the relevant subsystems at the defined time or position,
- Acquisition and processing of sensors data and information needed for controlling and operating the spacecraft,
- Preparing, storing and sending spacecraft telemetry data via downlink to the ground station,
- On-board communication management,
- Time management and distribution of the spacecraft time,
- Autonomously monitoring and responding to on-board problems that might occur,
- Data handling of mission payloads if no separate payload data handling is deployed.

The on-board C&DH subsystem comprises Telemetry and Telecommand modules (TM/TC), On-Board Computer (OBC), Data Storage and Mass memories, Remote Terminal Units (RTU), communication protocols and data busses [11].

An essential part of the C&DH subsystem is the on-board data bus which form the basic infrastructure for performing the tasks of spacecraft controlling and operating. In the following subsection, the different network topologies will be described and discusses regarding to their adaptability on modular spacecraft.

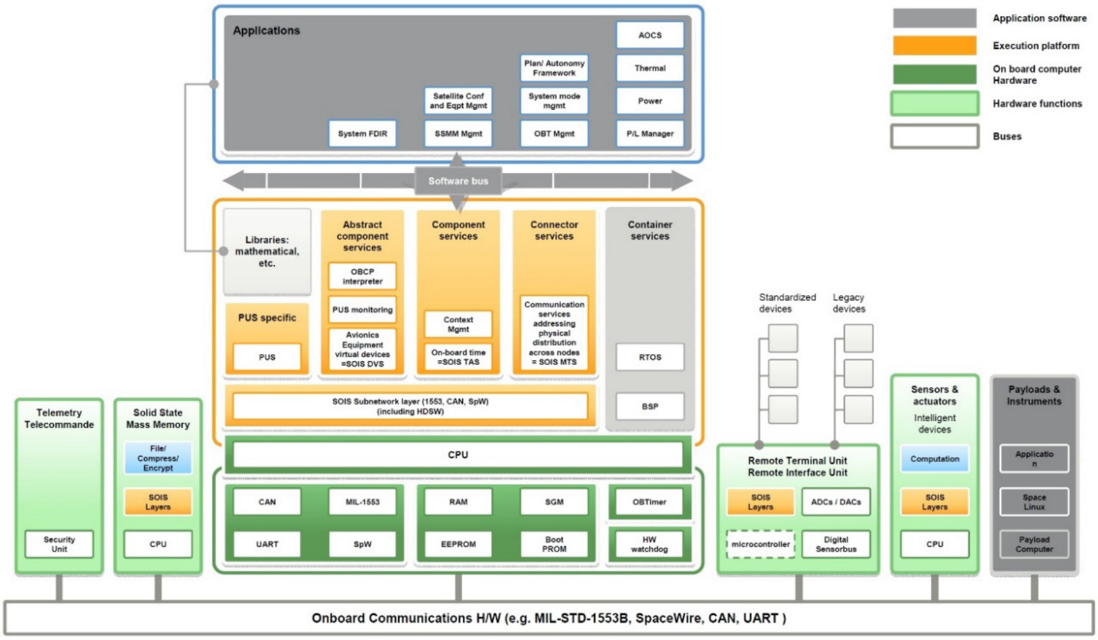


Figure 2.1: An overview of hard- and software architecture of a spacecraft [12]

Typically, the data traffic on-board spacecraft are distinguished in two different main categories, housekeeping and spacecraft control data traffic, e.g. the transfer of sensor signals and actuator commands, and the traffic of payload data, e.g. images from a payload camera system. These both types of data traffic involve different requirements to the communication network as characterized in Table 2.1.

Table 2.1: Properties and requirements of on-board data traffic [13]

	Housekeeping and control data traffic	Payload data traffic
Application/Example	• Sensor reading and actuator commands of AOCS, TCS, PCDU, etc. • Communication interface to TM/TC • Redundancy management	• High resolution images, videos from payload camera systems • Measurement data
Required data rate	Less than 1 Mbps	More than 100 Mbps
Real-time communication	needed	Not needed
Recommended communication protocols	Time-triggered	Event-triggered
Other requirements	Fault tolerant, safety critical, low latency, reliability	High data throughput
Challenges	Time synchronization, limited flexibility, etc.	High processing capacity of the OBC or separated P/L data handling processors

The functions aboard the spacecraft that generate data traffic can be divided in a hierarchy of different criticality range from safety critical (loss of function means loss of spacecraft) to mission critical (loss of function means failure of the mission) to low critical (spacecraft maintenance decision after mission is over) or no critical.

Table 2.2: Criticality range of functions aboard spacecraft [14]

Application types	Data rate	Criticality
Sensor bus – transducer systems – For monitoring many nodes	low	Non-critical
Audio data (info multimedia)	Low	Non-critical
TM interfaces to science instruments Video data Visual monitoring cameras	High	Non-critical
Sensor system/platform bus Thermal, Power monitoring, S/C health sensing	Low	Low to high criticality as function of the situation in the system and its redundancy policy
Sensors, Actuators, Space link	Low	Reliability critical
Guidance sensor interface	Medium	Availability critical
Platform bus Control and Command	Low	Reliability/Availability critical
RVD I/F Operational audio/video	medium	Safety critical

2.1.1 On-board data networks

The architecture of a data communication system, whether centralized or distributed, incorporates physical and logical frameworks used to interconnect communication nodes. A data communication network can have different topological structures. One distinguishes two categories of network topologies, the physical topologies and the logical topologies.

The logical topology is described by the network protocols and specify the networking concept defining the communication mechanisms for all network nodes without regard to the physical interconnections. They are often closely associated with media access control methods and standards.

Contrary to that, the physical network topologies only refer to the physical pattern of the network interconnections. Some typical physical topologies of spacecraft data networks are listed below [15, 16].

- **Linear bus topology:** Networks with a linear bus topology typically consist of a standardized, share, linear data bus to which all spacecraft subsystems or components are connected. Whereas on typical spacecraft, the wiring harness is minimized and connection error of one network node have no impact on the network connectivity, it requires a well-designed communication protocol for governing the bus access to pretend collision or babbling-idiot failure.

- **Ring topology:** The interconnected nodes of a network with ring topology form a closed chain. Every network node is connected to two neighbour nodes. Ring topology typically requires less wiring harness and minor dependencies among subsystems. Compared to the linear bus topology, network with ring topology may need longer data transmission time between nodes. But usually, a ring topology with bi-directional data transmission can show a higher reliability than the linear bus topology because of the dual connection lines of the network nodes.
- **Star topology:** The star bus topology contains a central network node which is connected to all other nodes and takes over the network management function. On spacecraft, this node typically represents an OBC or a centric data switch. The star bus topology requires more wiring harness and is prone to the failure of the central node. From a computational point of view, the star topology is the example of a centralized communication architecture.

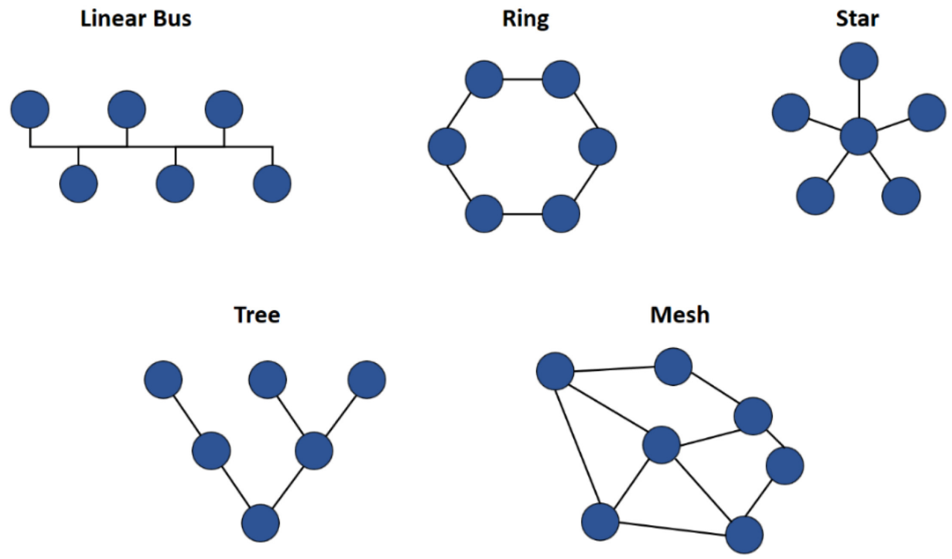


Figure 2.2: Different network topologies

- **Tree bus topology:** The tree topology has at least three levels of hierarchy, otherwise it is a star topology. It differs from the star topology by distributing the role of the central node to several nodes on several layers of the tree. A failure in one node cannot disable the whole network but it can cause that the network section below this node is disconnected to the rest of the system.
- **Mesh topology:** The main advantage of the mesh topology is obviously the ability to provide redundant data paths between nodes. Together with the ring topology with bidirectional data transmission it provides inherent redundancy for network communication to mitigate node or link failure. If the redundant data paths are used simultaneously, the maximum network traffic volume can be noticeably increased. On the other side, this topology requires a lot of wiring

harness and contains network loops which can be a problem for some communication protocols.

2.1.2 Communication Reliability

Generally, reliability is the most important constraint upon the design of spacecraft on-board data handling. Although modular architecture designs have the potential to change the mass-cost-performance equation and to minimize the recurrent engineering costs and the costs for assembly, integration and test (AIT) on mid- and long-term, a space system still represents a high cost investment. Except some short-term technical demonstration missions, space systems are usually intended to survive and to operate for several years in an environment, where repairing and maintenance are either not possible or are associated with high costs and efforts. The data communication system aboard spacecraft which is responsible for the data exchange between the spacecraft subsystems can be regarded as an important system-critical part of any spacecraft comparable to the neural network in a human body. Designers of such a reliable communication system shall consider the following criteria [15, 17]:

- Analysability
- Predictability
- Testability
- Extensibility
- Fault tolerance
- Resource utilization

In general, it can be distinguished between two types of communication architectures, time-triggered and event triggered communication.

In an event-triggered communication network, activities e.g. message generation and message transmission are initiated by the occurrence of significant events in the system. The process in that a message is generated by a node and transferred to another node or to several nodes is based on a defined non-temporal event in the network, for example, when a node requests sensor information from another node. These messages are sent based on an event that can occur at any time with no discernible regularity. The best-known event-triggered system is the Ethernet standard [18–20].

In a time-triggered system, bus access by a node is predefined by a time schedule. Activities are initiated at predefined time points referenced by a global time base and each node of the network is given a finite time slot, during which messages can be transmitted. This sequence of time slots in the schedule is repeated periodically leading to the advantage of a quasi-deterministic behaviour during regular operation. The global time base can be either controlled by a bus master or by masterless distributed time synchronization [19]. But time-triggered systems also have some disadvantages. One noticeable disadvantage is the significant amount of needed design effort for generating the time schedule model. Another disadvantage is that a strict configuration of the system is imposed. If the physical network topology is changed by adding or removing of one or more nodes, the time schedule needs to be completely recalculated. Furthermore, if the time schedule is not optimized, it can happen that many reserved time slots exist even when the nodes have no information to send. Thus, the design of the time schedule model of a time triggered network can be regarded as a key factor for the efficiency of the bus loading. The major advantage of time triggered systems is the determinism of the data communication [18–20].

For safety critical systems like spacecraft, the following conclusion can be drawn, for spacecraft data bus applications where reliability is one of the most important design specifications and resource are limited, communication architectures with time-triggered protocols are preferred over those with event-triggered protocols for regular or time-critical data traffic. For non-regular data traffic initiated by the sporadic and aperiodic occurrence of events, the event-triggered protocols shall be preferred.

2.1.3 Deterministic and real-time communication

On highly modular spacecraft with distributed system intelligence, where several system operations are accomplished simultaneously by spatially distributed processors in several modules, the reliability and determinism of the on-board intermodular communication belong to some of the most important key design specifications of the spacecraft. Though the modular spacecraft architectures increase the system's capabilities and versatility, the design complexity heightens the dependencies between the modules and thus, the occurrence of critical errors. Therefore, it is very important that the data communication system on modular spacecraft shall be able ensure the transmission of critical data in time. This goal can only be achieved by using reliable, fault-tolerant network technologies and have the necessary mechanisms implemented to take care of network failures like preventive maintenance, fault diagnosis and isolation and malfunction handling.

As a fact, it is difficult or nearly impossible to support fault tolerance in safety-critical applications without determinism. Deterministic operation is a precondition for the certification of communication standards for safety critical systems. The definition of determinism applied for real-time communication is: A deterministic system is a system whose time evolution can be predicted exactly. Any communication architecture that uses arbitration, e.g. CAN bus, cannot be deterministic, because minor variations in the timing of system functions will cause changes in which messages are arbitrated and transmitted at any given time in a certain communication cycle. So, the message transmission will change and cannot be exactly predictable [20].

In the area of the fieldbus systems, several communication technology solutions have been developed which allow to exchange information or data in real-time in a reliable way. In [21] an overview of these technologies and their evolution over the years is given by Sauter. Also, Sauter emphasized that reliability of data communication networks in the context of achieving real-time performance can be given by solving the dilemma of concurrent access on the communication channels. Therefore, appropriate Medium Access Control (MAC) mechanisms are the key factor in network engineering to ensure deterministic and real-time data communication. MAC protocols can be classified in four categories whereby only three of them can ensure a reliable data communication, as shown in Figure 2.3.

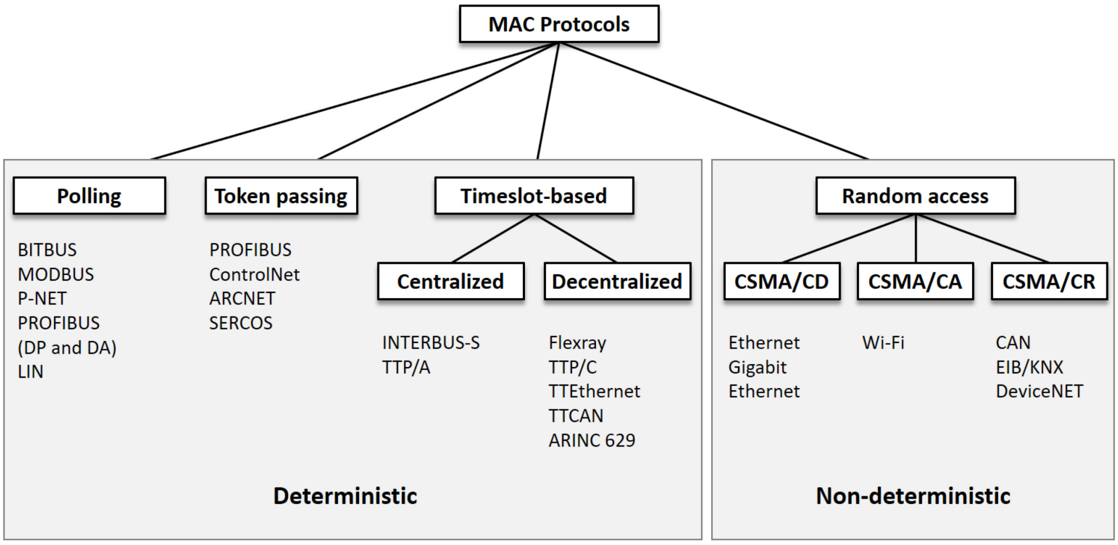


Figure 2.3: Categories of MAC protocols and their application in bus standards [21]

In the figure above, the **random access** approach uses a carrier sense mechanism for collision detection (CD) or collision avoidance (CA) or collision resolution (CR). Although this approach is broadly used in many well-known data and communication standards, e.g. Ethernet (CSMA/CD), CAN (CSMA/CR), Wi-Fi (CSMA/CA), etc., a data exchange in a network using this MAC protocol is typically event-triggered, not predictable and not deterministic.

Polling is typical master-slave approach for MAC protocol. In polling-based protocols a master node polls slave nodes by using a polling message and allows them to send data packets according to a certain schedule. The access to the medium is controlled pervasively by the master all the time. Thus, real-time and deterministic communication can be guaranteed by using this approach.

Token passing is also a deterministic medium access protocol. Similar to polling, only the node who actually has the *token* is authorized to communicate on the channel. The difference to polling is that no predefined master is needed. After the data was transferred, the token is passed to the neighbouring node. This approach requires a logical ring topology since the *token* signal rotates in a ring of nodes. With a priori knowledge of the maximum length of the data packets, the waiting time, propagation time and other time delays can be determined by worst-case-assumption [22].

Time-slot-based MAC protocols are also known as Time Division Multiple Access (TDMA) based protocols. Herein, each node of the network is assigned to a predefined time slot in which it is allowed to send a defined amount of data. This approach requires a precise time synchronisation in the network. The reliability of the network using these protocols is obviously given since the medium access is controlled exactly by a predefined time schedule and thus, data communication in the network is deterministic and real-time capable.

2.2 Modular spacecraft projects

The idea of OOS capabilities of satellites has its origins in the eighties. The American space agency National Aeronautics and Space Administration (NASA) has already demonstrated its capability in satellite servicing since the first servicing mission with astronauts on the space shuttle [23]. The successful series of repairs of the Hubble Space Telescope and the On-Orbit Assembly of the International Space Station denoted milestones of this development. Especially in the last two decades, several technological achievements in the space and spaceflight technologies have driven the space engineering community to take on the new challenges of developing modular, reusable, serviceable spacecraft and to carry the consciousness of the need to reduce, reuse and recycle from earth to space. The mindset of every space engineers shall not be focused anymore to the simply launching of complex spacecraft and systems, but to develop the next generations of sustainable spacecraft designs providing the abilities to be scalable, extendable, maintainable and serviceable on orbit.

There are some known programs of modular spacecraft. The most successful ones are surely the International Space Station (ISS) and the Hubble Space Telescope (HST). Further, there are also some well-known research and development projects like the Multimission Modular Spacecraft (MMS), the German iBOSS project, the Reconfigurable Operational spacecraft for Science and Exploration (ROSE), the DARPA's PHOENIX program with three successfully launched modular satellites and the ARAMIS project (Italian acronym standing for modular architecture for small satellites).

Over the years, modularity in spacecraft has been categorized in different levels as shown in Figure 2.4. Due to the categorization illustrated in that figure, classical monolithic satellites use the typical spacecraft bus where integration of the components into the spacecraft is individual, using custom interfaces and optimized to harness, mass, space and power consumption requirements. The next level of modularity is represented by the minimally modular bus, where the spacecraft are composed of few large modules with the focus to reduce the assembly, integration and test (AIT) efforts and to allow a parallel schedule path [24]. Spacecraft with serviceable modular bus concept like the HST or the ISS provide already some standardized interfaces with the target of serviceability in operation.

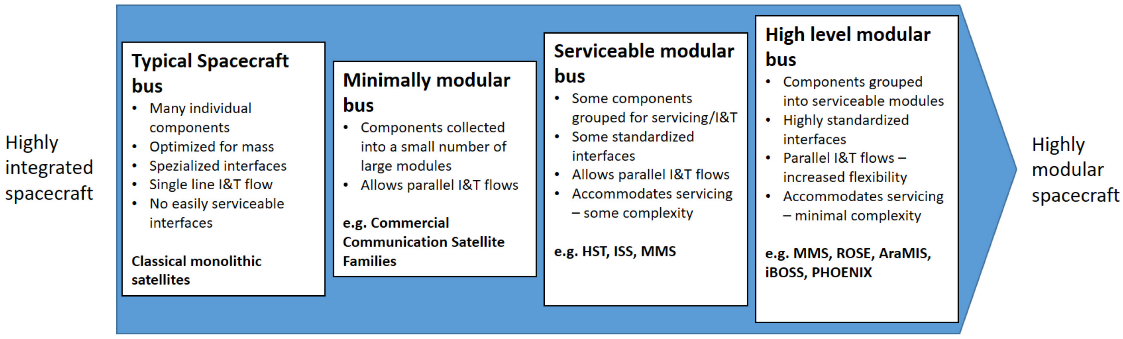


Figure 2.4: Modularity Spectrum

On the next modularity level, highly modular spacecraft provide standardized interfaces and standardized modules with integrated components inside. On this level, single module can be easily removed or exchanged during the AIT phase and in OOS scenarios. Representatives of this level are the ROSE project, the DARPA's PHOENIX program or the German iBOSS project [23, 24]. Within this thesis, the terms *module* and *building block* are used synonymously in the context of modular spacecraft.

2.2.1 MMS

One of the first pioneers in the history of modular serviceable spacecraft was the NASA's Multi-mission Modular Spacecraft design. The MMS design was conceived in the seventies and has been developed and implemented in several mission until the nineties. Since its inception, the design has been applied on six spacecraft [25]. The modularity of the spacecraft in the MMS design incorporates the grouping of components in three major subsystem modules, the Power Module, the Attitude Control Module and the Command and Data Handling Module. The MMS design also provides an optional module for propulsion and offers theoretically a refuelling possibility by fuel tank replacement, what, however, has never been demonstrated in orbit. These modules are attached to the central module support structure as shown in Figure 2.5.

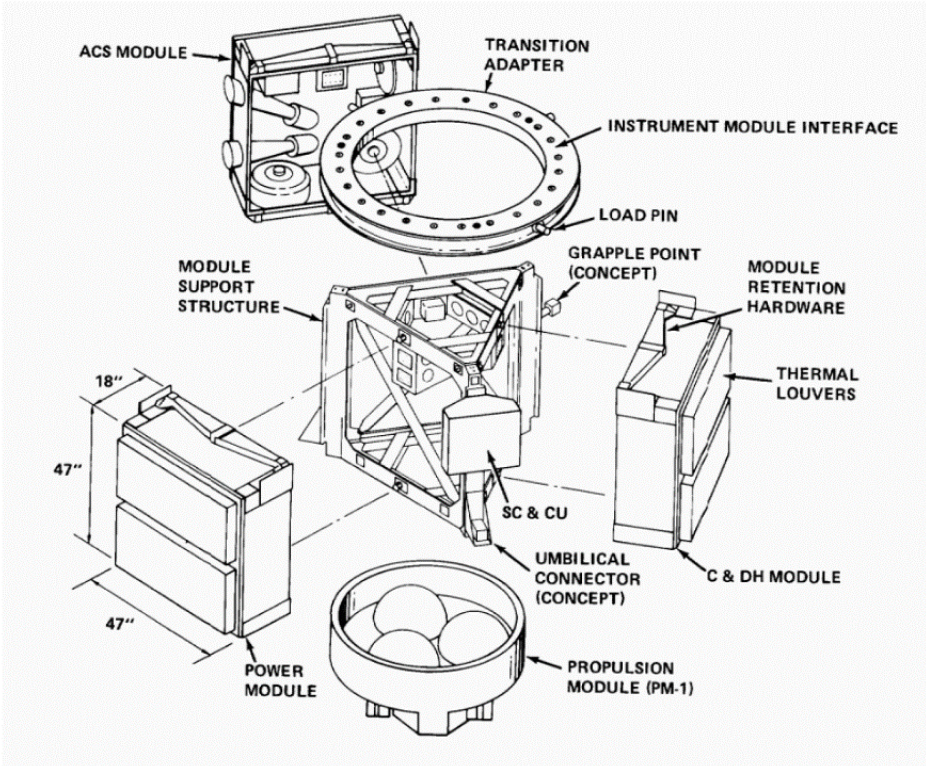


Figure 2.5: MMS component structure [26]

The modular architecture of the MMS design provides a high flexibility. All modules could be developed and tested separately and independently. The reduction of time and cost factors during the AIT phase is a big advantage by comparison to classical spacecraft bus design beside of the serviceability on orbit. The serviceability of the MMS design was successfully demonstrated on the Solar Maximum Mission (SMM) in 1984, when the SMM with a failure in the Attitude Control Module was captured by the Challenger Space Shuttle and the failed parts were replaced by an astronaut [27].

After all, the control architecture of the MMS design is still centralized. Therefore, there was no need for a flexible reconfigurable data bus architecture. However, no information about the applied data bus on the MMS can be gathered. I assume, on the earlier missions they used some proprietary data bus implementation. Maybe on the latter missions of MMS in the nineties, the standard MIL-STD-1553 was used.

2.2.2 ROSE

The Reconfigurable Operational spacecraft for Science and Exploration (ROSE) is a concept for a modular spacecraft developed by NASA/Goddard Space Flight Center on the American Institute of Aeronautics and Astronautics (AIAA) Space Conference and Exposition in 2015. ROSE is built based on the experience from MMS and is introduced as a low-cost implementation of modularity on spacecraft with the objective of long-term affordability. The ROSE concept comprises standardized modules and standardized module interfaces with the aim to minimize the non-recurring engineering costs for a spacecraft as well as the reduction of time and costs for the AIT process.

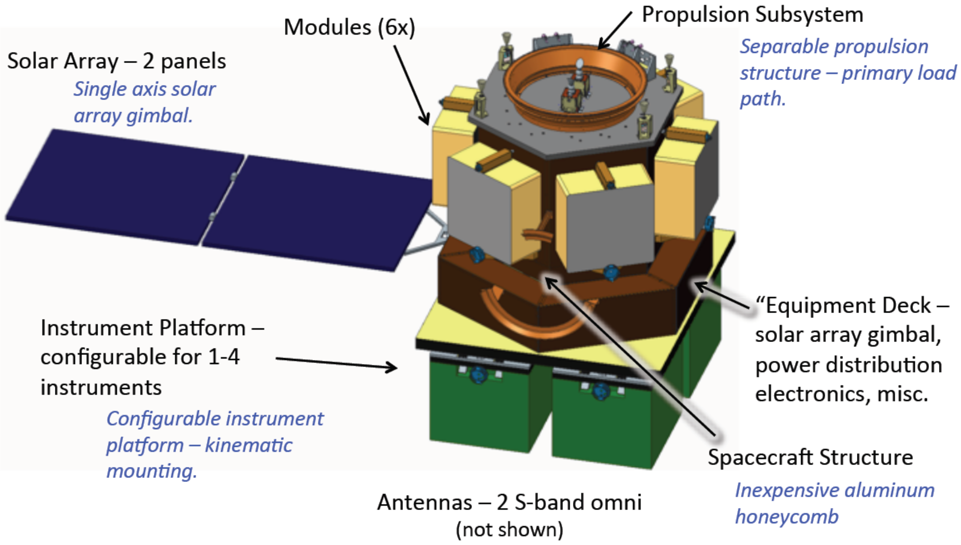


Figure 2.6: ROSE Spacecraft overview [24]

Due to the heritage of MMS, the ROSE spacecraft architecture consists of three main subsystems, the propulsion subsystem, the central spacecraft structure with the modules and the instrument platform, which can also be used as a payload platform. Six modules are accommodated in the central spacecraft structure, two attitude control system (ACS) modules, a reaction wheel assembly (RWA) module, a C&DH module and a power module. The key component of ROSE's distributed avionics approach is the Module Interface Unit (MIU). There is one MIU in each module with a standardized electrical and data interface connected to the rest of the spacecraft [24].

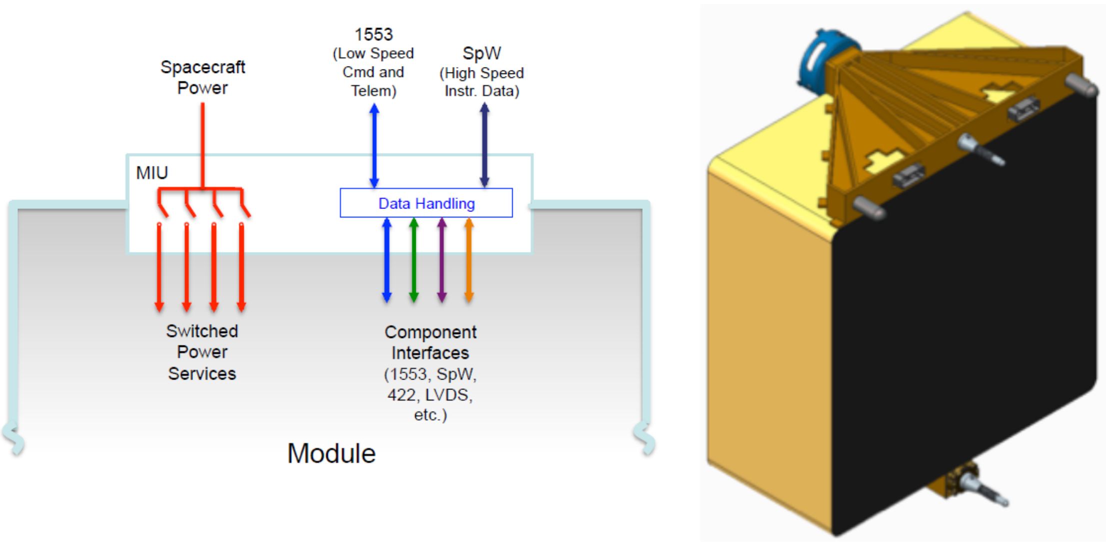


Figure 2.7: ROSE Module Unit Interface [24]

The on-board data communication system of ROSE consists of two data busses, MIL-STD-1553 and SpaceWire (SpW). For critical on-board data traffic like commands and spacecraft telemetry, the slow but high reliable MIL-STD-1553 serial data bus is used. SpW is reserved for the communication of instrumental/payload data because of its high data rate. Figure 2.7 illustrates the backside of a module, where two data connection interfaces are visible.

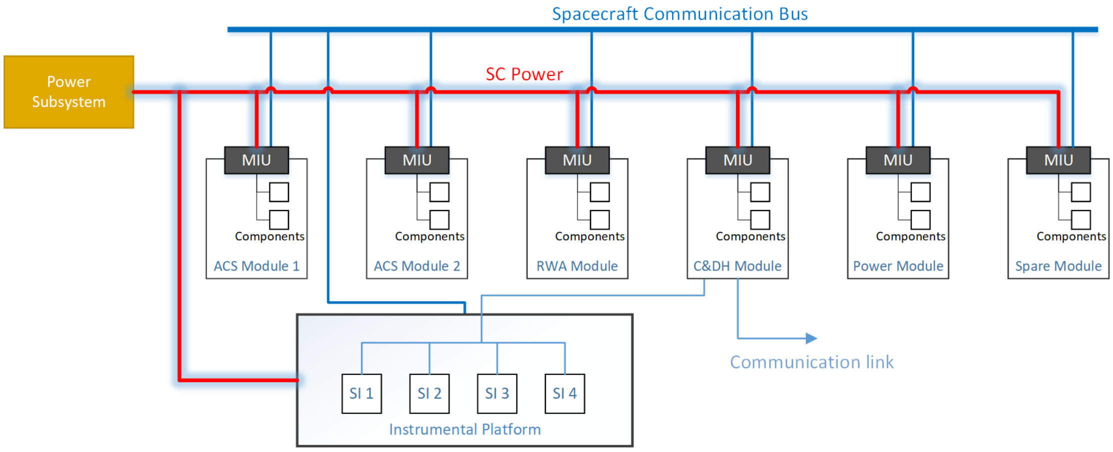


Figure 2.8: ROSE avionics architecture – including two ACS modules, a RWA module, a C&DH module and a power module [24]

In summary, ROSE can be seen as a further development of the MMS concept with the focus on affordability. Although the concept provides high modularity and a distributed avionics concept with the modules, the modular structure is still predefined by the central spacecraft structure and therefore, the variety of configurations of the modular spacecraft is far more limited comparing to other concepts like iBOSS or PHOENIX's HISats. As shown in Figure 2.8, there is a global housekeeping data bus connecting to all the modules. The command bus is physically separated from the data link between the instrumental platform and the C&DH module. Due to the fact, that the payload and spacecraft components are physically separated, this simple approach is sufficient in this case.

2.2.3 ARAMIS

AraMIS (acronym for Modular Architecture for Satellites in Italian) is an academic project of the polytechnic university of Turin, Italia. The main rationales behind the modular architecture design for nano- and microsatellites proposed by AraMIS are reusability and modularity to effectively minimizing the design and non-recurrent fabrication costs. The sustainable usage of the standardized modules in different space missions allows to share design, qualification and testing costs and to shorten effectively the AIT phase [28, 29].

The modularity of the AraMIS architecture is realized by modular intelligent tiles, which are connected. They form an outer satellite structure bearing the user-defined payload in the centre. A modular 2x2x2 satellite structure of AraMIS is shown in Figure 2.9. The tiles have the standardized dimensions 15cmx15cmx15cm and are distinguished in *outer tiles* and *inner tiles*.

The *outer tiles* are of two types:

- The power management tiles are composed of solar panels, a rechargeable battery, a battery charger and a microcontroller-based control module;

- The telecommunication tiles, which can be identified by the deployed antennas on their outer side, are used to receive commands and control packets from the Ground Station and to send back telemetry and status information.

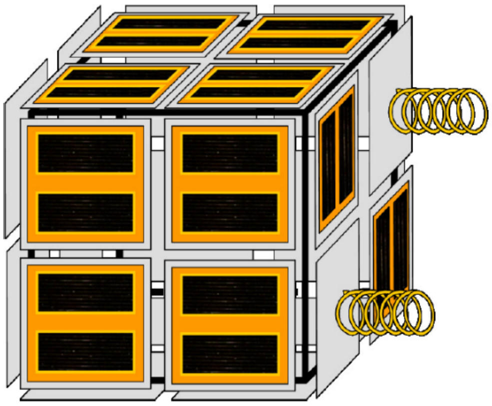


Figure 2.9: AraMIS 2x2x2 satellite concept [28]

Up to date, only one type of inner tile is developed and published in the project. It incorporates the on-board processor and the payload support and shall take care of the on-board data handling and provide the interfaces to the user-defined mission payload.

The AraMIS approach proposes the usage of a global data bus for the on-board data communication. Compliant to the design rationale of the project, reliability at low-cost is the key factor. One of the primary requirements to the data communication system in AraMIS is that all modules of the AraMIS satellite shall be able to communicate with the others for interchanging information (e.g. sensors acquisition), for commanding attitude control actuators and for transmitting the telemetry data to the telecommunication modules and receiving telecommands from these. For this purpose, a proprietary data bus system named as Cross-Bus is developed and verified in AraMIS. Cross-Bus is an ad hoc, dual-redundant and fault-tolerant interconnection system integrated in every tile. It provides galvanic isolation between the tiles by using a nearfield wireless communication based on inductive data transfer [28, 29]. Galvanic isolated circuits and non-conductive data links between the tiles prevent the fault propagation on electrical level; a shortcut on one tile shall not have an impact on the functionality of the adjacent tiles.

Figure 2.10 illustrates the electrical concept of the physical layer of Cross-Bus on one module and the interconnection of a network with six tiles connected to form a simple cubic shaped small satellite. Due to the redundant data paths shown in the figure, every module is connected to all four neighbouring tiles and can exchange data packets with all neighbours.

The redundant communication concept using the Cross-Bus allows reliability and fault tolerance of the network. The data rate is 200 kbps and the data bus is only used for housekeeping and satellite control data traffic. The on-board communication data bus uses a multi-master communication scheme with the same priority in bus access for all nodes and provides a collision detection and avoidance mechanism on the data-link layer. If a bus collision occurs, all nodes involved wait for a random period before starting to send their packets to the bus.

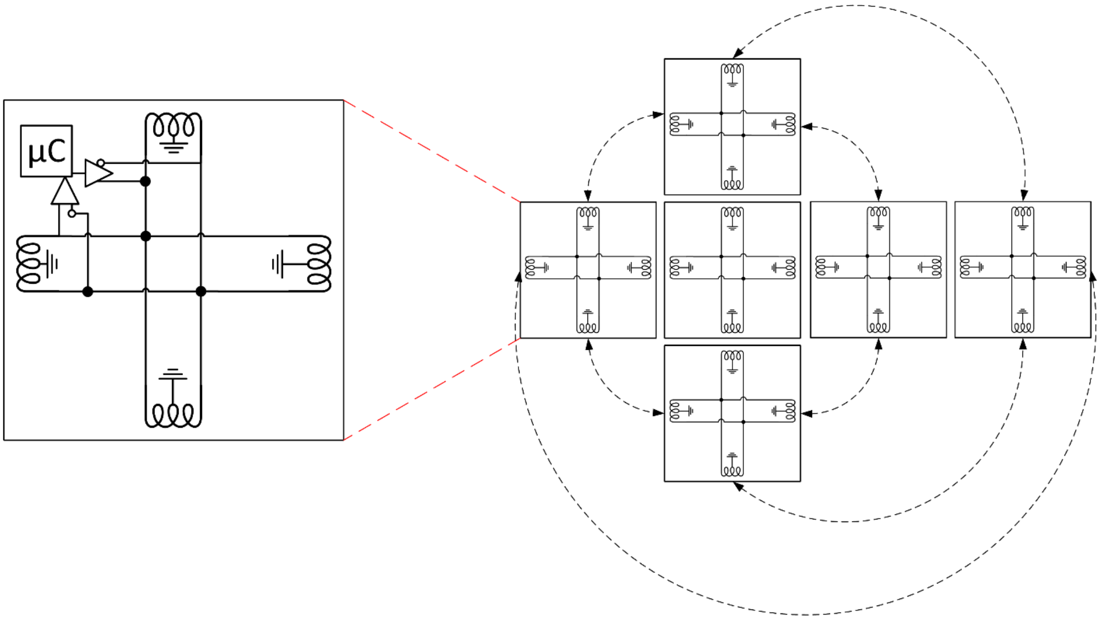


Figure 2.10: AraMIS on-board data bus concept [28]

An approach for the usage of optical wireless transceivers for the data communication network of AraMIS was presented in [30]. Herein, electrical design concepts for discrete infrared transceivers using COTS components were proposed. [31] and [32] presented the an wireless communication architecture using free-space communication or glass fiber in the intra-tiles communication.

From the author's point of view, although the proposed approach for the on-board data communication system of AraMIS provides many properties, which are needed for a modular satellite like robustness, fault tolerance and high redundancy, its applicability is only limited to small (nano- or micro-) satellites. The global data bus with a data rate of around 200 kbps, which is only sufficient for housekeeping and satellite control data traffic. If payload data shall be transmitted over the same media, the data rate allows the integration of only few payload systems with very modest demand for data rate and bandwidth. Since collisions in the bus access shall be detected and avoided by random waiting periods of the sending nodes, the communication network is not deterministic and not real-time capable. Further, due to [28] the problem of babbling-idiot failure cannot be managed by Cross-Bus despite of all the presented mechanisms and fault tolerance.

2.2.4 PHOENIX

In the DARPA's Phoenix program, a modular architecture concept for satellite systems based on the precept of cellularization was studied and developed. Its main objective is to improve the mass-cost-performance equation for space systems by using standardized and qualified modules [33–36]. The PHOENIX program is defined by three principal technical pillars [37]:

- Developing and demonstrating various robotic capabilities necessary for assembling and manipulating satellite parts on orbit (OOS)
- Development of a *Payload Orbital Delivery System* (POD) to increase the tempo of space access
- Development and demonstrating an innovative *Satlet* morphology for modular satellite systems

The term *Satlet* above defines a single cellularized subsystem or a single standalone cell-based module. There are two types of Satlets defined [35, 38–40]:

1. **Single Function Module** – A Single Function Satlet encompasses all components that are needed to perform one subsystem function and the function performance can be accumulated by the assembly of several units (e.g. multiple spatially distributed Reaction Wheel Assemblies to create a total momentum). Modular spacecraft consist of single function Satlets are referred to as heterogeneous modular system since the modules are of different types.
2. **System Module** – A System Satlet represents a standalone system with all essential components such as OBC, EPS (solar cells, batteries), attitude control components (ACS sensors and actuators), etc. The total system performance can be increased serially by aggregation of several System Satlets. Therefore, a System Satlet is comparable with a small satellite itself. Modular spacecraft comprise System Satlets are referred to as homogeneous since all modules are identical.

In 2012 and 2013, the company NovaWurks, located in Los Alamitos, California, won a contract with DARPA to develop and to build the Satlets for the Phoenix program with a total contract value of $42.6 million. The NovaWurks solution is called HISat (Hyper-Integrated Satlets) and follows the homogeneous approach [41]. That means every HISat is a System Satlet containing the required components to provide complete satellite functionality in nanosat-scale package as shown in Figure 2.11.

The dimensions of a HISat are 20cm x 20cm x 10cm and the weight of a module is about 7kg. All HISats provide mounting points and power and data interfaces to allow them to be connected mechanically to others and exchange power and data. An aggregation of several HISats is called a Package of Aggregated Cells (PAC) [41].

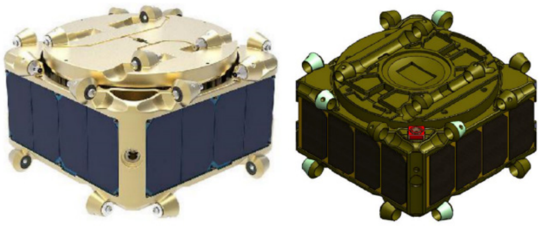


Figure 2.11: HISat – Hyper-Intergrated Satlet [38, 41]

Furthermore, HISats provide an app-based, open-source approach to core resource sharing cellular firmware so that the Satlet resource exposure and sharing is transparent to the operation of the system [38]. Figure 2.12 shows the artist's depiction of NovaWurks' visions of modular satellites built from aggregated HISats.

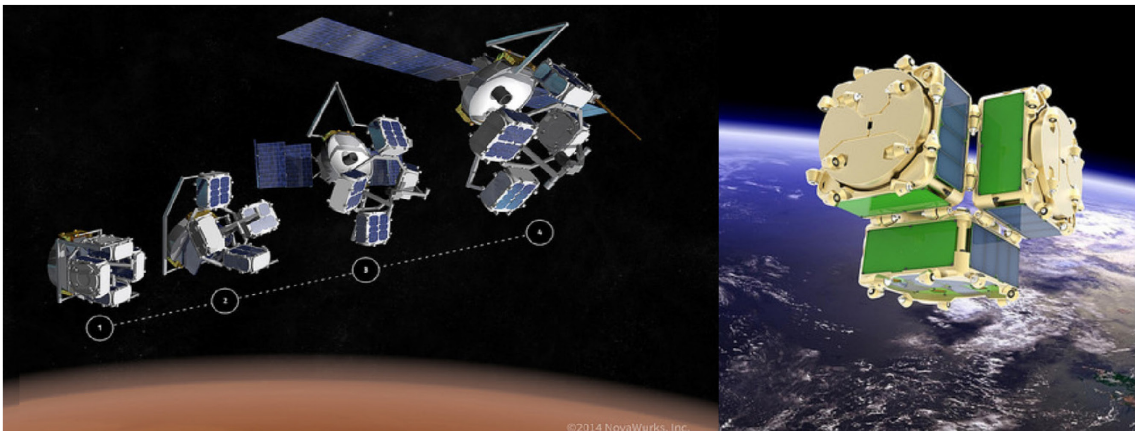


Figure 2.12: Novawurks' vision of modular satellites built up by HISats [41, 42]

Actually, there are three known satellite missions with HISats launched to the orbit [38, 39, 41, 43]:

- **SIMPL** – Satlet Initial Mission Proofs and Lessons is a technology demonstration mission in LEO launched on December 6th, 2015 with the Orbital ATK's Cygnus spacecraft to the ISS. The satellite composes of a six-HISats PAC and two deployable solar arrays as shown in Figure 2.13a. SIMPL was assembled by the astronauts on board of the ISS and deployed by the NanoRacks Microsat Deployer. There is no information available about the mission status after the deployment of the satellite. Nevertheless, the mission goal to demonstrate the ability of a modular satellite launched unassembled and be assembled in orbit was achieved and thus, the mission can be regarded as successful.

- **PODSat 1–** The Payload Orbital Delivery System Satellite is the second satellite mission to demonstrate the Satlets technology. The PODSat is a four-HISat PAC integrated with a Payload Orbital Delivery (POD) system and was launched on the March 6th, 2018 in a subsynchronous geostationary transfer orbit. PODSat should demonstrate the ability of the Satlets technology to incorporate a structural element, the POD chassis, into a PAC. Similar to SIMPL, there is no public information available about the mission status. A depiction of PODSat is shown in Figure 2.13c.

- **eXCITe** – eXperiment for Cellular Integration Technology is a 14-HISat PAC launched on a SpaceX Falcon 9 rocket on the December 3rd, 2018 into a 450km sun synchronous orbit. The mission objectives were to demonstrate the aggregation ability to withstand launch environment, to perform and to maintain thermal control, to communicate with the ground, to reconstitute traditional spacecraft bus capability and to demonstrate the aggregation ability to support a simple and complex payload. According to [44], eXCITe had some communication issues after deployment and was not able to communicate to some of the devices aboard. Further, the spacecraft had technical difficulties to connect to some payload systems. Figure 2.13b shows an artist's illustration of eXCITe.

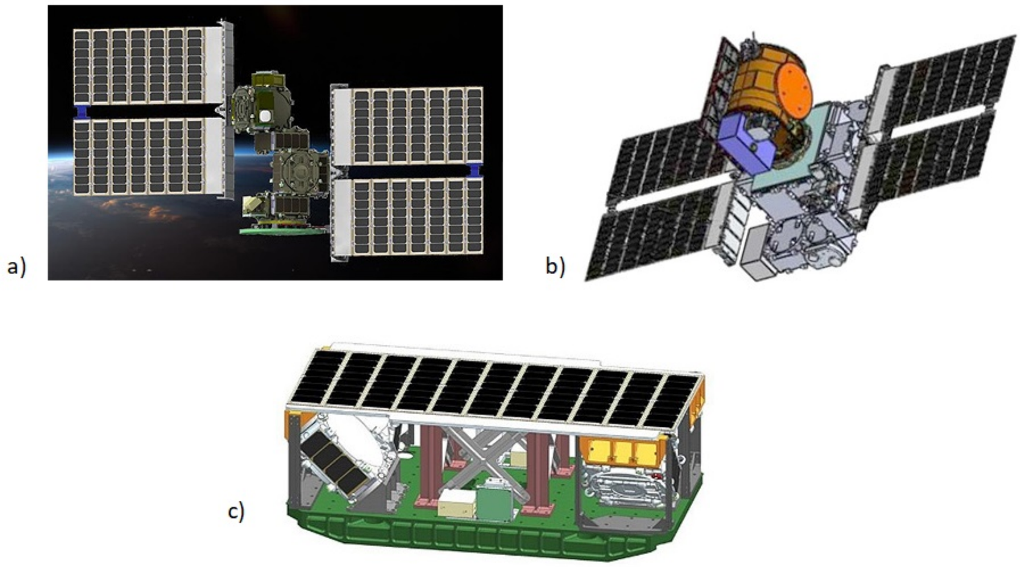


Figure 2.13: Launched satellites based on the HISats [45–47]: a) SIMPL (launched in 2015), b) eXCITe (launched in 2018), c) PODSAT 1 (launched in 2018)

Due to the DARPA publication policy, no information about the technical details of the project has been published. In [40], some very modest information are given about the ability of the ideally robotic performing of the connection of the Satlets in orbit and the possible transfer of electrical power, data and fluid over the connection. Data transfer capabilities between Satlets through the connectors would allow the communication of status, telemetry, commands, sensor data, etc. and imply a data distribution bus integrated in the Satlets. Even a wireless data communication between Satlets was mentioned as an option.

However, the problem of the on-board data communication on the eXCITe satellite mentioned above attested again the importance of a reliable data communication system particularly aboard a modular spacecraft.

2.2.5 IBOSS – A modular satellite design for OOS

In 2011, the German Space Agency started a research project called iBOSS (intelligent Building Blocks for on-Orbit Satellite Servicing and Assembly). The research consortium of iBOSS, led by the Technische Universität Berlin, has worked in iBOSS and two following projects iBOSS-2 and iBOSS-3 from 2011 till 2018 on the concept of a highly modularized satellite design [48–50]. Since the data communication system for modular satellites described in this work is developed and verified in the iBOSS project, the modular satellite concept in iBOSS will be presented in following in more details.

2.2.5.1 The main idea of iBOSS

The basic requirement for the conceptual design of a reconfigurable satellite is the definition of the level of modularity. An analysis of the decomposition levels of a standard satellite bus system is presented in [50] where several modularization levels (system level, subsystem level, component level, etc.) are considered and compared with each other.

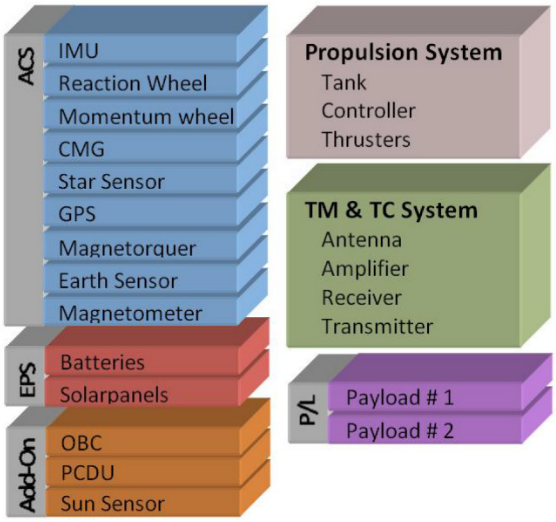


Figure 2.14: Classification of components to subsystems for the modular design in iBOSS [50]

As a result of this analysis, the fragmentation of the bus system on the component level provides most advantages due to scalability, flexibility and the potential of standardization of the building blocks. Therefore, the conceptual design of a modular satellite in iBOSS based on the modularization on component level.

Figure 2.14 illustrates the classification of the standard components of satellite bus systems according to their subsystems. Combination of several components into one building block is possible. On the other side, different components of one subsystem can be located in several building blocks. This concept provides a high level of flexibility but also means the shifting the complexity to the on-board data communication system, because components of the same subsystem is locally distributed in several building blocks. Monitoring, operating and coordinating of spatially distributed components of the same subsystems result in high requirements to the on-board data communication system of the spacecraft and increase it complexity essentially.

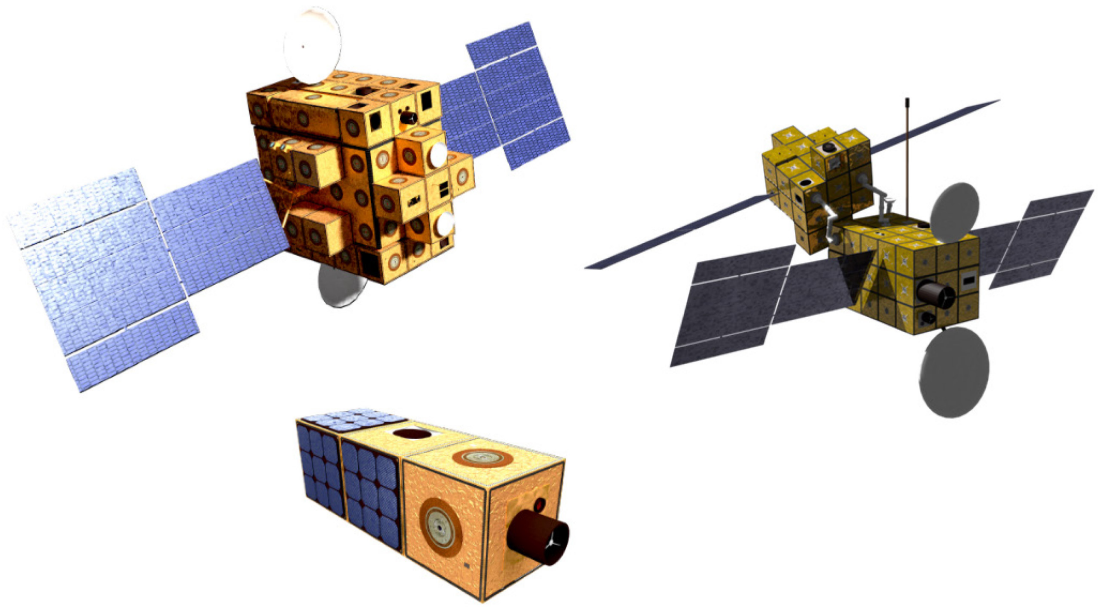


Figure 2.15: iBOSS modular spacecraft concepts and cube-shaped building blocks [10]

Figure 2.14 illustrates the modular spacecraft design concept of iBOSS and the artist's depiction of OOS of a modular client satellite by a robotic servicer with the same modular architecture. One of the key features of the highly modularized design is the capability for OOS which opens new opportunities to a sustainable satellite development and operation. The satellites can be robotically reconfigured, maintained, upgraded with new hardware components to enhance the system performance or even refueled to extend the lifetime on orbit. Beside to that, the standardization of the building blocks and the intermodular interfaces implicates time and cost saving due to the high flexibility in satellite development, production and qualifying processes as well as rapid-production-on-demand capabilities.

The iBOSS concept did not only focus on hardware development aspects but also on the development of dedicated software such as the operating system and basic software stack for the on-board computer, computer-aided satellite design, reconfiguration planning and detailed spacecraft simulations which also helps to reduce development time and costs [48].

2.2.5.2 iBLOCK - the building block of iBOSS

In the iBOSS concept, the satellite bus is decomposed to several building blocks with standardized form factor but individual functionality, the so-called iBLOCKs. The iBLOCKs have the standardized dimensions of 40cm x 40cm x 40cm. However, there are also the possibility to design double or triple iBLOCKs whose dimensions are a multiple of the dimensions of a standard iBLOCK.

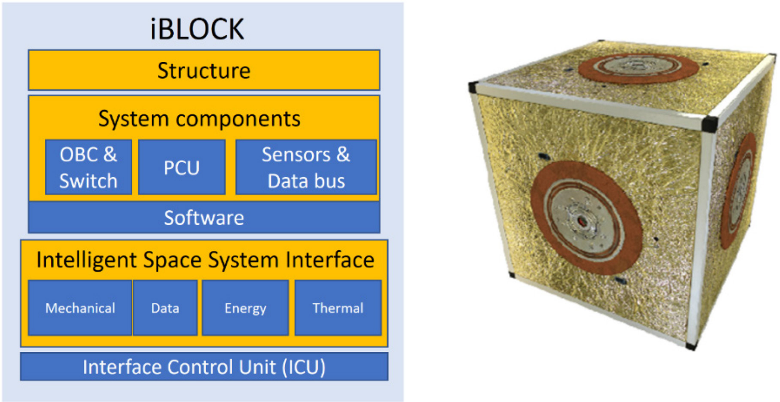


Figure 2.16: Standard components of an iBLOCK [49]

Every iBLOCK provides an On-board Computer (OBC), a Power Control and Distribution Unit (PCDU), up to six intelligent Space System interfaces (iSSI) and the same amount of Interface Control Units (ICU) as the standard interior equipment. The On-Board Computer bases on an ARM architecture and is powerful enough to run a fully-fledged Linux system with 3rd-party software required to build a communication system between the building blocks. The OBC also has a CAN interface to connect to local devices within a single building block [7–9, 13, 48].

Inside of the iBLOCKs, subsystems or payload components e.g. reaction wheels, inertial measurement units, attitude sensors, power control units, RF transceivers, mission payload, etc. are connected to the OBC via either the local CAN bus or via the global data bus based on the Ethernet standard. All components have to be integrated into the global software stack along with important metadata such as their position within the compound [48].

Figure 2.17: Software stack of iBOSS [48]

Figure 2.17 shows the software stack of the proposed operation system with several layers of third party software. On top of this, a combination with ROS as a middleware layer is used and a software infrastructure for decentralized computing is implemented including task management, update management and basic topology identification of the modular satellite.

2.2.5.3 iSSI - The multifunctional interface

One of the key component of every modular system is the standardized interfaces connecting the modules. In iBOSS project, the iBLOCKs are connected by a multifunctional interface with mechanical, electrical, thermal and data interconnection that are called iSSI – intelligent Space System Interface. An iSSI is allocated in the center of each side of the iBLOCK and comprises four individual interfaces, as shown in Figure 2.18.

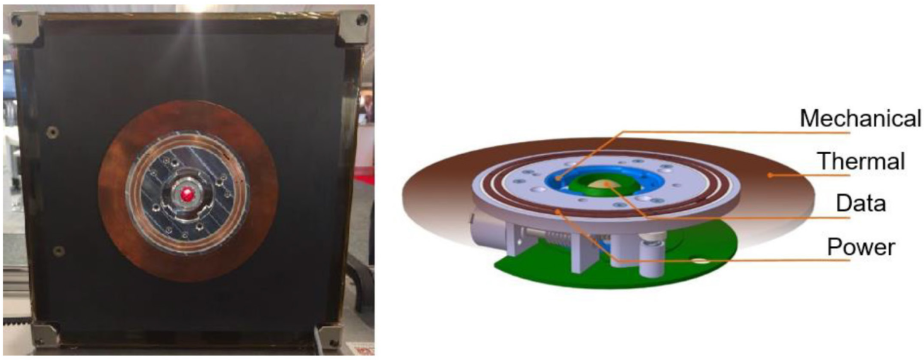


Figure 2.18: The multifunctional interface iSSI [7, 8]

Hence, the iSSI is a highly integrated multifunctional system unifying the functionalities of mechanical coupling, power, heat and data transfer. Hereby, the data transmission is realized by an optical communication system accommodated in the centre of the circular interface. The main parts of the mechanical interface including the coupling and guiding elements as well as the positioning pins are arranged around the data interface. The next ring layer consists of two electrically isolated, spring-supported copper rings, which allow the transmission of electrical power between building blocks, and represents the electrical power interface. The thermal interface is allocated as a ring construct surrounding the mechanical interface and consists of a carbon-nanotube copper-alloy composite material. This composite features a high conductive heat transfer coefficient, which allows a significant heat exchange between the modules even with relatively low contact pressure. Each iSSI is controlled and monitored by an Interface Control Unit which consists of sensors and actuators to provide the interface functionality [51].

2.3 Data communication systems in modular robots

Modular robots show many similarities and analogies to modular spacecraft regarding to the morphology and the architecture of both systems. Modular robots are a family of robotic systems built of interconnected module coupled by docking interfaces. These robots are composed of simple and self-contained building blocks which usually are alone-standing systems with own sensing, actuating and computing components.

Regarding to the ability to change the morphology of the system due to the modular structure, modular robots are referred to as reconfigurable or self-reconfigurable depending upon the autonomy level of the system [52]. Research contributions to the interdisciplinary field of modular and reconfigurable robots started in the late eighties with CEBOT (Cellular Robotic System). Since then, the research field of modular robotics has grown fast and widely. Many modular robots have been developed and the concepts have evolved over the years. In general, data communication is the key factor in all modular systems. Ones distinguishes here between intramodular communication and intermodular communication. The most used communication protocol for the intramodular communication on modular robots is I^2C beside of some other serial data bus standard like RS-232, RS-485 and SPI.

Intermodular communication on modular robots are typically realised by optical communication and wired CAN bus and I^2C. Some systems have started using wireless communications like WiFi, Bluetooth, ZigBee, etc. in the past few years. Although optical communication using infrared (IR) provides no high data rate, it is still preferred by many systems due to the galvanic isolation of the data link between modules. IR is used by Polybot, CONRO, Crystalline, Telecube, Atron and many others. Fable II uses a proprietary 2.4 Ghz wireless communication system. ZigBee is used by Transmode, Mobot and Linkbot and SMORES. Bluetooth is used by M-TRAN III and M^3 Express. Ethernet communication is used by CoSMO because of its high data rate. Table 2.3 summarizes the communication systems used in modular robots [53].

Table 2.3: Summary of intramodular and intermodular communication in modular robots [53]

Modular Robotic system	Intramodular communication	Intermodular communication	Protocol
Polybot (v1, v2, v3)	-	CAN, IR (v2, v3)	RS232(v1)
Crystalline		UR	Serial
CONRO	-	IR	Serial
M-TRAN	-	CAN, Bluetooth	CAN, Bluetooth
Telecube	-	IR, M2M	
ATRON	I^2C, Rs-485	IR	RS-485
Microtub	-	I^2C, M2M	USB
Superbot	I^2C	SPI, IR	SPI
CHOBIE	-	IR, M2M	
Molecube	RS232	I-wire	USB, wireless, Bluetooth
iMOBOT	-	IR	
Ubot	SPI	Serial	Wireless
SMORES	-	Wireless	Wireless

Transmote	-	ZigBee	ZigBee
M³ Express		Bluetooth	Bluetooth
CoSMO	I^2C	Ethernet	Ethernet
M-Blocks	I^2C	IR, serial	serial
Mobot, Linkbot	-	ZigBee	ZigBee
Kairo 3	-	CAN, M2M	Wireless, Ethernet
Hinged-Tetro	-	UART	UART
PetRo	-	IR	
Fable II	-	2.4GHz wireless	proprietary

Although modular robots show a lot of similarities and analogies to the highly modularized spacecraft design at first sight, the differences in the application of both systems especially in the operation environments result in different requirements to intermodular communication system. Most of the presented concepts modular robots are developed and constructed for terrestrial applications and their requirements to the data communication system distinguish essentially from the requirements to the data communication system of modular spacecraft especially regarding reliability and fault tolerance. From all the listed modular robotic systems in the table above only Polybot was analysed in [54] for its suitability to perform in space applications as manipulation in space and surface mobility in planetary missions.

2.4 Dependability of the data communication network aboard modular spacecraft

Dependability is defined in systems engineering as a qualitative and quantitative measure of the system. Dependable systems engineering has become a broad research field since the 1960s. Especially in aerospace area, dependability is a key requirement. In fact, one of the most important non-functional requirements to the subsystems on spacecraft is dependability.

Especially for highly modularized spacecraft with OOS capability, where the communication between critical components, which are spatially distributed in different building blocks throughout the spacecraft, forms the basic infrastructure for the cooperation of the building blocks to perform system-critical functions, dependability of the on-board data network becomes a key requirement to the on-board communication system of the modular architecture design. Due to the distribution of components of one subsystem in several building blocks, the critical data needed for performing the function of this subsystem may be transferred over several building blocks. The data path may contain many hops and is more prone to failure than on a monolithic satellite.

Hence, a failure of a component on the data path through the satellite can have critical impact on the functionality of the satellite. For example, ACS sensors and actuators are accommodated on different building blocks on a modular system, sensor data or commands to the actuator need to be transferred through several building block to their destination, and the failure of one switch on the path would have severe impact on the ACS function of the satellite. Compare to it, on monolithic satellites, there is usually only one bus connection between the sensor, actuator and the processor where the control algorithm is implemented. Thus, the data path is simple and robust. In following, the meaning of system dependability will be described and applied to the design of the communication system aboard modular spacecraft.

The collective term *dependability* encompasses the availability, reliability and maintainability of a system. Avižienis and others categorized *dependability* in [55] in three elements - Attributes, Threats and Means.

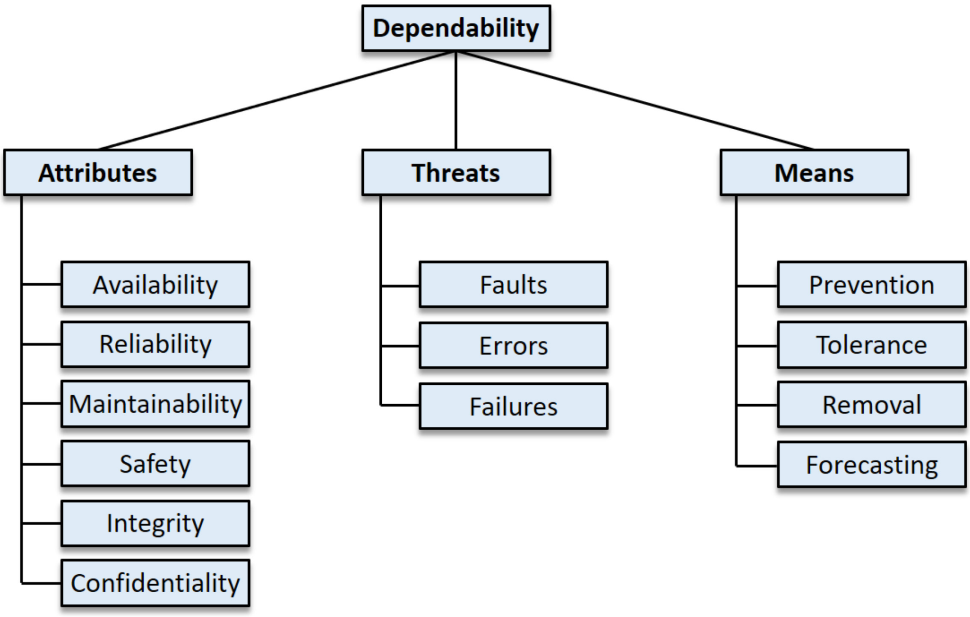


Figure 2.19: Systematic exposition of the dependability concept [55]

Dependability attributes represent qualitatively and quantitatively assessable features of a system. Availability of a system A can be described quantitatively by the ratio of the mean time between failures t_{up} and the mean time to repair the system t_{down} assuming the system is not available during this repairing.

$$A = \frac{t_{up}}{t_{up} + t_{down}}$$

Reliability is the continuity of correct service delivery and can be assessed by the mean time between two system failures t_{up}. Reliability engineering is a sub-discipline of systems engineering and is excessively adapted in spacecraft engineering. Maintainability is the ability of a system for repairs and modifications and can be quantitatively measured by the time required to restore a system after a failure occurrence [18]. Safety, Confidentiality and Integrity are not regarded in this thesis and may be handled in future works.

Threats of dependability are events or undesired states that affect the dependability of the system in negative manner. Three categories of dependability threats are defined [55]:

- **Faults** - A Fault is a defect within the system that can cause errors. A fault is active if it leads to an error, otherwise it is a dormant fault.
- **Errors** - An Error refers to an unintended system state that leads to a failure. Errors are manifestation (symptoms) of a fault.
- **Failures** - A Failure is deviation of system behaviour from the intended service defined by the system specifications. A system failure means that the system cannot perform its intended function.

Dependability means refer to the utilization of techniques or mechanisms to increase or to ensure the system dependability. Following means are identified [55]:

- **Fault prevention** - Faults in a system can be prevented by methods and techniques of quality assurance and control during design, development and manufacturing process and in the AIT phase of the spacecraft. For example, prevention of operational physical faults can be attained by shielding, radiation hardening, etc.
- **Fault tolerance** - A system is fault tolerant if it is still able to provide functionalities to fulfil its specification in acceptable manner in the presence of faults. Fault tolerance encompasses error detecting, error handling and system recovery mechanisms.
- **Fault removal** - Fault removal can be performed during the development or during the operational phase of the system. During development fault removal consists of three steps: Verification, Diagnosis and Correction. In general, physical faults within a spacecraft in orbit are difficult to removal. The modular spacecraft design with OOS capabilities opens the door to new possibilities of fault removal in orbit by exchange the module with the faulty part by a new fault-free one. However, this requires the ability of fault containment in each module of the modular satellite. Fault containment is a system feature, which allows to limit the impact of a failure within the system where this failure occurs.
- **Fault forecasting** - System failures can be forecasted by evaluation and estimation of prospective occurrence of faults.

For the on-board communication system of modular spacecraft, faults in hardware and software need to be prevented already in the design phase. By using available space qualified hardware or by considering space hardened design in the development with subsequent qualification of the hardware for space applications, the occurrence of hardware faults shall be minimized. Some design specifications to harden the network device hardware are presented in Chapter 4.

The highly modularized satellite design enables the exchange of faulty module by OOS; thus, it is important that faults in a module need to be contained and limited within this module. The communication network shall provide the ability to deactivate and reactivate the data link between the modules.

Failures of the data communication on a modular satellite can be classified in two main categories – the failure of the data link between two adjacent building blocks and the failure of a building block. A concept to ensure the fault tolerance of the on-board data communication system for modular satellites will be described in Chapter 5.

2.5 Concluding remarks

The C&DH subsystem on a spacecraft can be compared to the human brain and the on-board data communication system corresponds to the nervous system of the human body. The on-board data communication is responsible for the exchange of data between space components using a data bus to ensure system functionalities. Therefore, its most important design constraint is dependability. In this chapter, the relevant fundamentals of the on-board data communication system of spacecraft are described and discussed.

Following to that, an overview of different approaches for modular satellite designs with different modularization levels was presented. Hereby, the focus was directed to the highly modularized designs and the deployed on-board data communication systems. Only modest information can be gathered here. Especially for the PHOENIX program with three launched modular satellites in orbit, no information about the deployed communication architecture has been published. Only in one publication, the approach of wireless communication was mentioned as an option. According to the information given in [44], eXITe, the only satellite in this program which shall demonstrate full functionalities of a modular satellite on orbit, had communication issues and the on-board C&DH was not able to communicate to some devices aboard. This fact affirms again the importance of dependability of the on-board data communications system. Indeed, highly modularized satellites are more prone to failures of the network hardware than classical satellites. Critical messages may need to be transferred over many hops to its destination on a modular satellite and higher probability of the failure occurrence of the involved network hardware is expected. In Section 2.4, system dependability for on-board the data communication system is discussed from the point of view of a modular satellite system.

Rounding up the chapter, an overview of modular robots was presented and discussed with respect to the deployed intermodular communication systems. Although modular reconfigurable robots have many conceptual similarities with modular spacecraft, most of them have been developed for terrestrial applications and the data communication systems applied on these robot systems cannot be deployed on modular spacecraft for many reasons, but mostly due to the missing of the necessary dependability, determinism and real-time capabilities.

3 DATA COMMUNICATION CONCEPT FOR MODULAR SATELLITES

This chapter begins in Section 3.1 with several fundamental conceptual considerations for develop an appropriate data communication system for highly modularized satellites. Based on the conclusions that can be drawn from the conceptual considerations and the fundamentals in the previous chapter, requirements to the on-board data communication system are derived in Section 3.2. They are followed by decision criteria, which are worked out for a comparison of several communication architectures that have been already used in space applications or have the potential to be used in future space missions in Section 3.3. The result of bus comparison represents a communication standard that can fulfil most of the requirements to the data communication network of modular satellites. However, challenges invoked by the reconfigurability of the modular systems still need to be met. The challenges and constraints of the application of the chosen communication architecture on modular satellites are described in 3.4 and build the basics for the hardware development and network management concepts in subsequent chapters. The chapter is concluded with the concluding remarks in 3.6.

3.1 Conceptual Considerations for the On-board Data Bus System in modular satellites

This section provides the conceptual considerations needed to design and develop a reliable and reconfigurable on-board data bus system for modular satellites. Firstly, physical network topologies presented in Section 2.1.1 are evaluated for their applicability on a modular architecture, following by the discussion about logical communication schemes and the consideration, how the two data traffic types (P/L data traffic and satellite control and housekeeping data traffic) can be handled on modular satellites.

3.1.1 Analysis of physical network topology for modular satellites

While classical monolithic spacecraft usually apply the linear bus, star or tree topology for the on-board data communication network due to the centralized system intelligence, these topologies are not recommended for the data communication network aboard satellites with highly modularized architectures. Highly modular satellites based on the decomposition of the subsystems imply the spatial distribution of spacecraft components in building blocks which need to be monitored, operated and controlled by a local processor in the same building block. Therefore, these satellites can be regarded, more or less, as distributed real-time systems whose distributed system intelligence need to collaborate and the performing of the system functions needs to be coordinated by using message transmission.

Especially, highly modularized satellite designs with modules of same form factors without a central structure like in the PHOENIX program or in the iBOSS project represent a network cluster of on-board computers distributed in their modules or building blocks. Every module can be connected to several neighbouring modules. Depending on the given form factor of the building blocks and the intermodular interfaces, not all of the physical network topologies described in Chapter 2.1.1 can be applied on highly modular satellites. For example, regarding an exemplary 2x2x2 modular satellite with the iBOSS design, the star topology cannot be adopted as the physical topology of the on-board data communication network, since there is no possibility to establish directly physical connections from one central building block to all others using the interface concept of iBOSS. Figure 3.1 shows the exemplary 2x2x2 configuration of a modular satellite using iBOSS building blocks. The red lines between the modules represent the possible direct physical data links between two building blocks.

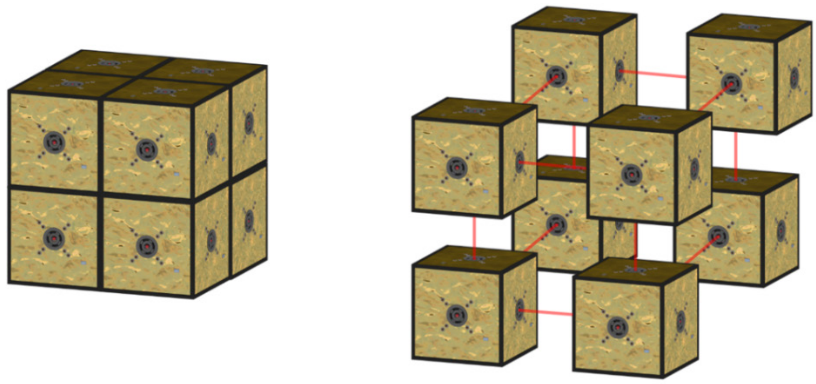


Figure 3.1: Illustration of an iBOSS modular satellite with 2x2x2 configuration

Except for the star topology, all topologies presented in Chapter 2.1.1 can be adopted for the data network of the building block configuration above, as depicted in Figure 3.2.

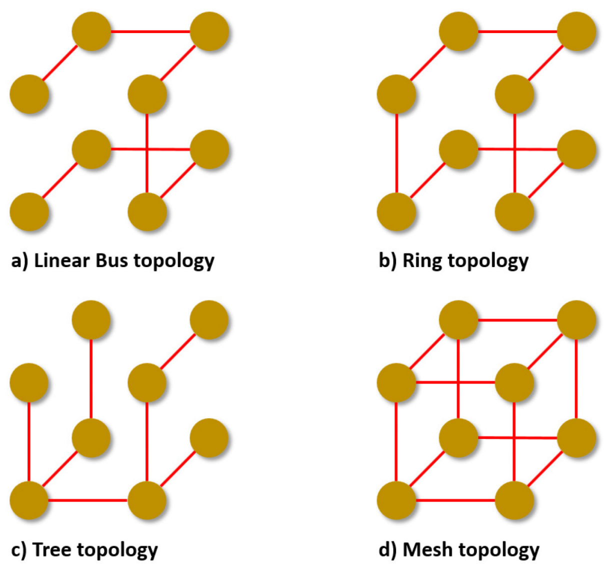


Figure 3.2: Different possible network topologies for the exemplary modular satellite configuration

For modular satellites, the linear bus topology and the tree topology are not recommended since a failure of the connection between two network nodes will inevitably lead to the dismantling of a section of the network from the rest. Depending on the criticality levels of the applications afflicted by the communication disruption, this may lead to a total system failure. Therefore, networks with linear bus topology and tree topology can have many single points of failure (SPOF). Further, a linear bus topology will limit the extensibility of a modular satellite, since modules cannot be added on an arbitrary spot of the satellite since this may change the network topology.

Networks with ring topology is less prone to a link failure since networks with this topology will fall back on a linear bus topology in this case. Comparing to networks with ring topology, networks with mesh topology provide clearly higher communication reliability due to multiple redundant data paths and so, they are resistant to link failures. However, these networks contain one or more network loops, which is the most common cause for broadcast storms. A broadcast storm can consume massively network resources, and, in some cases, it can crash or overload the whole data network. Especially on modular satellites with OOS capability, where modules shall be reconfigured, added or removed without an impact on the network, the mesh topology is the most recommended physical topology, but only if the network loops issue can be handled on higher layer of the OSI model.

It can be concluded that for highly modularized satellites, applying the mesh topology as physical topology for the on-board data communication network can increase the communication reliability due to the redundant data paths existing in these networks. As a trade-off, the data communication system must provide appropriate measurements to handle network loops.

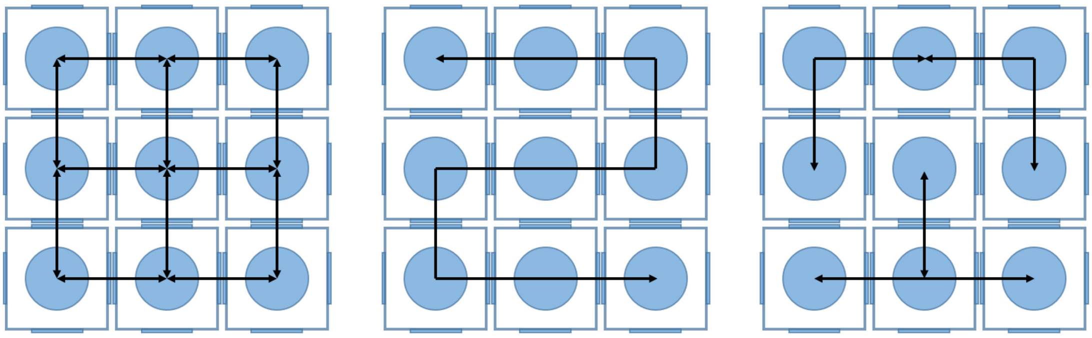
3.1.2 Logical communication schemes for modular satellites

Modular satellites are built of several modules connected with each other. Thus, data communication between the modules is an essential design factor of modular systems. Referring to the results of several research works in the area of distributed robotic systems and modular robots [56, 57], ones can usually distinguish between local communication scheme and global communication scheme. In this section, the local and global communication scheme are presented and their applicability on modular satellites as well as the advantages and disadvantages of the schemes are discussed in following.

Figure 3.3 illustrates the three different kinds of logical communication schemes within a modular system where the modules are represented by the squares and the data communication by arrows.

In a network with local communication scheme, every network node is connected to all of the adjacent nodes and can only communicates with these, as shown in Figure 3.3(a).

In this logical communication scheme, messages from one node to a non-adjacent node usually are sent as broadcast messages to all of its neighbouring nodes who will send the received message again to all neighbours. This will guarantee that a message will definitively arrive at its destination nodes in the network. Depending on the physical network topology, this also implies that one message can arrive several times at the destination nodes with different time via different paths through the network. On the one hand, these redundant transmissions of messages increase the reliability of the network communication. On the other hand, the network efficiency may be reduced by the high bandwidth consumption for every message and the data traffic load of the network grows rapidly with increasing node number of the network. Further, every network with loops like the example network illustrated in Figure 3.3(a) will need a control mechanism on higher OSI levels to filter and drop duplicated message to prevent broadcast storms.



a) Local Buses b) Global Bus c) Hybrid Buses

Figure 3.3: Different logical communication architectures within a modular system

On the other side, the local communication scheme has less or nearly no requirements to the physical layer and to the higher communication protocols like multiple bus access protocols and control mechanisms. Preferentially, simple serial physical busses like I^2C, SPI, RS-485, etc. can be applied here. Hence, the modules must have a dedicated transceiver for every local bus connection to the neighbours. This communication bus scheme provides a high flexibility level and other characteristics needed for a reconfigurable modular satellite like reconfigurability and extensibility with nearly no constraints on the network topology or on the network management. Additionally, this communication scheme provides high redundancy of the transmitted messages in the network.

As trade-offs of high flexibility and reconfigurability, the lack of determinism of this communication scheme due to the needed broadcast communication and the missing bus access control are the limited factors of the application of this scheme on safety critical systems like spacecraft.

A network with global communication scheme, as shown in the Figure 3.3(b) provides more controllability and deterministic data transmission and is more suitable for systems with time critical on-board communication [56]. While the local communication scheme only provides a communication medium to exchange messages between neighbouring network nodes, in a network with the global communication scheme, messages can be exchanged between all nodes. If the network management provides mechanisms to control the bus access of the nodes e.g. TDMA or similar to prevent collisions, real-time data communication between the nodes is possible. On such systems, one can apply well-known bus standards that have been successfully applied on safety critical systems like the MIL-STD-1553 or the time-triggered CAN (TTCAN) with high controllability and deterministic data transmission with high reliability. However, the deterministic data transfer also required static, a priori known network topology. Thus, the needed characteristics of the communication architecture of a modular satellite like reconfigurability or extensibility and flexibility are significantly limited.

In Table 3.1, the advantages and disadvantages of the local and global communication scheme regarding to the requirements of a modular satellite are compared to each other.

Table 3.1: Comparison of local and global data bus scheme

	Local bus scheme	Global bus scheme
Network efficiency	Low	High
Reliability	High	High
Determinism/Real-time communication	No	Yes
Flexibility	High	Low
Reconfigurability	High	Low
Extensibility	High	Low
Requirements to the network management	Low	High
Applicability for safety critical systems	Low	High

Due to the table above, neither the local data bus scheme nor the global data bus scheme is appropriate for the application on a modular satellite. Therefore, this thesis proposes a hybrid communication scheme that combines the advantages of the both other schemes as an applicable approach for modular satellites.

Within a modular system with the hybrid approach, several static non-global communication channels are implemented, allowing the communication between non-adjacent modules. According to the criticality levels of the messages, some channels can be prioritised over others and the communication of time critical messages is therefore deterministic without a limitation of the flexibility of the system. The desired characteristics of a modular system like reconfigurability or extensibility are provided in certain degree even if they are not fully provided like in a system with the local bus scheme.

The only disadvantage of this approach is that the possible options of usable bus standards are strongly limited. There are only few communication architectures that support the implementation of the hybrid communication scheme with the combination of flexibility of the network and the communication of time critical messages.

3.1.3 Separation of P/L data traffic and HK and Satellite control data traffic

In general, the communication bus system of a satellite is divided in two physically separated data bus systems, the payload data bus, which transfers payload data, and the housekeeping data bus, which is used to transport the housekeeping and satellite control data needed to monitor, control and operate the satellite system. The transmission of the mentioned two types of data require different and, in some cases, nearly contrary properties of the data communication bus as described in Table 2.1. How has this problem been handled on typical monolithic satellite?

Typical commercial and civilian satellite systems in earth orbits can be generally classified in the following main mission types:

- Earth observation missions
- Navigation missions
- Telecommunication missions
- Astronomical and Space observation missions

According to the mission objectives, there are different types of payload systems integrated on the satellites. The satellite system is usually separated in the satellite bus and the mission payload, which are developed individually and integrated before testing as shown in Figure 3.4.

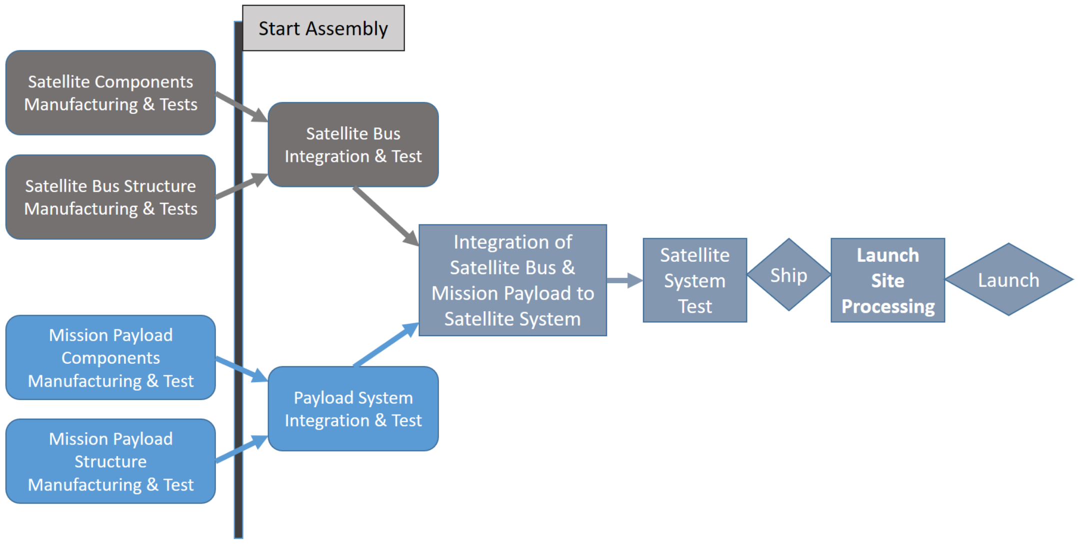


Figure 3.4: Typical satellite system production flow [58]

For achieving the optimum in weight and power as well as in functionality, customized interface design is used for the tight coupling of the satellite bus and the mission payload and the key communication linkage between the satellite bus and the mission payload is provided by the on-board communication data bus [58].

3.1.3.1 On-board data communication on reference satellite missions

For monolithic satellite systems, the approach above has regularly been adapted. As references, two satellite missions with different operating orbits and different mission objectives are discussed regarding to the on-board communication systems in following [49].

EnMAP

The EnMAP mission is a German LEO earth observation mission. The EnMAP spacecraft bus is based on the existing state-of-the-art bus technology LEOBus designed by OHB-System. The payload unit is a hyperspectral imager which can take up to 50 images in two different spectral regions of the earth.

As shown in Figure 3.5, the data traffics aboard EnMAP are carried out by two separated data bus systems. A high rate data bus (Channel Links 540 Mbps) is used for the transmission of large image data between the HSI unit and the payload communication unit (X-band transceiver). The on-board communication data bus for the transmission of satellite control data for operating and controlling the satellite system is the MIL-STD-1553 bus. The physical separation of the two data busses is the more suitable approach for EnMAP satellite, since the mission payload itself has a Payload Support Assembly unit which consists of a payload data handling part (Data Science Handling Assembly) and a own payload communication unit using X-Band for high downlink data rate (320 Mpbs) [49, 59].

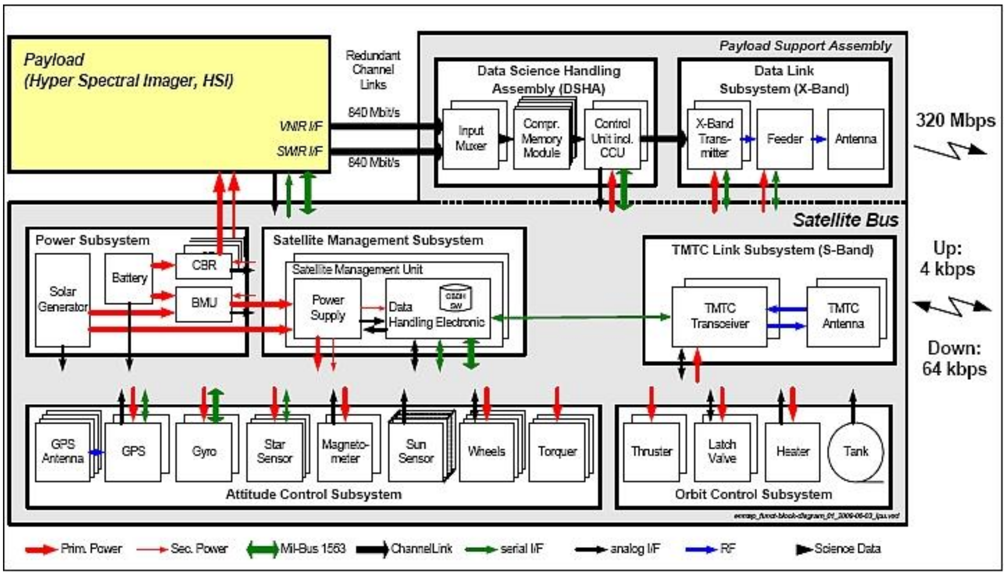


Figure 3.5: Functional block diagram of the EnMAP satellite bus [59]

SmallGeo

Another reference satellite design is represented by SmallGEO, a general-purpose small geostationary satellite platform project funded by the European Space Agency (ESA) and the German Aerospace Center (DLR). The project SmallGEO shall establish a general purpose, small geostationary satellite product developed by a European space company to demonstrate the ability of European space industry to take a significant role on the commercial telecommunication market for small platforms.

In 2008, a contract is signed by ESA and the German space company OHB System AG and Hispasat S.A. of Spain to proceed the first SmallGEO mission, Hispasat AG1, which is launched on January 28th, 2017.

The network architecture aboard the Hispasat AG1, is mainly carried out by the proven MIL-STD-1553, similar to the EnMAP spacecraft. This bus standard is also used for the data communication between SMU (Satellite Management Unit) and the PMU (Payload Management Unit). For the communication between the PMU and the communication payload a high rate optical link is used [49, 60, 61].

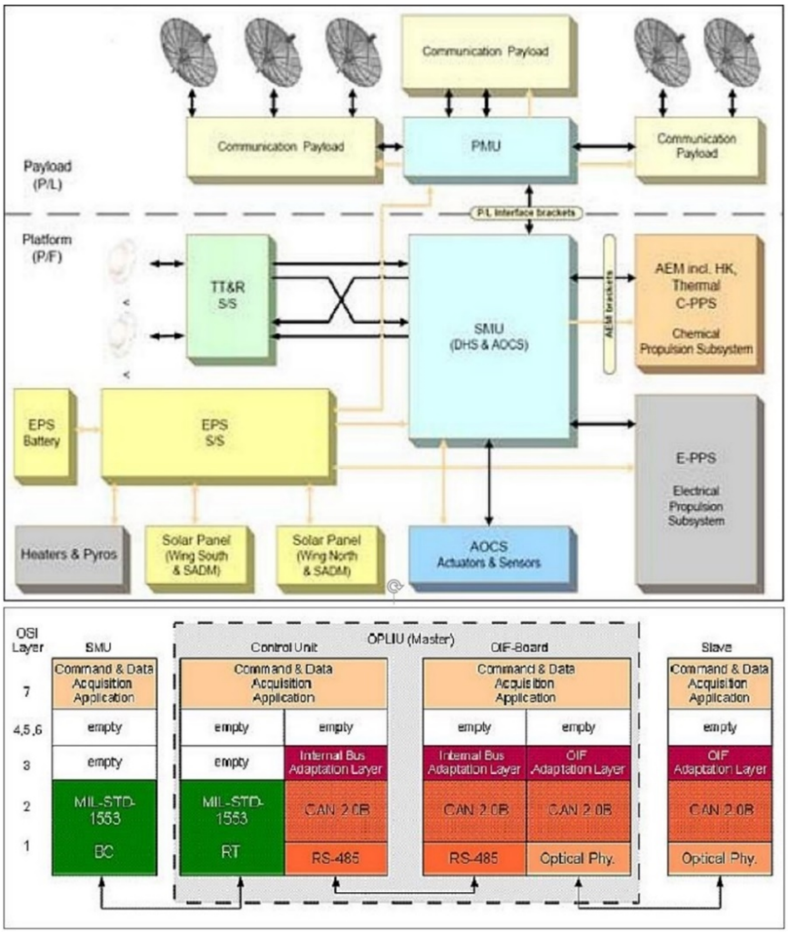


Figure 3.6: Conceptual view of SGEO subsystem electrical interfaces and network protocol stack [60]

On modular satellite systems, where the mission payload and the satellite bus shall be connected by standardized interfaces and where every unit or module, even the payload itself, can be exchanged or reconfigured due to the modular concept requirements, the data bus concept and the data interface between the modules shall provide functionalities and the necessary design to handle the two data traffic types on same time. According to the different characteristics of the data traffic types listed in Table 2.1, two different communication principles are recommended for the different data traffic types, time-triggered communication for housekeeping and satellite control data traffic due to its determinism and real-time capability and event-triggered communication for the payload data traffic.

There are two approaches to handle this problem, the physical separation of the data traffic by using 2 separated bus systems similar to data network design applied on monolithic satellites and the logical separation of the two main data traffic on one physical data bus using network protocols of higher layers of the OSI model [49].

3.1.3.2 Physical separation of data traffic types

The physical separation of the two types of data traffic aboard satellites refers to the known approach used on monolithic satellites. In this approach, two physically separated networks are implemented on the satellite, one for payload data traffic and one for housekeeping and satellite control data traffic. For modular satellites, where the payload modules shall be able to communicate with the satellite bus over standardized data interface, this results to the fact, that the data interface between the modules must provide 2 independent parallel data links.

The advantages of this approach for a modular satellite are:

- Physical separation of the two networks, failures of one communication network do not affect the functionality of the other network.
- Choice of the appropriate data bus systems for the data traffic types is significantly simplified, because the challenge of contrary requirements is solved by using two different communication bus standards. Suitable candidates with appropriate specifications for each data traffic type can be deployed.
- For both data traffic types, many space qualified bus systems can be used and so, space heritage is given.

On the other side, the complexity of the data communication is shifted from the software to the hardware. The disadvantages of this approach can be concluded as:

- Significantly higher complexity in the design of the data interface for the data transmission from module to module on the modular satellites
- More harnesses and more weight (two physically separated data busses require more network devices (transceivers, cable, etc.))
- Limited flexibility, reconfigurability and scalability
- More resource utilization for managing two parallel data networks

A possible realisation of that approach is presented in [49] with SpaceWire as payload data bus and CAN as housekeeping and satellite control data bus.

3.1.3.3 Logical separation of data traffic types

With this approach the critical housekeeping and satellite control data traffic and the payload data traffic shall be transferred by one single data bus system. One advantage of this approach is obviously the simpler design of the data interface between the building blocks. Using certain communication protocols and mechanisms on software level, different message types with various criticality levels can be transferred in the same network. Hence, the complexity is here mostly located on the software side.

Advantages of this approach:

- Simplified design of the data interface, a design of the data interface using single-line data transmission (e.g. optical data communication) can be considered and developed.
- High flexibility of the data communication (defining, deactivating and reactivating of data paths realized in network management software)
- Compliant with the modular satellite concept
- Innovated and future-oriented approach
- Upgradable

Disadvantages:

- High complexity of network management and data routing
- Very stringent choice of the communication architecture
- Significant efforts for adaption and modification of existing communication standards

In this thesis, this approach is adapted for the data communication concept on modular satellites.

3.2 Requirements to the data communication architecture of modular satellites

An appropriate design of a communication architecture for the modular spacecraft as a mixed critical distributed embedded system presumes an identification of the requirements to the on-board communication system. An additional important constraint of the communication architecture is the availability of the needed hardware. Regarding the conceptual considerations discussed above, requirements to the on-board data communication system of the highly modularized satellite design will be broken down from the high-level requirements to a modular satellite and presented in following [49].

System requirements: Requirements to the on-board data communication system are derived from the modular architecture design of spacecraft. Modular spacecraft design promises high potential and great benefits, as described in Chapter 2.2, like time and cost saving in development and qualifying processes as well as rapid-production-on-demand capabilities. Beside of that, one of the key features of the modularized design is the capability for on-orbit servicing operations, in which the modular satellite can be reconfigured, repaired by replacing faulty module with a new one or extended with new modules to enhance the system performance.

- **Reconfigurability:**
    - One of the main advantages of the modular spacecraft is the on-orbit reconfigurability and serviceability. The integrated communication network shall tolerate the coupling and decoupling of one or more network nodes during operation. Coupling and decoupling of network nodes shall not result in network instability or persistent network failures.
    - The system network shall detect temporary as well as permanent adding or removing of network nodes. All needed changes of the configurations by deactivating/reactivating of data links and network managements must be considered.
    - The system communication network shall be shut down and restarted without causing any instability.
    - Rescheduling of the on-board communication if using time-triggered network after a reconfiguration of the building blocks or after adding or removing of a network node shall be done in a fast, correct and reliable manner.
    - The on-board communication network shall provide features in hardware and software to apply new network configurations or new communication schedules in the system in autonomous manner.
- **Extensibility**:
    - The data communication network shall not have an upper limit for the number of network nodes.
    - The data communication network shall be able to be extended with new network nodes
- **Dependability (Reliability and Fault tolerance)**:

- The data communication network shall work in a dependable manner.
- The data network system shall be maintainable. The network shall be able to detect link or node failure and provide appropriate countermeasures to ensure the system communication.
- The downtime of the system caused by failures shall be minimized.
- The data network system shall work in a reliable manner.
- In case of the failure of one or more network nodes caused by internal or external events, the network shall be able to operate with the remaining functional nodes in a stable manner.

Other requirements to the on-board communication system of modular satellites

- **General**
    - The communication architecture should support symmetrical and at least half-duplex communication. Full-duplex communication is however desirable.
    - The communication architecture shall provide broadcasting and multicasting abilities.
    - The communication architecture shall be masterless or multimaster.
    - The communication architecture shall provide error detection mechanism (e.g. cyclic redundancy check).
    - The communication architecture shall provide appropriate mechanisms for collision avoidance.
    - The communication architecture shall provide appropriate mechanisms to handle network loops in the physical network topology.
    - The communication architecture shall have fault isolation and provide ability to eliminate error propagation (fault containment).
    - The communication architecture shall be able to detect a babbling idiot failure and to exclude the node with that failure from the network communication.
- **Requirements for payload data traffic**
    - The communication architecture shall provide a data rate of at least 100 Mbps for payload data traffic.
    - The P/L data communication architecture should be event-triggered. However, if a time-triggered architecture is chosen, appropriate pre-emption buffer shall be regarded.

    - If the P/L data communication is time-triggered, it shall be able to split payload data packets into chunks transmitted over several transmission cycles. The receiver shall be able to rebuild the packets by lossless merging the data chunks.

- **Requirements for housekeeping and satellite control data traffic**
    - The communication of housekeeping and control data shall be deterministic.
    - The data rate shall be at least 500 Kbps.
    - The communication architecture shall provide scalability. There should be no design decision that makes it difficult to extend the network to higher data transmission rate or to restrict the number of nodes.
    - The communication architecture shall be fault tolerant and provide ability to detect communication and node failure.
    - The communication architecture shall employ distributed global time synchronization.

3.3 Comparison of data communication systems

In order to find an appropriate communication architecture for the on-board data communication system of modular satellites, it is firstly necessary to get an overview of the known communication standards applied on spacecraft or in similar terrestrial systems and applications.

There are several communication architectures known for their deployment in distributed control systems in industrial applications. Although all of them are successful in their application fields, the requirements for aerospace applications differ significantly from those for terrestrial industrial applications. Therefore, the communication standards selected and discussed in following shall either be already used in aerospace applications or, at least, to have the potential to be used in future space missions. Further, to be applicable as the intermodular data communication system aboard modular spacecraft, the regarded communication architectures shall include the definition of the physical layer, media access control, data flow control and some level of error detection/correction.

Simple serial buses, which are only defined by electrical specifications like I2C, SPI, RS-232, etc., will not be regarded in the comparison. An also important criterion is the availability respectively the acquirability of hardware as COTS and needs to be regarded in the choice of the candidates [20, 49, 62, 63].

The following communication architectures are chosen and discussed in this thesis:

- **MIL-STD-1553:**

  MIL-STD-1553 is the most commonly used serial communication architecture in military and civil avionic and aerospace applications. The very high reliability of this standard is given by its origin design as an avionic data bus for military avionics. The standard defines a redundant line physical layer using the twisted pair cables and support the multidrop topology. The main design specification of this standard is focussed on the extremely high reliability. MIL-STD-1553 bus systems are commonly implemented in a dual redundant configuration for critical applications.

  An optical implementation of this bus using optical fiber is known as MIL-STD-1773. Another very promising approach for the optical substitution of the physical layer of MIL-STD-1553 is also presented in [64]. The maximum data rate of MIL-STD-1553 is 1 Mbps. The maximum node number is limited at 31 and the media access is controlled by the master-slave-principle. Since bus messages can only be transmitted at a defined timeslot determined by the bus master, MIL-STD-1553 can be considered as a deterministic communication data bus with real-time capability [65–67].

  This communication architecture is not suitable for a deployment on modular spacecraft although its widely deployment in several space missions and its high reliability. The most significant weak points are the strongly limited flexibility (no masterless or multimaster ability) and the modest maximum node number of 31, therefore, it is not suitable for modular satellite designs which shall provide the ability of extension. This bus standard is a typical example for a very reliable data bus with no flexibility and reconfigurability.

- **TTP/C:**

  TTP/C is a time-triggered fault tolerant fieldbus developed at the Technical University Vienna and designed for hard real-time communication in automotive area and is suitable for high-speed single-failure operational safety-critical applications. It is deployed for control data communication in many terrestrial transportation systems like automotive, railway and aviation. It is deployed in the air cabin pressure control system of the Airbus A390.

  The fault tolerance and high reliability of TTP/C are given by a strict time schedule with known delay and bound jitter over dual redundant communication channels. The time base is secured by a fault-tolerant global system clock. Further functions and services as autonomous message transport, the capability to implement redundant nodes and to execute of redundant functions on multiple nodes complement this standard. A failure of a single communication node does not result in a disruption of the communication of the remaining nodes. Fault containment, hot swap function and availability of radiation-tolerant hardware complete the portfolio of this communication standard.

TTP/C can be deployed on different physical layer, the maximum data rate is 5 Mbps using RS-485 physical media and can be extended to 25 Mbps if using Ethernet physical media. A very important feature of TTP/C which makes it to a possible candidate for modular satellites is the ability to handle time-triggered communication and event-triggered communication in one network. This feature meets the requirements of the logical data traffic separation approach mentioned above [20, 69–71].

- **CAN:**

  The CAN bus is a multi-master serial fieldbus standard from the automobile industry. It is widely used as communication bus between microcontrollers in vehicles without a host computer. The CAN bus uses carrier sense multiple access with collision detection and arbitration on message priority (CSMA/CD) for message arbitration. CAN bus only supports the bus topology. Although it is not deterministic, it has been successfully deployed on several small satellite missions like the BEESAT series of the Technische Universität Berlin [72]. Two extensions of the classical CAN bus are considered below.

  - **TTCAN**

    TTCAN is the application of time triggered protocol on CAN and enables a time triggered communication on CAN bus. TTCAN can be implemented in software or hardware to use a system matrix that defines a schedule for message transmission over a communication cycle. This schedule includes slots for specific messages that are sent every cycle and slots for standard arbitration, so event-triggered messages can be transmitted. TTCAN still uses CSMA/CD, as implemented in standard CAN controllers, to ensure proper arbitration during the arbitrated frames. During the scheduled frames there should be no bus contention, and the arbitration service will not be used. TTCAN can only be implemented on CAN controllers with the capability to turn off the retransmit feature. The supported topology of TTCAN is multidrop. The maximum transmission rate is 1 Mbps but is typically lower in application, on the order of 500–650 Kbps. Due to the time triggered approach the communication in TTCAN is deterministic [75, 76]. The low transmission rate of TTCAN makes it only useable as a HK and satellite control data bus for modular satellites.

  - **CANAerospace:**

    CANaerospace is a higher layer protocol based on CAN standard developed in 1998 for aeronautical applications. CANAerospace has continuously been developed and was published as Advanced Aviation Transport Experiments Databus Standard in 2001 by NASA. It is designed for the highly reliable communication of microcomputer-based systems in airborne application via CAN. CANAerospace supports the masterless communication and specifies the electrical characteristics, bus transceiver requirements and data rates based on CAN. It uses the special time-triggered concept of managing the available bandwidth of a multidrop CAN network to ensure predictable behaviour [49, 77].

- **SpaceWire**:

  SpaceWire is a communication standard for aerospace applications which was standardized by the ECSS in 2003. It has been successfully applied on several space missions of ESA, NASA, JAXA and many other organizations and companies. SpaceWire is characterized by its high-level reliability and the maximum data transfer rate of 400 Mbps. In addition to that, more and more equipment suppliers for space application favour the integration of SpaceWire interfaces to their products. There is no upper limit for the number of nodes in a switched SpaceWire network. Rad-hard SpaceWire hardware can be purchased from the STAR-Dundee company in Scotland or from many other companies over the world [49, 79]. A further development of SpaceWire is SpaceFibre, which is a very high-speed serial data link which is intended for used in data-handling networks for high data-rate payload. SpaceFibre can operate over fibre-optic or single-wire copper cable and supports data rate up to 2 Gbps [80].

- **IEEE-1394B (FireWire):**

  IEEE-1394B also known as FireWire is a serial peripheral bus standard for high-speed communications and isochronous real-time data transfer. The physical media is specified over twisted pairs, optical fiber and wireless. The currently available data rate is at 800 Mbps and can be up to 3.2 Gbps.

  The maximum number of nodes on a single bus is limited by 63, but the network can be extended to 1032 buses by addressing. Although it is not specially developed for safety critical systems, IEEE-1394B still generated much interest in aerospace applications because of its high data rate and the availability of intellectual property (IP) core for use in the fabrication of application specific integrated circuit (ASIC) devices.

  IEEE-1394 supports tree topology and does not support loops, but the protocol provides the capability to disable and enable ports. At start up, an identification process is used to provide addresses to the nodes, select root nodes, and isochronous master. Adding or removing devices requires the identification process to execute again. Due to its missing fault containment and babbling idiot avoidance, IEEE-1394B does not provide the necessary reliability for a deployment in modular satellites [20, 49].

- **TTEthernet:**

  TTEthernet (defined by SAE AS6802) is an expansion of classical Ethernet standard with time-triggered protocol to provide the services to meet the requirements of deterministic and safety-critical systems. It is developed by the Technische Universität Wien und TTTECH based on the experiences with TTP and claims to have reunified the best properties of TTA and Ethernet.

The standard is fully compatible to IEEE 802.3 Ethernet and integrates transparently with Ethernet network components. This technology offers deterministic real-time scheduled communication and conventional asynchronous traffic parallel on the same network. Currently, COTS hardware is available with up to 1000 Mbps data rate. It is able to transfer event-triggered messages and time-triggered messages over the same physical layer by preemption the event-triggered messages until the time-triggered messages, which is marked by a specific name field in the header, are completely transmitted. Two different time synchronization strategies are supported, time synchronization and fault-tolerant time synchronization.

TTEthernet can operate in 3 operation modes with different criticality levels. This communication standard also provides a seamless communication system for all types of distributed non-real-time and real-time applications, from very simple uncritical data acquisition tasks, to multimedia systems and up to safety-critical control applications. It should be possible to upgrade an application from standard TTEthernet to a fault-tolerant configuration with minimal changes to the application software. Furthermore, a fault-tolerant global time base and strong fault isolation mechanisms guarantee the determinism of this communication architecture.

In 2012, TTEthernet is deployed as single network for both critical data and payload data in the crew module of the ORION MPCV (multi-purpose crew vehicle) developed by NASA and in the service module developed by ESA. Thus, space heritage of this communication standard is also given [84, 85].

The communication architectures presented above are compared to each other in several characteristics and features. The bus analysis is completed by a comparison matrix with several decision criteria based on the requirements to the on-board data communication system for highly modularized satellite with OOS capability in Section 3.2. A section of the comparison matrix with the most relevant criteria is shown in Table 3.2. In the table, the red highlighted cells correspond to the characteristics or features of the bus standards which are not compliant to the requirements for an application in a highly modularized satellite architecture with OOS capability. The complete comparison matrix is contained in the appendix.

Table 3.2: Comparison matrix of the communication architectures [8, 9, 62]

Features /bus standards	MIL-STD-1553	TTP/C	TTCAN	SpaceWire	IEEE-1394B	TTEthernet
Communication Control Scheme	Master-Slave	Time-triggered and event-triggered	Time-triggered and Event-triggered	Event-triggered	Isochronous data transmission controlled by cycle master, asynchronous transmission arbitrated by cycle master in the left time	Time-triggered and event-triggered
Multimaster or masterless	No	masterless	masterless	masterless	No	masterless
Determinism	Yes, since controlled by bus master	Yes, for time-triggered mode	Yes, during time triggered mode	No	No	Yes, for time-triggered mode
Maximum Data Rate	1 Mbps	25 Mbps	1 Mbps (125 Kbps for fault tolerant mode)	400 Mbps	800 Mbps	1000 Mbps
Maximum Node Number	31	64	120	224	63 on a bus with up to 1023 buses	Depends on the number of available switch ports
Topology	Multidrop	Multidrop, point-to-point, star by using hub	Line bus topology	Point-to-point and switched	Peer-to-peer, tree by using bridges	Point-to-point, mesh and switched topology
Fault Tolerance (Fail-operation/Fail-Safe) Mode	Yes	Yes	Yes	No	No	Yes
Network loop tolerance	No	No	No	No	Yes, by disabling ports to break loop	Yes, with dynamic routing

The comparison of the regarded communication architectures can be summarized in the following conclusions [49]:

- Although the proven high safety and fault tolerance, MIL-STD-1553 does not provide the flexibility and reconfigurability needed for an application on modular satellites especially regarding to its master-slave-scheme. Furthermore, the low data rate, the missing extendibility and limited node number of the network are additional counterarguments of this standard.
- TTP/C is a fault tolerant fieldbus with high reliability and several features which would meet many requirements of the modular spacecraft architecture. Especially the capability to handle two data traffic types (time-triggered and event-triggered) in the same network is a great advantage. However, the defined maximum node number of 64 nodes is a limit for large modular space systems.
- TTCAN can be used on a modular satellite due to its reliability and determinism, but its low data rate only allows an application as housekeeping and satellite control data bus, a secondary bus with high data rate will additionally be needed for the P/L data traffic.
- SpaceWire is a space-qualified bus standard with high data rate but it is primarily developed as a P/L bus. Missing determinism and fault containment make it only be adaptable on modular satellites in combination with a reliable housekeeping and satellite control data bus. Further, one contribution to the space-qualifying character of SpaceWire is given by its definition of the highly shielded and robust physical layer. On a modular satellite, where data need to be transferred between building blocks and these building blocks need to be designed with the capability of coupling and decoupling on orbit for OOS, the highly reliability of the physical layer must be given up or result in a very complex design for the data transfer between modules to keep its defined reliability.
- FireWire is obviously not suitable for modular satellites since several critical requirements to the on-board data communication system are not met by this standard.
- TTEthernet seems to be the most suitable solution for an application on modular spacecraft, since it meets several critical requirements. However, a deployment as the main data communication system aboard modular satellites, further complex concepts to make a time-triggered communication system reconfigurable need to be developed and implemented on top. Additionally, the network management software must provide methods to detect and handle specified failures of the modular systems.

3.4 Application of TTEthernet as the global data bus on modular spacecraft

The most promising candidate for the on-board data communication bus which can meet several requirements of a modular satellite can be found in the time-triggered Ethernet standard TTEthernet. TTEthernet defines an expansion of classical Ethernet standard with time-triggered protocols to provide the determinism and reliability needed for safety-critical systems. TTEthernet supports the design of communication systems where applications with mixed time criticality share a single physical network [84, 85]. The 3 parallel operation modes of TTEthernet are:

- **Time-triggered mode**: In this operation mode, the TTEthernet network is fully time triggered. Messages are sent at predefined times. The occurrence and latency of the transmission is guaranteed.
- **Rate-constrained mode**: This mode is used for applications with less stringent determinism and real-time requirements. Bandwidth is predefined for each application and delays and temporal deviations have defined limits.
- **Best-Effort mode**: In this mode the network operates like the classical Ethernet. Best-effort operation mode shall be used for transmission of large data without any critical requirements.

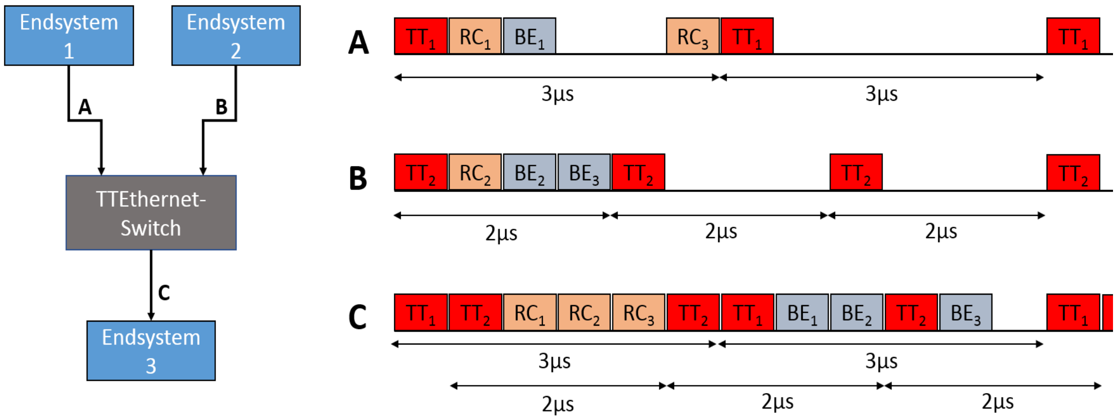


Figure 3.7: Exemplary transmission of the 3 data types in TTEthernet

All three traffic modes can be operated together within the same physical network. The transmission of time-triggered (TT) communication has the highest priority and cannot be hindered by less time-critical event-triggered rate-constrained (RC) or best-effort (BE) messages. The bandwidth which is not taken by time-triggered messages can be used for the RC or BE communication. Further, RC traffic is prioritized over BE traffic [84].

Using the TT communication mode, critical data communication as the HK and satellite control data traffic can be transferred deterministically and in real-time. BE communication can be used for P/L data traffics. The high transmission speed and bandwidth of TTEthernet can meet the requirements of data rate for the P/L data traffic. RC communication can be used for event-triggered but critical messages.

Because of the capability to the classical Ethernet standard, system components or payload units with Ethernet interface can be fully integrated in the network. This feature presents an extraordinary advantage since the Ethernet interface has become more and more a standard interface in many space components in the last years.

3.5 The scheduling problem of the on-board data network of modular spacecraft

On highly modularized satellites, components of subsystems can be spatially distributed in several modules. For example, the ACS subsystem on spacecraft usually consists of several components like sensors for attitude measurement e.g. star tracker, sun sensor, earth sensor, magnetometer, gyroscopes etc. and actuators e.g. reaction wheels, propulsion system thrusters, etc. to stabilize and control the spacecraft attitude so that its downlink antennas may be accurately pointed to earth for communication or for other reasons. Depending on the type of the components, their allocations on spacecraft is strictly constrained. Earth sensors, for example, must be integrated on the earth pointing side of a satellite while magnetometers or reaction wheels can be integrated inside of the spacecraft bus. On a modular satellite, the components of ACS subsystem are allocated in different modules. Sensor raw data can be periodically gathered and processed by the local OBC in the same module and then transmitted to the OBCs on which the ACS control algorithms are implemented. The derived commands to the actuators need to be transmitted to the [18]actuators which are perhaps allocated in different modules. The data exchange between the components of such critical subsystems e.g. ACS must be synchronized and coordinated in a reliable manner. Regarding to this, highly modularized satellites are distributed real-time systems whose system performance does not only depend on the delivery of the critical data between their modules but also on the time at which these data are delivered.

Spacecraft system functions like attitude control, thermal control, etc. can be represented as services and each service consists of a set of jobs which are accomplished by different devices in different modules. These jobs can have different criticality levels and are classified in three categories [18, 88, 89]:

- Periodic jobs with the a priori defined attributes like period, deadline, required network resources, etc.
- Sporadic jobs without defined invocation times but defined minimum arriving times
- Aperiodic jobs without any timing constraints.

To ensure the delivery of data packets on time to meet the deadlines of jobs, the duration of all computational and communicative activities of the relevant components must be known a priori. The end-to-end latency of a job is the time duration between job initiation and job termination and is defined as the Worst-Case Execution Time (WCET) of a job.

Depending on the on-board applications which invoke these communication jobs, there may exist dependencies between the jobs. Some jobs cannot be executed before the completion of their predecessor jobs. For example, control variables for ACS actuators cannot be calculated and transmitted before the needed sensor data for this control cycle are available.

The dependency relationships for every service can be described by using a Directed Acyclic Graph (DAG) [90–92]. Figure 3.8 shows an exemplary DAG for an ACS service of a spacecraft. Each job in the DAGs must provide the following parameters:

- Job ID
- Message size
- WCET

Further parameters of the graphs are criticality level, service deadline, traffic type and timing parameters like period of the service, minimum arriving time, etc.

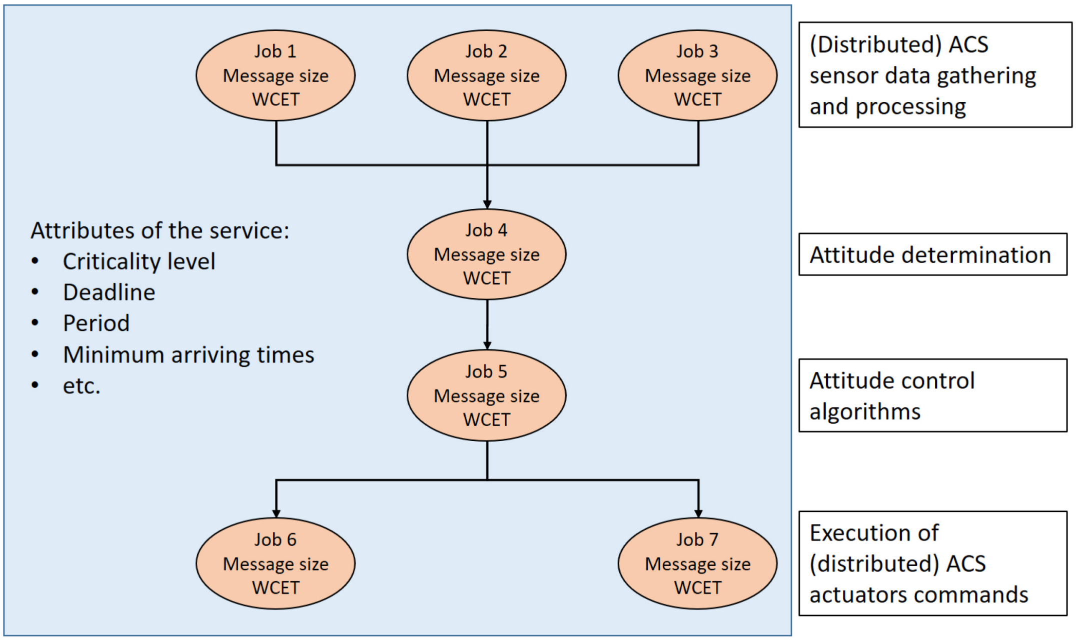


Figure 3.8: Acyclic directed graph of the ACS service example

The DAGs of the services of the spacecraft encompass information about services and can combined to generate inputs to build a logical network model of the on-board communication network.

Beside of the logical model, physical characteristics of the data communication network are contained in the physical network model. Physical network model includes information about the types of network nodes like end-systems, switches and media types of the connections between nodes, the physical network topology, the transmission speed between connected nodes, etc.

The logical and physical network model are mathematical abstraction of the communication network of a distributed real-time system and can be used as input for scheduling the communication. The scheduling problem is solved when a feasible communication schedule for the set of system services is found which can meet all the requirements of the planned distributed real-time system inclusive the reliability and deterministic constraints of the on-board data communication network of a highly modularized satellite. Additionally, some optimization objectives can be defined like e.g. minimized maximum end-to-end latency of the critical messages or maximizing the bandwidth utilisation efficiency or jitter minimization, etc.

However, optimal scheduling for distributed real-time systems is a NP-complete problem due to the fact that the complexity of solving grows with the number of network nodes [93, 94]. Overviews of several solving approaches for the scheduling can be found in [18, 90, 92] and will not be handled further in this work.

3.6 Concluding remarks

An important contribution in this chapter is given by the conceptual considerations for designing the on-board data communication system for reconfigurable and highly modularized satellites. It seems that the mesh topology is the most suitable physical topology for the on-board data network since it provides several redundant data paths as a possibility to enhance the dependability of the communication. Due to several discussion in the area of modular robots, a comparison between different bus scheme is made and for modular architectures, the hybrid bus scheme is recommended.

Regularly, on spacecraft, payload data traffic and satellite control and housekeeping data traffic are usually physically separated since in a typical satellite system production flow, development of mission payload and satellite bus are co-running processes and mostly as enclosed system with interface to each other. Additionally, the requirements to the communication systems of these two data traffic types are in some cases even conflicting. For modular satellites, two competitive approaches are regarded and compared to each other, the physical separation of the two busses similar to the solution approach for classical monolithic satellites and the logical separation of the two data traffic types on one bus. Both approaches can be applied on modular satellite but result in different level of complexity in different areas. While physical separation simplifies the choice of bus standards for each data traffic type as the most qualified bus standards are defined specially for one data traffic type. On the other side, it limits the flexibility and reconfigurability of the modular system and complicates the design of the data interface between the modules. Logical separation is more suitable for modular architectures, but the choice of a suitable bus standard is very stringent.

Based on the mentioned conceptual considerations and the fundamentals in the previous chapter, requirements to the on-board data communication system of reconfigurable and highly modularized satellites are concluded in Section 3.2.

A further contribution of this chapter is the comparison of different communication architectures to their applicability on modular satellites. It seems that TTEthernet is the most suitable candidate since this standard provides many features to meet the described requirements. But the time-triggered communication scheme of TTEthernet is a limitation on the desired reconfigurability of the data network on a reconfigurable and serviceable modular satellite. Therefore, further protocols, methods and mechanisms need to be defined or developed and adapted to complete the concept of the on-board data communication system of modular satellites.

4 HARDWARE CONCEPTS FOR THE DATA COMMUNICATION SYSTEM

This chapter contains the design concepts of the network hardware needed for the on-board data communication system for modular satellites. The requirements and constraints are determined by the physical and logical network layout described in the previous chapters. The design concept was realized as hardware for the planned In Orbit Demonstration (IOD) satellite in the iBOSS project.

In this chapter, section 4.1 gives an overview of the network components needed in every module of the highly modularized satellite designs. In Section 4.2 the design of the data interface to transfer data between the modules of the modular satellite is described and completed with the test results of the data interface. The multifunctional component OBC-Switch-Board is an specified design containing the OBC, the TTEthernet switch and the TTEthernet end system integrated on one board. The development of the OBC-Switch-Board is described in Section 4.3. Section 4.4 concludes the chapter with concluding remarks.

4.1 Network hardware in the module

A building block of a highly modularized satellite shall provide a TTEthernet switch since every building block can be connected to multiple neighbouring building blocks. Data packets from the OBC or from other network participants in the building block can be sent to other building blocks of the satellite via the data interfaces connected to the TTE-Switch. Vice versa, data packets to the OBC or to other network components within the building block will firstly arrive at the TTE-Switch and are then routed to their destinations. Mission payloads with an Ethernet interface can be integrated in the TTEthernet network and send data packets via BE traffic communication with high bandwidth to the OBC inside of the same building block or in other building blocks for processing and transferring to the ground station via downlink.

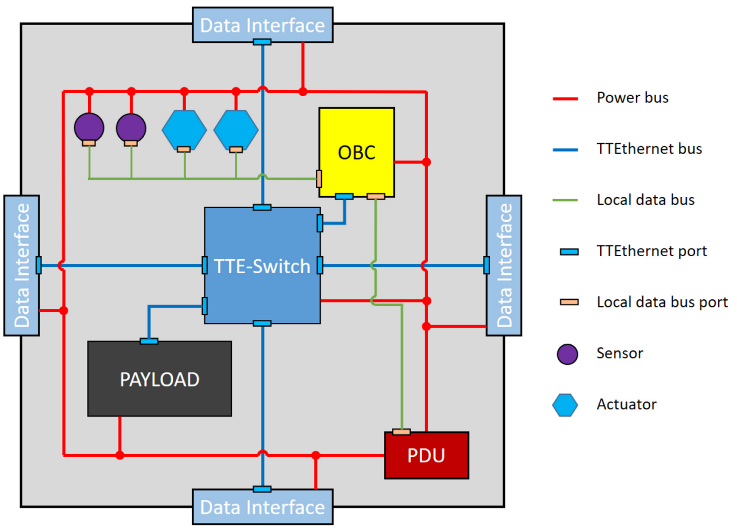


Figure 4.1: Block diagram of the network hardware and the components inside of a building block

Figure 4.1 shows the data and power bus concept of a building block on a highly modularized satellite.

Thus, the TTE-Switch implements the functionality of a router since it routes the data packets of other building blocks incoming from a data interface to another interface without involving the OBC. Beside of the OBC, the TTE-Switch and the optional payload components, any components with Ethernet interface can be integrated in the global data network of the modular satellite via TTEthernet. So, components of a building block can be addressed by OBCs or other network participants of any building block on the satellite. Components without Ethernet interface like small sensors or actuators can be connected and controlled by the OBC via a local data bus. Optionally, the power distribution unit (PDU) of the building block can be equipped with an Ethernet interface and integrated in the global network [48]. In Figure 4.1, the PDU is connected to the OBC via the local data bus.

4.2 The Optical Data Interface

The data interface incorporates the function of data transmission into or out of a module of the modular satellite. Every module can be connected to the neighbouring modules via data interfaces. In a modular satellite consisting of modules coupled with each other, data transferred from a module to another module may pass several data interfaces on their paths. There are many concepts for the data transmission between modules. A discussion about different data transmission methods follows in the next subsection.

4.2.1 Data transmission method between building blocks

Several different standards for short-range communication have been considered for the application aboard spacecraft. Wired data transmission standards transmit data as electrical signals along a metal conductor and are known as the most mature and reliable method of data transmission for space applications. For modular satellites, an application of optical data transfer or data transfer via radio frequency for the data transmission between building blocks is discussed in [49].

Figure 4.2 shows an overview of different data transmission technologies.

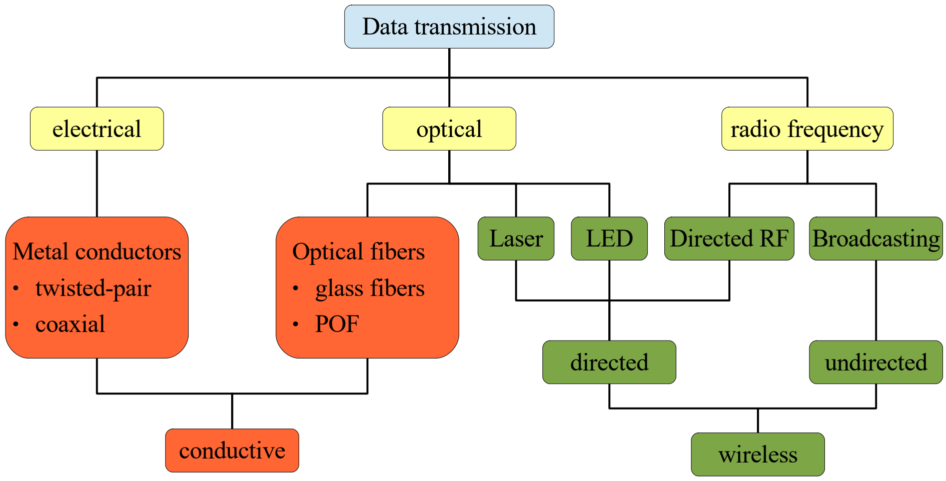


Figure 4.2: Overview of data transmission technologies known in terrestrial applications [13, 49]

In this thesis, the approach of optical short-range communication is proposed for the data transmission between the modules of a modular satellite. The advantages of using optical communication for data interface over conventional electrical systems are galvanic isolation, fewer electromagnetic interference/electromagnetic compatibility (EMI/EMC) concerns, high data rate and reduced weight. Ease of integration reduces the time to integrate and test an interface, which consequently leads to cost savings. Fewer EMI/EMC concerns reduce design as well as test time. Furthermore, the approach of short-range optical communication allows a single-line data transmission between the building blocks. Especially for the Ethernet standard, there are many mature concepts of electrical-optical conversion. Availability of COTS components is therefore given [49].

4.2.2 Requirements of the Data Interface

The data interface for the data transmission between building blocks of a highly modularized satellite design shall comply with the following requirements [48]:

- The data interface shall be universal deployed (no male-female pairing).
- The data transmission over the data interface shall be apply the short range optical data communication principle.
- The data interface shall provide an Ethernet interface.
- The data transmission via the data interface shall base on the Ethernet standard.
- The optical frontends shall be accommodated in the centre of the mechanical interface.
- The data transmission shall work over a distance up to 2cm between two data interfaces.
- The data rate of the optical data transmission shall be at least 800 Mbps.
- The communication shall be full duplex and bidirectional.

- The design of data interface shall be rotationally symmetric.
- The data transmission over the data interface shall be robust to an angular misalignment of up to 2° and an axial misalignment of 1mm as well as to reflections of the own optical beams or diffuse ambient light.

Choice of the components

- Used electronic components on the PCB shall have an operating temperature range between -30°C and +80°C.
- Only ceramic capacitors shall be used. Using of electrolytic capacitors shall be avoided.
- Used oscillators shall withstand shock and vibrations.

Production of PCBs

- The PCB surface finishing method shall be chemical tin, since ENIG or gold surfaces are more sensitive to temperature fluctuation and provide less stability.
- Used solder paste shall not be lead-free, since a solder paste with more than 70% tin can cause the Whisker effect.
- The soldering technology shall be vapour phase soldering, since this causes less thermal stress to the components. Furthermore, it does not involve the risk of overheating and is oxygen-free.
- The insulating backbone of rigid PCBs shall be FR-4.
- The solder mask shall meet the outgassing requirements of the NASA specification SP-R-0022A.

Connectors and cables

- Used connectors shall be locked together by screw connections to prevent inadvertent disconnection.
- The used cables shall be Teflon PTFE cables.

4.2.3 Hardware design of the Optical Data Interface

In the iBOSS project, the first concept of the data interface developed for the modular satellite concept based on optical fiber and a glass lens. In this approach, either the integrated switch shall provide optical interface for the optical fiber or an additional media converter for the bidirectional electrical-optical conversion needs to be integrated in every building block. The advantage of this concept is the reduced weight of the optical fiber comparing to the copper cable and the increased EMC characteristics and the very simple concept of the data interface without any electronic.

On the other side, the trade-off of this concept is the increased AIT effort caused by the limited minimum bending radius of optical fiber and the limited experience of the application of optical fiber in spacecraft especially regarding to temperature fluctuations and irradiation in space environment. This leads to the relocating of the bidirectional electrical-optical conversion in the data interface in the latter phase of the iBOSS project. As a consequence, the data interface must provide some intelligence to convert electrical Ethernet signals to optical signals and vice versa. From the network view, a coupled pair of two data interfaces can be regarded as a mere physical-layer substitution with nearly no impact on the network architecture and hence absolutely transparent to the communication bus nodes.

The prototype of the data interface in iBOSS was developed in cooperation with the Fraunhofer Gesellschaft IPMS in Dresden, Germany. IPMS provides a lot of know-how, expertise and modern R&D infrastructures in the field of optical sensors and actuators, integrated circuits, microsystems and nanoelectronics. Especially in the field of optical data transmission the IPMS has been working intensively for the last 10 years and has developed amongst other products an optical wireless communication link with up to 3 Gbps data rate for nearfield optical communication [48, 95].

The developed data interface consists of a frontend part (printed circuit board (PCB) with integrated optical frontend) and a backend part (PCB with Ethernet interface) connected by a high frequency suitable cable, as shown in Figure 4.3.

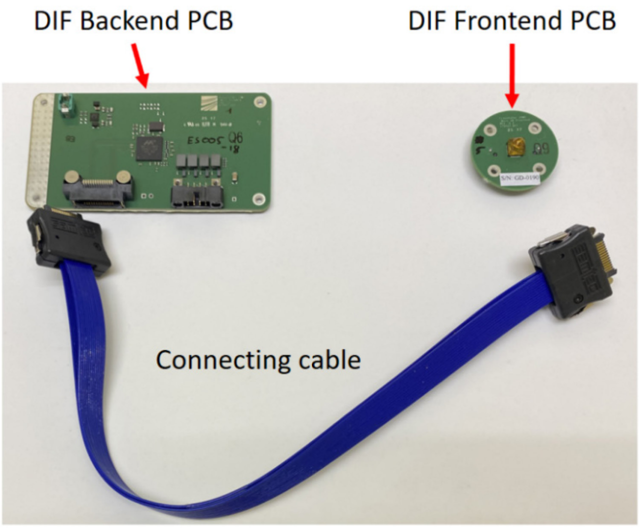


Figure 4.3: Components of the data interface

Due to the result of the bus comparison in Chapter 3, TTEthernet was chosen as the data bus system for the on-board data communication concept. Because of the known compatibility of TTEthernet to standard IEEE 802.3 Ethernet, the data interfaces shall be designed to convert Ethernet package coming from the TTE-Switch in the modules into optical signals, transfers these to the counterpart of the neighbouring module, where the optical signals are reconverted into electrical signals of Ethernet packages and forwarded to the destination network nodes. Figure 4.4 shows the block diagram of the communication path between two TTE-Switches via a pair of coupled data interfaces.

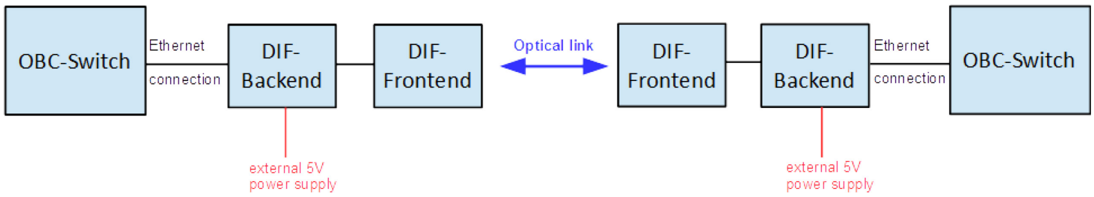


Figure 4.4: Block diagram of the connection between the data interface and the network switch [48]

The frontend PCB of the data interface is accommodated in the center of the mechanical interface of the iSSI and meets the iBOSS requirements to the iSSI in respect to the rotation symmetry, androgyny and coupling/decoupling capability. The DIF backend PCB can be mounted behind the iSSI inside of the building block.

4.2.3.1 Data Interface Backend

The backend part of the data interface hosts the intelligence needed for the conversion of Ethernet packages into discrete signals for the driver of the optical frontend. The DIF backend PCB meets all the components and production requirements mentioned in Section 4.2.2.

The data interface backend provides 3 electrical interfaces and a thermal interface. Beside of the power connection for a power supply of 5VDC and a supply current of maximum 500mA, it has an interface for the connection to the frontend and an Ethernet interface. The dimensions of data interface backend are 85mm x 45mm.

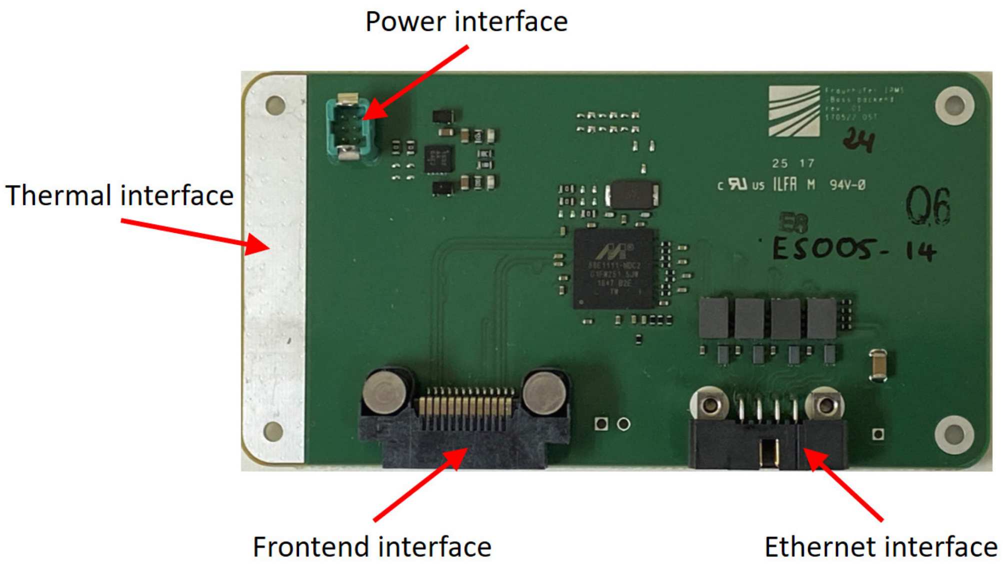


Figure 4.5: Connectors of the DIF backend PCB

The power connector of the data interface backend is a Harwin Gecko G125-MV10605L1P connector. The data interface backend needs a positive power supply of 5V and a supply current of 600 mA. Via the Datamate J-Tek Male M80-5400842 connector, the backend can be connected to an Ethernet network or directly to a PC via a Cat-5 cable. The connection between the data interface backend and the frontend is realised by a cable of 25 cm and two latched connectors of SAMTEC [48].

A concept for thermal dissipation is realised on the data interface backend to meet the requirements of the data interface especially regarding to the temperature fluctuations in space environment. In this concept the multilayer PCB of the backend is extended with some heat conducting layers which transfer the heat produced by the electronic components integrated on the board to the thermal interface on the mounting edge of the backend PCB. The heat can then be dissipated to the holding structure via the thermal interface [48].

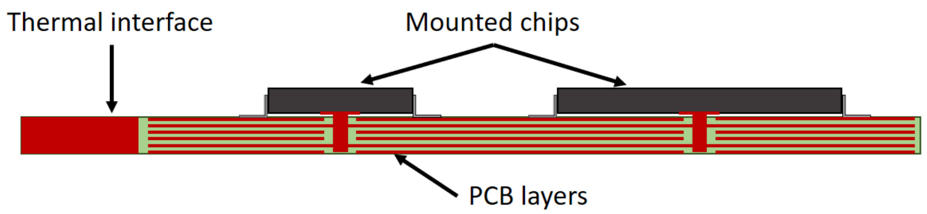


Figure 4.6: Thermal concept of the DIF backend PCB

4.2.3.2 Data Interface Frontend

The DIF frontend PCB has an optical frontend integrated on the front side, on the backside the frontend PCB provides an interface for the connection to the backend via the connecting cable. The DIF frontend converts the data stream from the backend to optical signals and transmits the data via an optical link, vice versa, it receives optical signals from the respective opposite frontend of the adjacent module, converts them back to electrical signals and sends the data to the DIF backend.

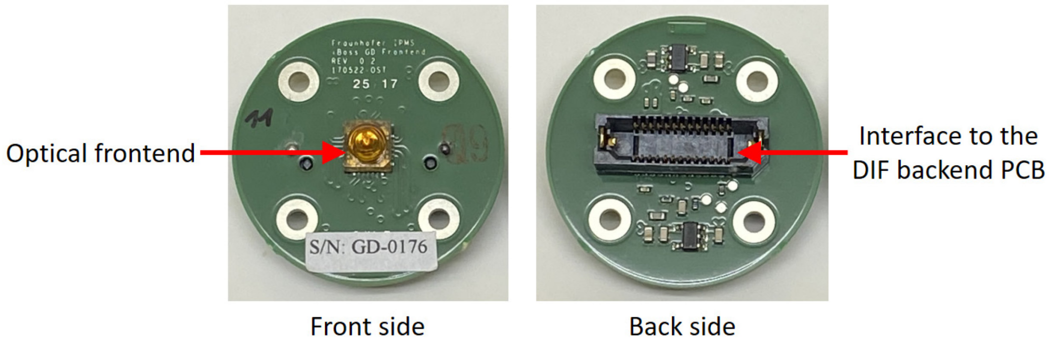


Figure 4.7: Front side and back side of the DIF frontend PCB

According to the requirements defined in Section 4.2.2, the integrated electronics was chosen for an application in space environment as well as a new connector type which should withstand the qualification process, the frontend shall have no electronics integrated on the front side and on the back side, in a circle of 5 mm around the center due to the shielding concept of the data interface.

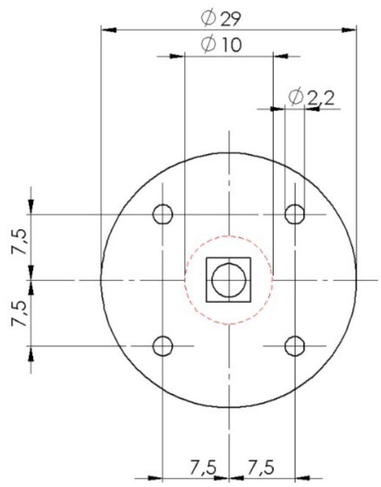


Figure 4.8: Dimensions of the DIF frontend [48]

4.2.4 Functional tests of the Data Interface

Several functional tests of data interface have been run to assess the performance of the data interface. For the tests of the data interface a test rig is designed and built. The design guidelines for test rig are defined regarding to several aspects. Especially the functionality of the data interface for the case of alignment and distortion need to be assured.

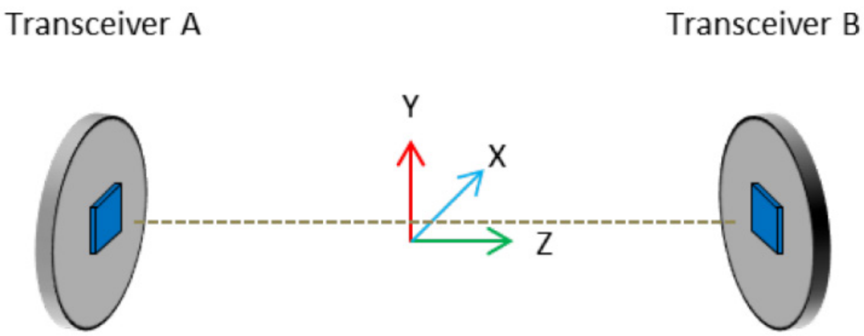


Figure 4.9: Coordinate system of the test setup for the data interface [48]

The data interface tolerates an alignment of 1mm in each direction of the X-Y-plane in the Figure 4.9 [48]. The test rig comprises of three moveable parts, the railguide and two mounting brackets, one fixed and the other moveable along the railguide to enable measurements in several distances between two data interfaces. The mounting brackets have four slotted holes where the frontend can be mounted and moved 2mm in positive or negative X and Y direction.

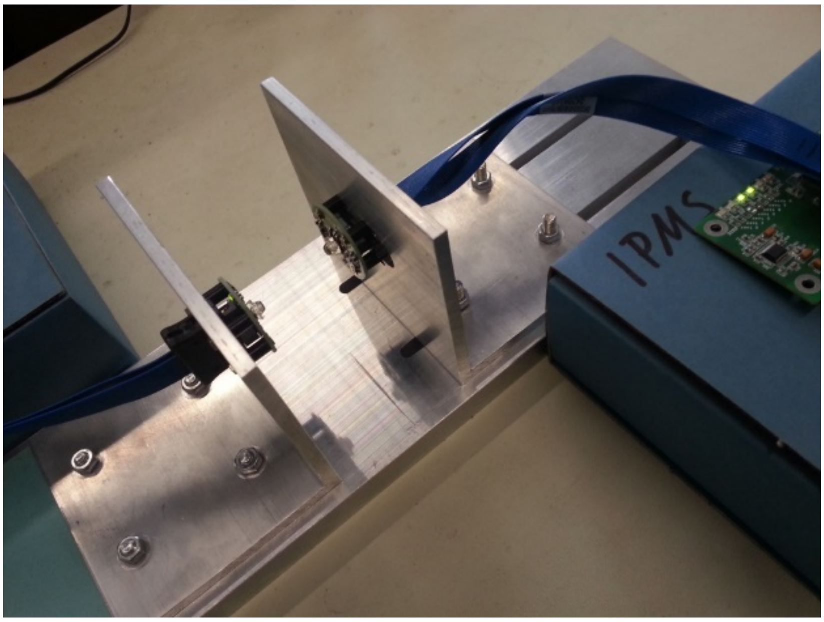


Figure 4.10: Test setup of the data interface [48]

The data interface meets the requirements of rotation symmetry. The data rate stayed constant for any rotation of the DIF frontend. The result of the test of rotation symmetry of the data interface is shown in Figure 4.11.

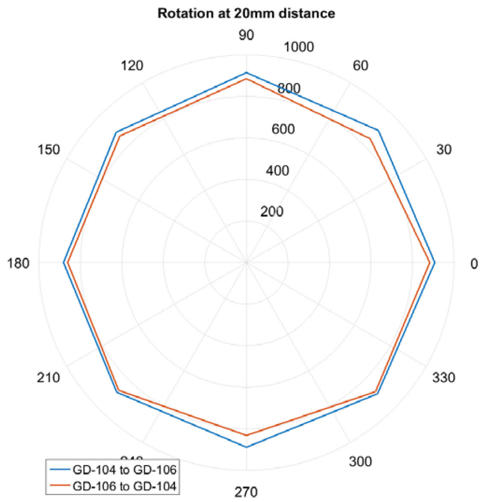


Figure 4.11: Measured data rate of rotated data interface as the proof for rotation symmetry [48]

The frontends of a coupled pair of data interfaces tolerate an alignment of up to 2mm in X- and Y-direction. No impact on the measured data rate of the data interface can be registered. The results of the alignment test is shown in Figure 4.12.

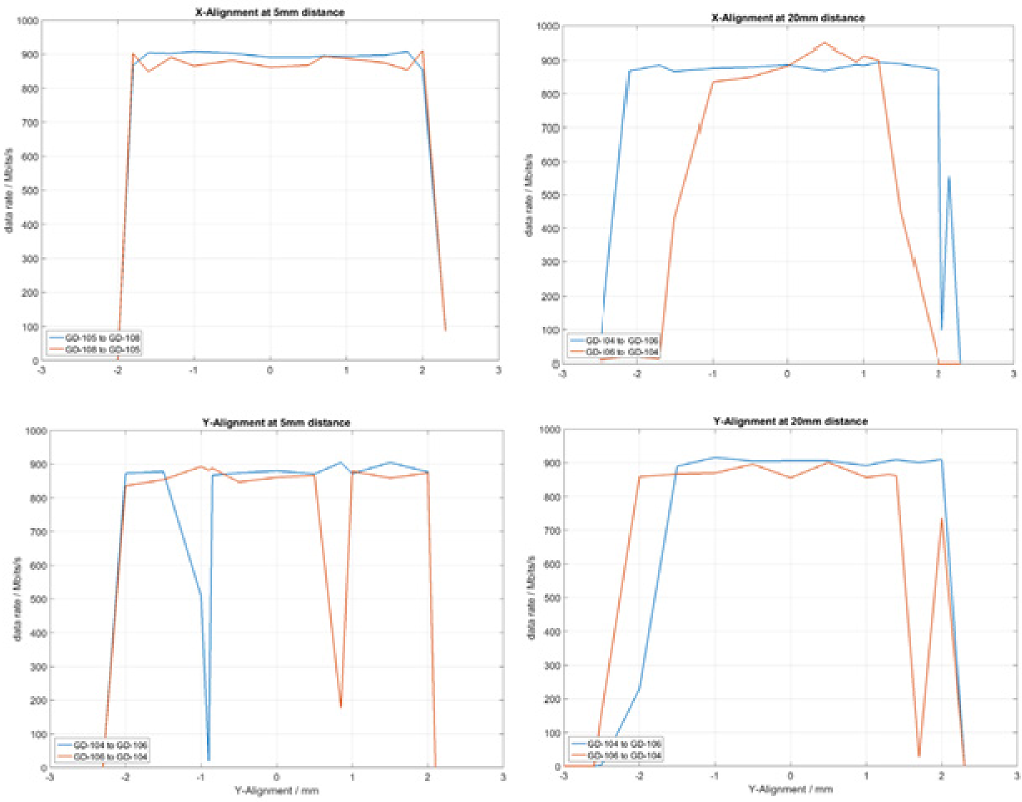


Figure 4.12: Measured data rate of the data interface by alignment in x- and y-direction [48]

The measured maximum distance between two DIFs for a fully functional data transmission is 70mm.

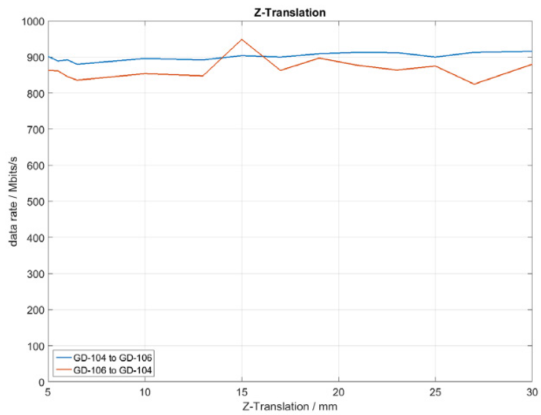


Figure 4.13 Measured data rate over z-translation [48]

4.2.5 Characteristics of the Data Interface

The measured maximum data rate of the optical data interface is 1000 Mbps. In order to compensate the mechanical inaccuracy of the mechanical coupling of the building blocks, the data interface tolerate a distortion angle of maximum 2° referred to the Z-axis.

A coupled pair of data interfaces can transfer data with a distance up to 70mm. Some other characteristics of the data interface are listed in Table 4.1.

Table 4.1: Characteristics of the DIF [48]

Maximum data rate	**1000 Mbps** **(at least 800 Mbps net data rate)**		
Communication Distance	**X-direction**	**Y- direction**	**Z- direction**
	max 1 mm	max 1 mm	5-70 mm
Distortion	max 2°		
Power Consumption	High rate mode (1000 Mbps)	Low rate mode (100 Mbps)	
	1.105 W	0.435 W	

4.2.6 Testing of the Data Interface toward radiation hardness

During the iBOSS project, several tests to qualify the optical data interface for an application in space environment have been conducted. These included the radiation test and the EMC test of the data interface. For the radiation test, following assumptions have been met [48]:

- In order to qualify the network components for the IOD mission under the most stressing, but realistic environmental influences, the orbit with respect to the considered environmental influence shall be used as mission defining constraint.
- The test shall be done for an application in space with an assumed on-orbit lifetime of 1 year.
- The orbit chosen for the worst-case calculations is 900 km with the inclination of 90°.

Due to the assumptions above, an irradiation of 3.096 krad (Si) was calculated. To regard the needed margin, a total irradiation in the test was set two times higher than the calculated value. A coupled pair of data interfaces was exposed to an irradiation of 6 krad (Si) in operation.

The test results of the radiation test of the data interface is shown in Figure 4.14. The data interface shows flawless data transmission throughout the entire test sequence at a very constant data rate, which is shown in the blue line in the figure.

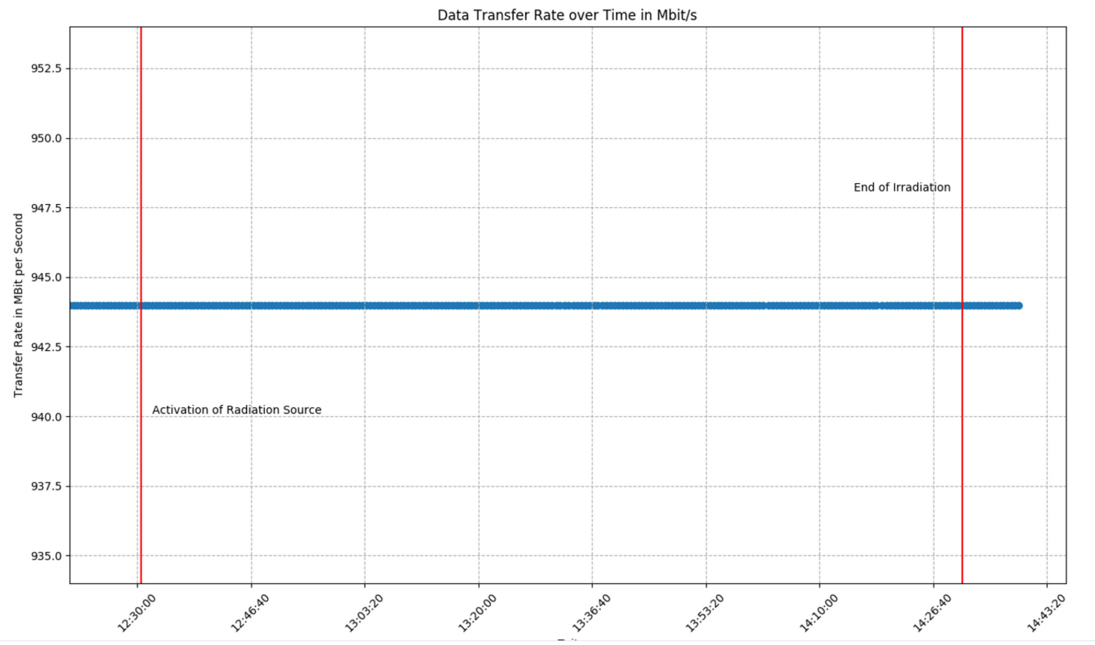


Figure 4.14: The radiation test results of the data interface [48]

4.3 The OBC-Switch-Board – A multifunctional component

The OBC-Switch-Board is one of the key components of the proposed on-board data communication system for highly modularized satellites. It represents a composition of an performant multi-processor unit as the OBC in a building block of the modular satellite and TTEthernet network components in order to span a TTEthernet network over all building blocks of the modular satellite. The component hosts the functionalities of an OBC, a TTE-Endsystem and a TTE-Switch which are integrated together on one board. The OBC-Switch-Board is designed and developed in cooperation of the company TTTECH in Wien, Austria, during the iBOSS project.

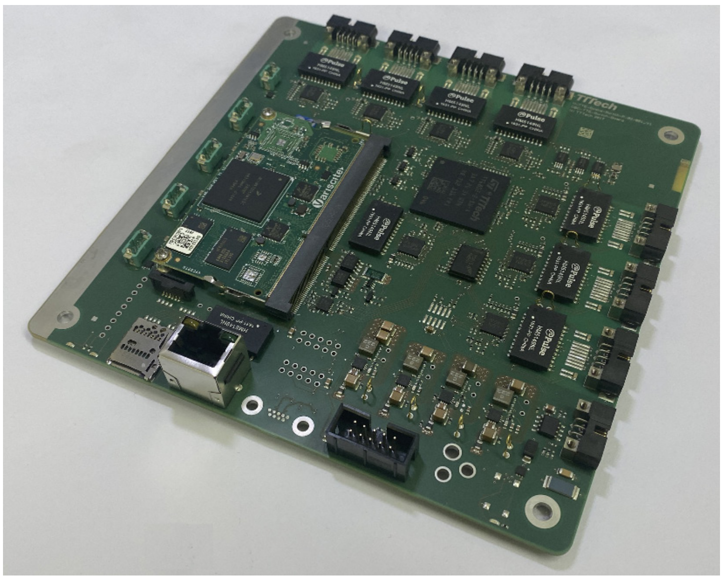


Figure 4.15: The OBC-Switch-Board

4.3.1 Hardware design of the OBC-Switch-Board

The main tasks of an OBC in a module of the modular satellite is monitoring and controlling the components integrated in this module and processing and communicating the data which are needed for the global control of the spacecraft with other OBCs. The OBC is integrated together with the TTE-Controller of TTTech on a single board to save harness and integration efforts. The TTE-Controller consists of the TTE-Switch and the TTE-Endsystem. The TTE-Endsystem represents an TTEthernet interface to the OBC and is connected to the TTE-Switch via an internal TTEthernet link. The TTE-Switch provides up to 12 TTEthernet ports, but only 7 ports are assembled on the OBC-Switch-Board. When integrated in an iBOSS module, the TTE-Switch can be connected to up to six iSSIs and so enables the module to be connectable to six neighbouring modules. The last TTE port can be used to connect to an optional payload component or other components with Ethernet interface.

Besides of the TTE interfaces the OBC-Switch-Board provides many serial interfaces like CAN, I^2C, SPI, UART, GPIO, A/D-Inputs as a broad variety of interfaces to be connected to other space components within the building block for peripheral controlling. Furthermore, interfaces like USB, SpaceWire, not assembled Ethernet interfaces, PCI and SPI exist as further options of the board. In the actual version of the OBC-Switch-Board these interfaces are not assembled as physical interfaces but can be realized on future versions of the OBC-Switch-Board if needed.

The TTE-Controller is a radiation hardened ASIC which contains the functionality of the TTE-Switch and TTE-Endsystem. A embedded Leon 2 CPU hosts the management functions of the TTE-Switch.

The TTE-Controller is connected to the OBC via a Quad SPI interface. During the iBOSS project, many COTS processors are considered and compared as the CPU of the OBC with respect to the given architecture, LINUX compatibility, computational power and power consumption. For the OBC-Switch-Board, ARM architecture is chosen due to the fact that the in iBOSS used software like ROS-2 is compatible to ARM architecture. The operating system of the OBC is LINUX since used middleware and software infrastructure in iBOSS are LINUX compatible. The OBC is a System-on Module (SOM) integration of the iMX7 of NXP® Semiconductors N.V., which is based on a silicon-on-insulator process. The SOM provides a dual core processor with a clock frequency of 1GHz and supports real-time applications. It is LINUX compatible and has a power consumption of around 5W [48].

The block diagram of the OBC-Switch-Board is shown in Figure 4.16.

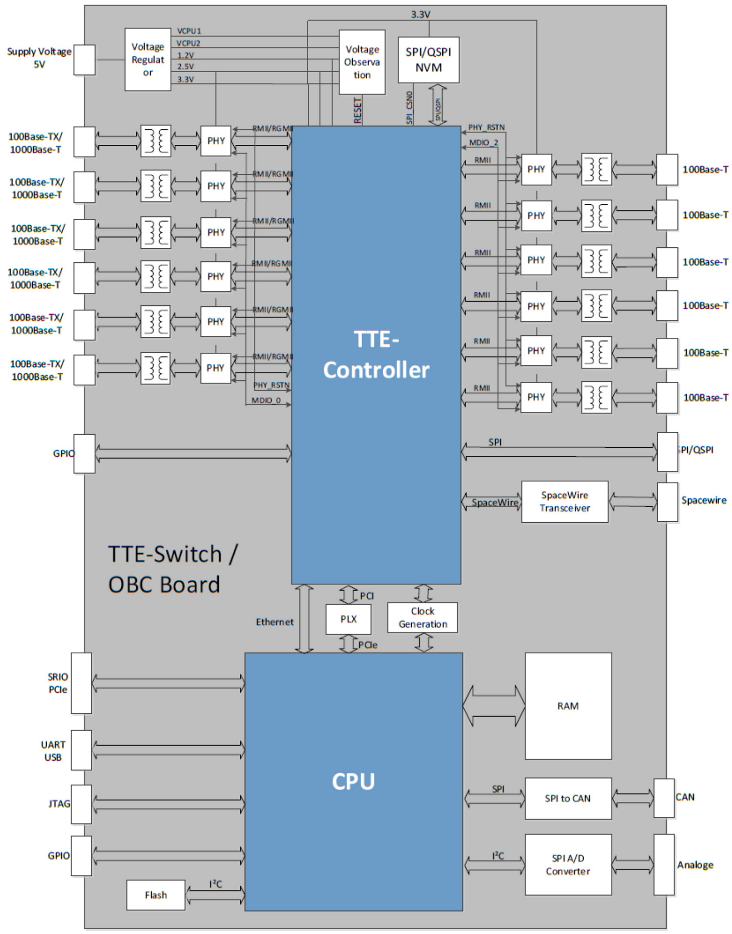


Figure 4.16: Block diagram of the OBC-Switch-Board [48]

The supply voltage of the OBC-Switch-Board is regulated voltage of 5VDC. The maximum power dissipation is around 12.5W.

Subsystem	Total	Unit
EXT_PHY	5222	mW
OBC	1296	mW
TTE_ASIC	4200	mW
Grand Total	10719	mW
		mW
With Conversion Losses	12542	mW

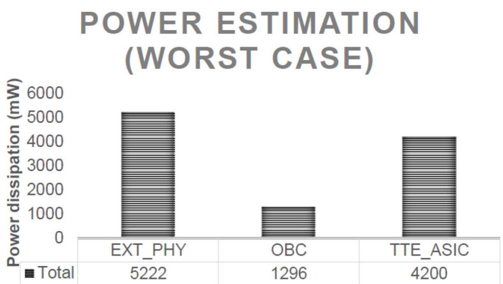


Figure 4.17:Maximum expected power dissipation of the OBC-Switch-Board [48]

The dimensions of the OBC-Switch-Boards are 15cm x 15cm x 2cm. Further specifications are listed in Table 4.2.

Table 4.2: Specifications of the OBC-Switch-Board [96]

Dimensions of the board (W x Lx H)	150mm x 150mm x 20 mm
Operating ambient temperature	-20°C to +70°C
Power supply	Nominal: 5VDC (±10%)

	Absolute maximum ratings: 6VDC Undervoltage lockout at 4.1VDC (with ±0.15V hysteresis)
Weight	155g

The components mounted on the board are illustrated in the Figure 4.18 and listed in Table 4.3.

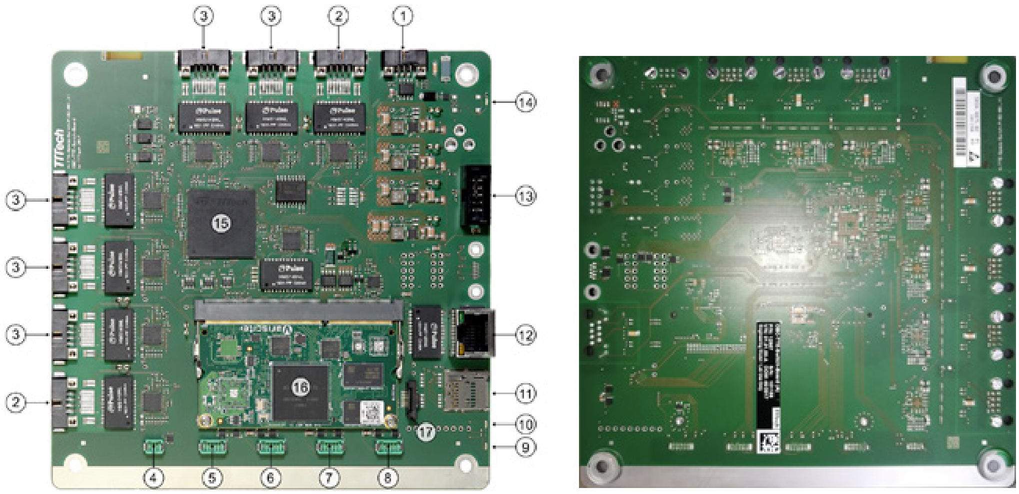


Figure 4.18: Bottom and upper side of the OBC-Switch-Board with references for the mounted components [48, 96]

Table 4.3: Components of the OBC-Switch-Board [48, 96]

1. Power/CAN port 10. User LED	10. User LED
2. 100 Mbps Ethernet port	11. microSD card slot
3. 1 Gbps Ethernet port	12. Debug Ethernet port (SOM)
4. ADC port 13. DSU port (TTE-Controller)	13. DSU port (TTE-Controller)
5. GPIO port 14. Power LED	14. Power LED
6. SPI port 15. TTE-Controller	15. TTE-Controller
7. I2C/UART port	16. System-On-Module (iMX7)
8. JTAG port (SOM)	17. Optional JTAG FFC bridge (SOM)
9. Power LED (SOM)	18. SOM – TTE Ethernet Link LED

4.3.2 Design aspects to enhance the dependability of the OBC-Switch-Board

The following design aspects are realised in order to enhance the dependability of the OBC-Switch-Board. The design layout of the OBC-Switch-Board provides separate power supply lines for the OBC and the TTE-Controller. The independent power supply of the components ensures the functionality of the TTE-Switch in case of a failure of the OBC.

Un that case, the routing functionality of the TTE-Switch is still given. Data packages routed over this module can still reach their destinations.

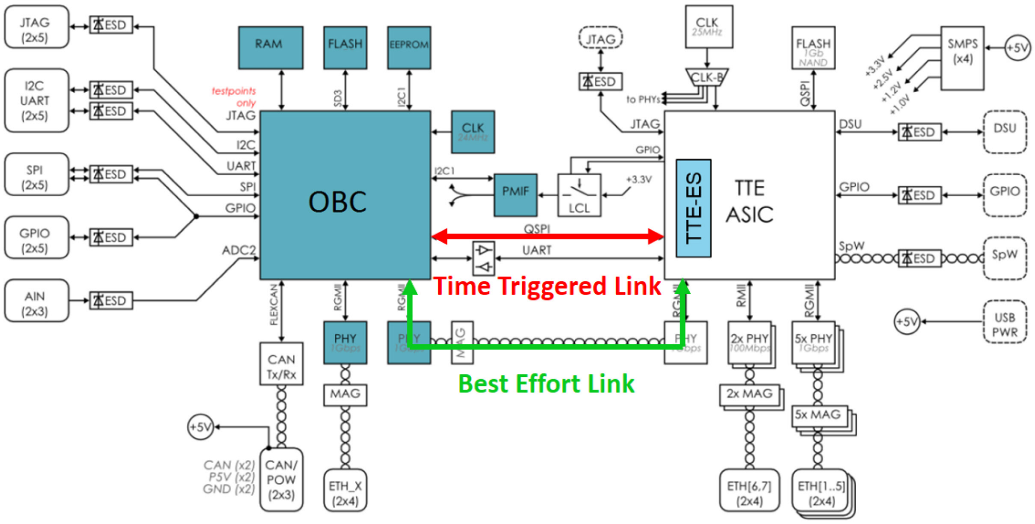


Figure 4.19: Electrical block diagram of the architecture of the OBC-Switch-Board [48, 62]

The electrical concept of the OBC-Switch-Board architecture is illustrated in Figure 4.19. To enhance the connectivity between the iMX7 and the TTE-Switch, the design of the OBC-Switch-Board provides an additional direct Ethernet links between the iMX7 and the TTE-Switch beside of the TTEthernet link via QSPI to the TTE-Endsystem hosted in the TTE-ASIC. The additional link is connected to the Ethernet interface of the iMX7 and allows only BE communication. An additional assembled RJ-45 Ethernet interface iMX7 allows the OBC to be connected easily the iMX7 to a common PC.

In summary, following functional and hardware design aspects are realized on the OBC-Switch-Board in order to enhance the reliability of the board:

- Separate power supply lines of the OBC and the TTE-Controller
- Overcurrent limitation for the OBC in case of latch up
- Additional Ethernet link between the TTE-Switch and the OBC
- Radiation hardened TTE-ASIC
- Used connectors for the TTEthernet interfaces are screwable Datamate J-Tek connectors by Harwin
- The jackscrew threads on the connectors are connected to signal shield on the OBC-Switch-Board and enable the termination of shielded cables.

The grounding concept of the OBC-Switch-Board is illustrated in the Figure 4.20.

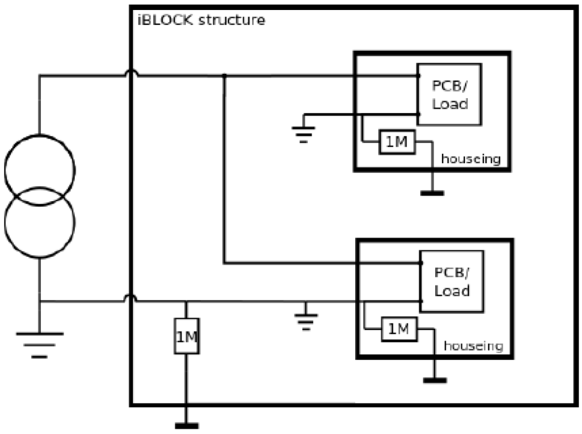


Figure 4.20: Grounding concept of the OBC-Switch-Board [48]

The design of the OBC-Switch-Board also regards a concept for heat dissipation similar to the DIF. In the middle of the PCB, two additional copper planes increase conductive heat transfer from the TTE-ASIC as well as from the OBC to a heat dissipation interface on an edge of the board. The heat exchange is improved by the sandwich structure of the layers.

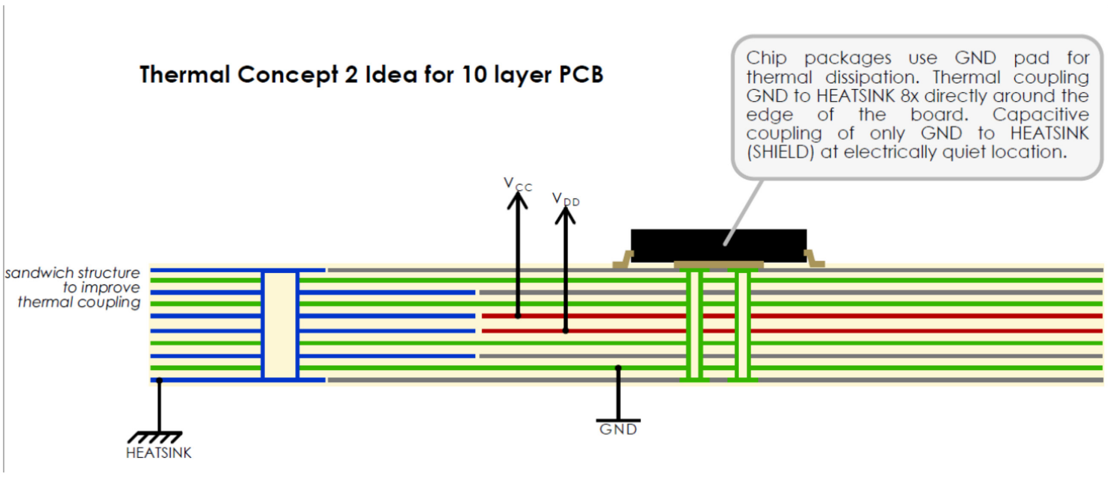


Figure 4.21: Thermal concept of the OBC-Switch-Board [48]

The development test concept of the OBC-Switch-Board comprises the Smoke test and several functional tests such as Ethernet loopback, switching, time-triggered data traffic etc.

4.4 Concluding remarks

In this chapter, design concepts for the network hardware are described. The network hardware concept comprises of the hardware design for optical data interface and for the OBC-Switch-Board. The described hardware devices were designed and developed in cooperation with IPMS (Dresden, Germany) and TTTECH (Wien, Austria) during the iBOSS project and shall be integrated in the planed iBOSS IOD satellite. In the hardware design concept, several aspects for an application in space are regarded with respect to the fault prevention methods described in the previous chapter.

5 NETWORK CONCEPTS FOR THE ON-BOARD DATA COMMUNICATION OF HIGHLY MODULARIZED SATELLITES

This chapter deals with the methods and concepts in the engineering process of the data communication network for modular satellite systems. Section 5.1 starts with the necessary steps needed for scheduling the data communication and loop handling of the on-board data network on a modular satellite. In Section 5.2, methods to enhance the reliability of the communication network are conducted, herein, an analysis of different failure types in the network is presented and followed by concepts to detect the failures and to recover network functionality in case of failures. With the focus on the fact that the running systems of the modular satellite shall be influenced as little as possible, approaches for a seamless reconfiguration of the on-board data communication network of modular satellites for different OOS-scenarios are described in Section 5.3 and the chapter is concluded with the concluding remarks in Section 5.4.

5.1 Engineering of the reconfigurable data network

In this chapter, an approach for a reconfigurable data communication network based on the TTEthernet standard which can be deployed on modular satellite systems as the on-board global data bus is presented. As described in Chapter 3.4, the features and mechanisms provided by this communication standard make an application on modular satellites possible since the requirements for high performance in terms of reliability, determinism and real-time of aerospace applications can be satisfied by the time-triggered communication principles and on the same time, the needed high data rate for payload data traffic as well as the advantages of the well-known standard Ethernet are also included. Nevertheless, the modular structure design of such satellite imposes high claims on dynamics and flexibility of the network communication systems. The unification of the inherent on-orbit reconfigurability and the time triggered communication architecture, which is generally not known for flexibility and reconfigurability, represents a special challenge. In following, approaches and concepts to increase the flexibility of the time triggered architecture based on TTEthernet will be presented, which will allow the on-board data network to be reconfigurable on orbit. Finally, the applicability of the presented concepts is demonstrated exemplary by applying them to configure a reconfigurable on-board data communication network on a 2x2 modular satellite based on the iBOSS concept.

Figure 5.1 shows the block diagram of the 2x2 iBOSS modular satellite. Each module of the satellite shall provide an OBC and a TTE-Switch. Mission payload components can be connected to the TTE-Switch if they have an Ethernet interface or to the OBCs using other interface standards like CAN or SpaceWire. The network communication control scheme is multimaster. All OBCs have the same priority. For the internal component bus of the modules, a CAN bus is used additionally. The TTE-Switches can operate independently of the OBCs and the TTEthernet network functionalities are still guaranteed even when a malfunction of an OBC in a module occurs. The modules are coupled to the neighbours via the multifunctional interfaces iSSIs. The optical data interfaces integrated in the iSSI convert the electrical signals of the global Ethernet bus into optical signals and vice versa; and thus, galvanic isolation of the intermodular data transmission is given.

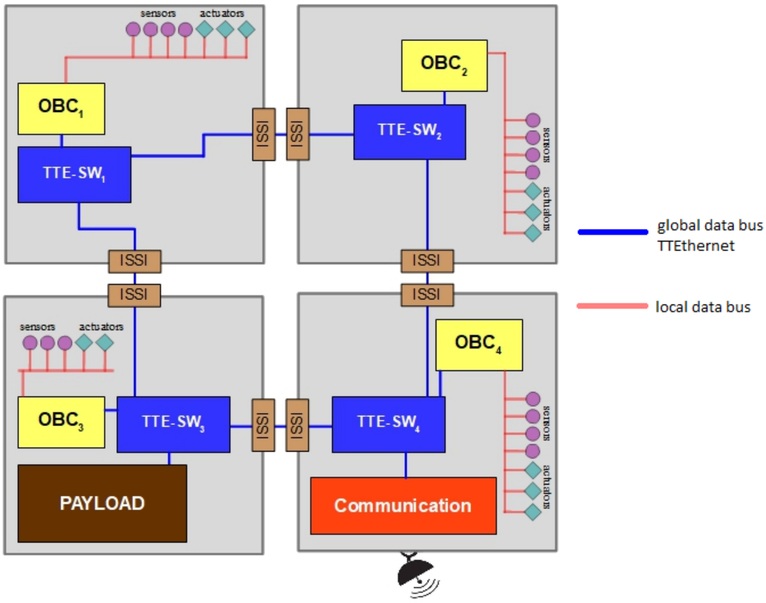


Figure 5.1: Block diagram of a small 2x2 modular satellite based on the iBOSS concept [7–9]

5.1.1 Scheduling of the on-board data communication

Usually, for space missions, the details of the spacecraft design are defined completely at the latest in Phase C of the project life cycle [97]. With the knowledge of the system designs including all on-board system hardware and software, the data exchange between applications can be scheduled accurately. The first step of the scheduling process of the data communication network on-board the modular satellite begins with the break-down of the given system designs to input parameters for the physical network model and the logical network model. The physical and logical model of a data communication network will be needed as inputs for scheduling of the on-board data communication. In this section, the scheduling approach will be explained first on conceptional level. After that, the proposed concept will be applied to the exemplary 2x2 modular iBOSS satellite. For simplicity, only time-triggered data traffic and best-effort traffic will be considered in following. Therefore, additional attributes of the rate-constrained data traffic like bandwidth allocation gap (BAG), jitters, etc. will be disregarded.

5.1.1.1 *Physical and logical network model*

With the given system designs, the details of the network hardware and software are known. The physical network model defines the physical topology and other parameters of the on-board data communication network. The following information are comprised in the physical network model:

- **Node number**: The number of the nodes in the network (switches and OBCs) is given by

$$n \in \mathbb{N}$$

- **Connectivity**: The symmetric connectivity matrix $\boldsymbol{C}$ represents the physical topology respectively the physical connections between the nodes.

$$\boldsymbol{C} = \begin{bmatrix} c_{1,1} & \cdots & c_{1,n} \\ \vdots & \ddots & \vdots \\ c_{n,1} & \cdots & c_{n,n} \end{bmatrix} \in \{0,1\}^{n \times n}$$

- **Transmission speed**: The data transmission speed between two neighbouring nodes is given in

$$\boldsymbol{R} = \begin{bmatrix} r_{1,1} & \cdots & r_{1,n} \\ \vdots & \ddots & \vdots \\ r_{n,1} & \cdots & r_{n,n} \end{bmatrix} \in \mathbb{Z}^{n \times n}$$

  Since the transmission time between 2 components can be regarded as the same for both directions for simplicity, the array $\boldsymbol{R}$ is symmetric: $r_{i,k} = r_{k,i}$; and for $c_{i,k} = 0$, there is no direct connection between node i and node k and the transmission speed between i and k is set as $r_{i,k} = r_{k,i} = -1$.

The logical network model contains the following parameters:

- **Transmission job**: A transmission job describes the time-triggered or best-effort transmission of data from one OBC to another. The number of the transmission job is defined by

$$j \in \mathbb{N}$$

- **Sender nodes**: The sender nodes of the transmission jobs are denoted in

$$\vec{s} = \begin{pmatrix} s_1 \\ \vdots \\ s_j \end{pmatrix} \in \{1, \dots, n\}^j$$

- **Destination nodes**: The Boolean array $\boldsymbol{D}$ denotes the receiver nodes for every data transmission job

$$\boldsymbol{D} = \begin{bmatrix} d_{1,1} & \cdots & d_{1,n} \\ \vdots & \ddots & \vdots \\ d_{j,1} & \cdots & d_{j,n} \end{bmatrix} \in \{0,1\}^{j \times n}$$

- **Maximum frame size**: The maximum allowed size of the transmitted TTEthernet frames in each transmission job is described in:

$$\vec{f} = \begin{pmatrix} f_1 \\ \vdots \\ f_j \end{pmatrix} \in \mathbb{N}^j$$

- **Job periods**: The period of a time-triggered job is defined as the time duration between two consecutive transmissions of the same job. For non-periodic jobs (best-effort data traffic), the value is set to -1.

$$\vec{t} = \begin{pmatrix} t_1 \\ \vdots \\ t_j \end{pmatrix} \in \mathbb{Z}^j$$

- **Job type**: To distinguish between the time-triggered data traffic and event-triggered data traffic, the communication type of a transmission job is denoted in the vector $\vec{m}$. $m_i = 1$ implies the critical traffic of job i, in this case job i is time-triggered, and a best-effort transmission of job k is described by $m_k = 0$. In TTEthernet, the term **Virtual Link** (VL) is used for a time-triggered communication job.

$$\vec{m} = \begin{pmatrix} m_1 \\ \vdots \\ m_j \end{pmatrix} \in \{0,1\}^j$$

The logical and physical network model are provided as inputs in a scheduling algorithm which calculates the schedule for the on-board data communication network, as shown in Figure 5.2.

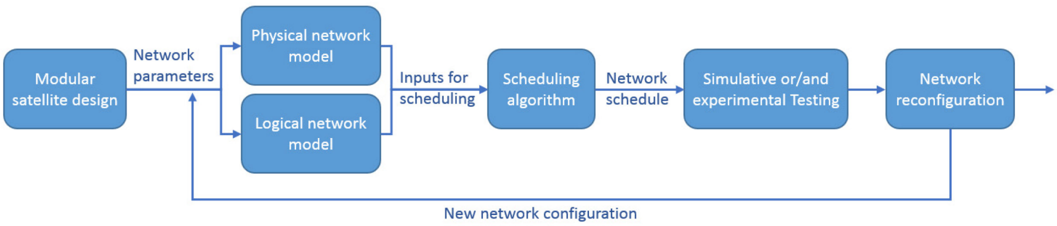


Figure 5.2: Scheduling process of a reconfigurable modular satellite

5.1.1.2 Decision variables and constraints of the network schedule

For assessing the quality of the schedule, some decision variables are defined in following.

- **Hops count**: During a transmission job, the transmitted data packet needs to be transported over at least one switch to the destination OBC. The number of the hops needed for all jobs is given by $\vec{h}$, whereby the maximum number of hops can be the number of the switches in the network SW_{max} plus one. In the case of a communication job having more than one receiver, the maximum number of hops of all paths is used. For example, if a job $i \epsilon \{1, \dots j\}$ may consist of the transmission of data from OBC_a to OBC_b and to OBC_c and the needed number of hops from OBC$_a$ to OBC$_b$ is $h_{i,1} = 3$ and from OBC$_a$ to OBC$_c$ is $h_{i,2} = 6$, then the i^{th} vector element is $h_i = max\{h_{i,1}, h_{i,2}\} = 6$.

$$h_{max} = SW_{max} + 1$$

$$\vec{h} = \begin{pmatrix} h_1 \\ \vdots \\ h_j \end{pmatrix} \in \{3, \dots, h_{max}\}^j$$

On a modular satellite, where every module provides an OBC connected over the switch to the neighbouring modules, the minimum number of hops is 3.

- **Job Path**: The array $\boldsymbol{P}$ refers to the path through the network of every job. Hereby, a path includes the sender and destination OBCs and the number of all switches on the route where the data packets of the job are transported along. Like the hops count vector, the longest one of all paths for a job is regarded. The index p_{max} can be calculated with:

$$p_{max} = h_{max} + 1$$

$$\boldsymbol{P} = \begin{bmatrix} p_{1,1} & \cdots & p_{1,p_{max}} \\ \vdots & \ddots & \vdots \\ p_{j,1} & \cdots & p_{j,p_{max}} \end{bmatrix} \in \{1, \dots, n\}^{j \times p_{max}}$$

For example, $p_{i,k} = a$ means that the k^{th} node of the longest path of job i is the node a in the network with $a \in \{1, \dots, n\}$.

- **Latency time**: A further criterion for measure the quality of the schedule is the latency time of a job. The latency time is the total duration of the transmission of a data packet from sending it to it arrival on the receiver. Similar to the two variables above, the duration of the longest path of the job is considered.

$$\vec{l} = \begin{pmatrix} l_1 \\ \vdots \\ l_j \end{pmatrix} \in \mathbb{N}^j$$

Using of additional constraints for the network communication can limit the solution space of the scheduling algorithm. Some exemplary constraints regarding the network reliability are:

- **Connectivity Constraint**: If there is no direct connection between two nodes, then no job path in the array $\boldsymbol{P}$ can contain the hop between these two nodes. For two arbitrary nodes a and b in the network, this constraint can be formulated as follow:

$\forall i \in \{1, \dots, j\}$, $\forall k \in \{1, \dots, p_{max}\}$, $\forall a \in \{1, \dots, n\}$, $\forall b \in \{1, \dots, n\}$ and $a \neq b$:

$$c_{a,b} = 0 \Rightarrow \ (p_{i,k} = a) \wedge ((p_{i,k-1} \neq b) \wedge (p_{i,k+1} \neq b))$$

- **Loop Avoidance Constraint**: Every node shall be visited only one time for every job to ensure the transmission time of the job message being minimized:

$$\forall i \in \{1, \dots, j\}, \forall k_1 \in \{1, \dots, p_{max}\}, \forall k_2 \in \{1, \dots, p_{max}\} \text{ und } k_1 \neq k_2:$$

$$p_{i,k_1} \neq p_{i,k_2}$$

Some other usable constraints types supported by the network scheduling tool ^{TTE}Plan of the company TTTech are listed below [98]:

- ***StartTimeConstraint*** – With this constraint, start time for transmission of time-triggered data from a sender can be defined or for receivers. If applied to receivers, the reception will start after the defined time.
- ***EndTimeConstraint*** – This constraint defines an absolute time representing the deadline of a transmission job. The frames of the job will be received before the defined time point.
- ***E2ELatencyConstraint*** – The end-to-end latency of a job can be limited in a time interval by applying this constraint.
- ***PrecedenceContraint*** – In order to regard the dependencies of jobs, this constrain can be applied to control the order of the transmission and reception of the messages of certain jobs.
- ***FullRedundancyConstraint*** – This constraint can be applied to ensure the redundant transmission messages of a job over different independent data paths.
- ***RoutingConstraint*** – This constraint defines the exact routes for BE links instead of letting the scheduling algorithm calculate the routes based on optimal load balancing.
- ***BlacklistRoutingConstraint*** – Using this constraint, certain device ports can be excluded from the routing solution space.

More information about the mentioned constraint types and some further constraints are contained in [98].

5.1.1.3 Network configuration using the tool package ^{TTE}Tools of TTTECH

For a TTEthernet network with TTEthernet hardware, the company TTTECH provides the tool package ^{TTE}Tools to configure the network. The tool package consists of a tool chain to support the network configuration step for step from specifying the input parameters of the network up to generating the binary images for the network hardware (TTE-Switches and TTE-Endsystems).

The tool package ^{TTE}Tools contains several tools for use in different step of the network configuration process and is built around open, layered XML databases. For comfortability, an Eclipse terminal-based GUI editor is provided. The software tool package comprises:

- ^{TTE}Plan: The TTEthernet network planning tool

- ^{TTE}Build – Network Configuration: The network configuration generation tool
- ^{TTE}Build – Device Configuration: The device configuration generation tool
- ^{TTE}Load: The loading software to load the binary images on TTE-Switches
- ^{TTE}Utilities: A software tool provides functionality to patch XML files.
- Eclipse terminal-based GUI editor

All the inputs of the network scheduling process can be defined in the *network description file* using the XML syntax.
In the *network description file*, following information can be specified [98]:

- Types and characteristics of the network components and the physical network topology
- The logical network topology implies the timing parameters, synchronisation configuration parameters, redundancy, transmission speed, network periods, fault-tolerance requirements
- Definition of time-triggered and rate-constrained virtual links and best-effort traffic flows
- Definition of constraints and other network parameters

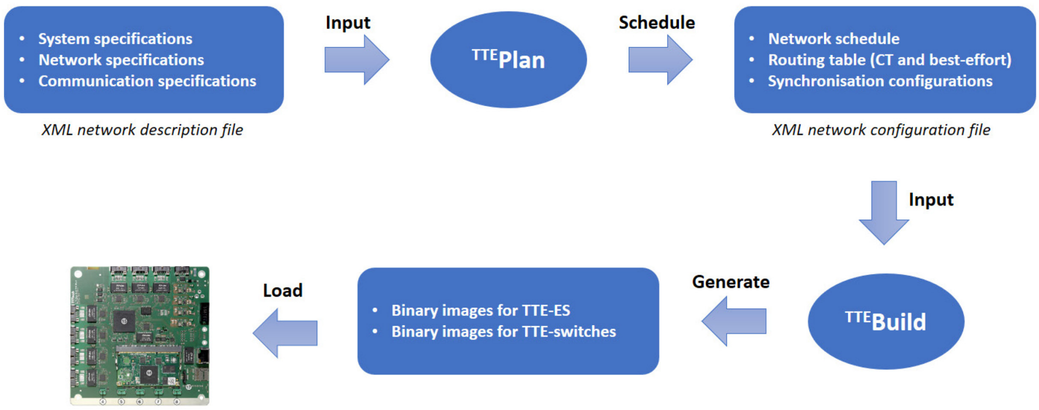


Figure 5.3: Overview of the network configuration process using ^{TTE}Tools

The next step is using the tool ^{TTE}Build to generate the XML *network configuration file*. ^{TTE}Build can be used to perform the following tasks:

- ^{TTE}Build Network Configuration uses a network configuration as input for generating device configuration files for the devices in the TTEthernet network.
- ^{TTE}Build Device Configuration converts XML-based device configuration files to binary images, which can then be downloaded to a switch via an Ethernet connection using ^{TTE}Load.

The binary image files can be loaded to the TTE-Switches using ^{TTE}Load or another suitable data loader, and to the TTE-Endsystems using the application.

5.1.1.4 *Network configuration for a 2x2 iBOSS modular satellite*

For better understanding of the network configuration process, the network configuration process described in the previous chapters is applied exemplarily on a 2x2 iBOSS modular satellite. The block diagram of such a satellite is shown in Figure 5.1. The satellite comprises of four modules and each module provides an OBC and a TTE-Switch. The TTE-Switches are connected to each other and form a ring. Thus, the on-board data network contains a network loop. Neglecting the impact of the electrical-optical-electrical conversion of the data interfaces, the resulted simplified physical network topology is illustrated in Figure 5.4.

The data network contains 8 network nodes comprising of 4 OBCs and 4 TTE-Switches. Assuming the following communication jobs:

1. Node ① sends a time-triggered periodic frame with a period of 10ms to node ④.
2. Node ④ sends a best-effort message to node ① and node ⑥.
3. Node ④ sends a time-triggered periodic frame every 5ms to node ① and node ⑧.
4. Node ⑥ sends a time-triggered periodic frame every 10ms to node ① and node ④ and node ⑧.
5. Node ⑧ sends a best-effort message to node ④.

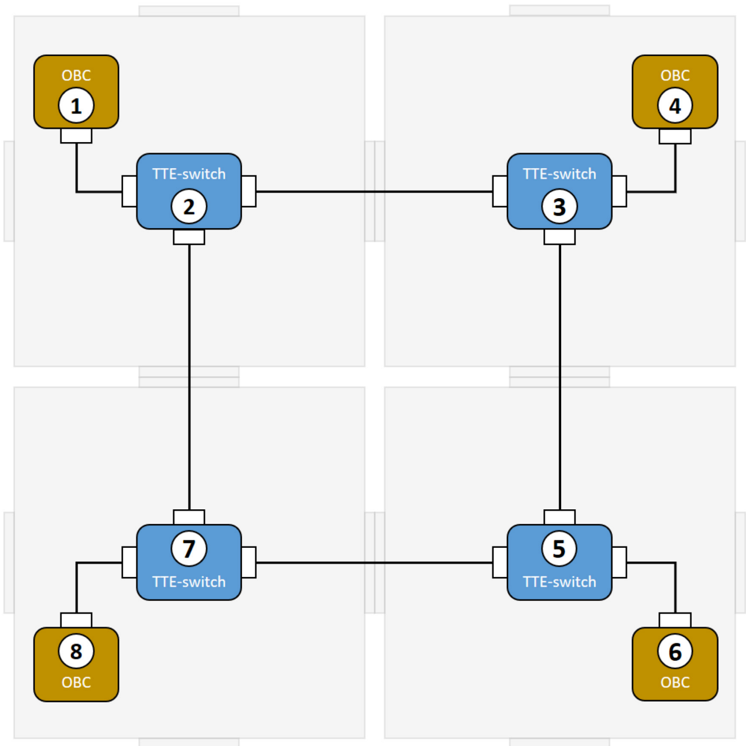


Figure 5.4: The physical network topology of the exemplary modular satellite

For simplification, the transmission speeds of all data links are assumed as constant with 100Mbps and the maximum size of the payload data in the frames of all transmission jobs is set constantly at 1500 bytes. For the most TTE-ES, the supported maximum payload size on the MAC layer ranges between 0 and 1500 bytes. The maximum frame size includes MAC header, MAC checksum and the payload and is 1518 bytes.

In TTEthernet, a data packet can be split into several fragments, and each fragment is packed into an Ethernet frame. The received fragments of a packet are then reassembled to the original packet by the receiver. If a maximum payload size of a message is set greater than the possible payload size of a frame, ^{TTE}Build automatically enables the feature defragmentation for the affected VLs. For event-triggered data traffic, defragmentation is not supported for best-effort traffic and only recommended for rate-constrained traffic. For time-triggered traffic, it makes no sense, since the application engineer should know that the full payload will arrive after some periods and handle this on the application level [98].

Table 5.1 depicts the input parameters of the scheduling algorithm comprising of the physical and logical network model based on the given network and communication specifications.

Table 5.1: Physical and logical model of the exemplary communication network

Input Parameters		Variables	Data
Physical network model	**Node number** (OBC and TTE-Switches)	n	8
	Connectivity	$\boldsymbol{C} \in \{0,1\}^{n \times n}$	$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$
	Transmission speed	$\boldsymbol{R} \in \mathbb{Z}^{n \times n}$ $[Mbps]$	$\begin{bmatrix} 0 & 100 & -1 & -1 & -1 & -1 & -1 & -1 \\ 100 & 0 & 100 & -1 & -1 & -1 & 100 & -1 \\ -1 & 100 & 0 & 100 & 100 & -1 & -1 & -1 \\ -1 & -1 & 100 & 0 & -1 & -1 & -1 & -1 \\ -1 & -1 & 100 & -1 & 0 & 100 & 100 & -1 \\ -1 & -1 & -1 & -1 & 100 & 0 & -1 & -1 \\ -1 & 100 & -1 & -1 & 100 & -1 & 0 & 100 \\ -1 & -1 & -1 & -1 & -1 & -1 & 100 & 0 \end{bmatrix}$

Logical network model	**Transmission jobs number**	$j \in \mathbb{N}$	5
	Sender nodes	$\vec{s} \in \{1, \dots, n\}^j$	$[1\ \ 4\ \ 4\ \ 6\ \ 8]^T$
	Destination nodes	$\boldsymbol{D} \in \{0,1\}^{j \times n}$	$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$
	Maximum frame size	$\vec{f} \in \mathbb{N}^j$ $[byte]$	$[1518\ \ 1518\ \ 1518\ \ 1518\ \ 1518]^T$
	Job periods	$\vec{t} \in \mathbb{Z}^j$ $[ms]$	$[10\ \ -1\ \ 5\ \ 10\ \ -1]^T$
	Job type	$\vec{m} \in \{0,1\}^j$	$[1\ \ 0\ \ 1\ \ 1\ \ 0]^T$

The communication schedule for the exemplary network of the 2x2 iBOSS modular satellite is calculated by using the scheduling module integrated in the tool ^{TTE}Plan. The results of the communication schedule are shown in Table 5.2 for the time-triggered data traffic. Best-effort data can be sent during two time-triggered slots.

Table 5.2: Scheduling results of the exemplary communication network for a 2x2 iBOSS modular satellite

Job	Sender	Receivers	Traffic type	Period	Path (longest)	Max Latency
1	①	④	Time-triggered	10ms	①→②→③→④	579µs
2	④	①,⑥	Best-effort	-	④→③→②→①	-
3	④	①,⑧	Time-triggered	5ms	④→③→②→⑦→⑧	620µs
4	⑥	①,④,⑧	Time-triggered	10ms	⑥→⑤→⑦→②→①	873µs
5	⑧	④	Best-effort	-	⑧→⑦→⑤→⑥	-

The whole network schedule is depicted in Figure 5.5. The maximum bandwidth utilization of the time-triggered data traffic on a physical link in the network is 3.6912% for the physical link between node ⑦ and ⑧ (for the direction TTE-Switch to OBC). The physical links with the most utilized bandwidth between two network switches are the links ③→② and ②→⑦ with both the same bandwidth utilization of 2.4608%.

The execution time of the scheduling process is negligible with 0.95s. The low bandwidth utilization and the short execution time of the scheduling process are due to the sparse data traffic in the exemplary network. In real modular satellites, the data communication between the network nodes is way more demanding regarding data traffic and bandwidth utilization.

The output of the ^{TTE}Tools are the binary image files for the network devices. To apply the scheduled network configuration, the binary files need to be loaded on the TTE-Switches and the TTE-Endsystems. For real mission, the networks configuration of the regarded modular satellite shall be precisely calculated, optimized, tested in the AIT phase. Before launching, the tested final network configuration shall be loaded and applied on all network hardware of the modular satellite.

For a network reconfiguration in orbit, the new network schedule needs to be transmitted via satellite uplink to the satellite and distributed to all modules. Alternatively, network schedules for predetermined configurations can be loaded and stored on the satellite before launching.

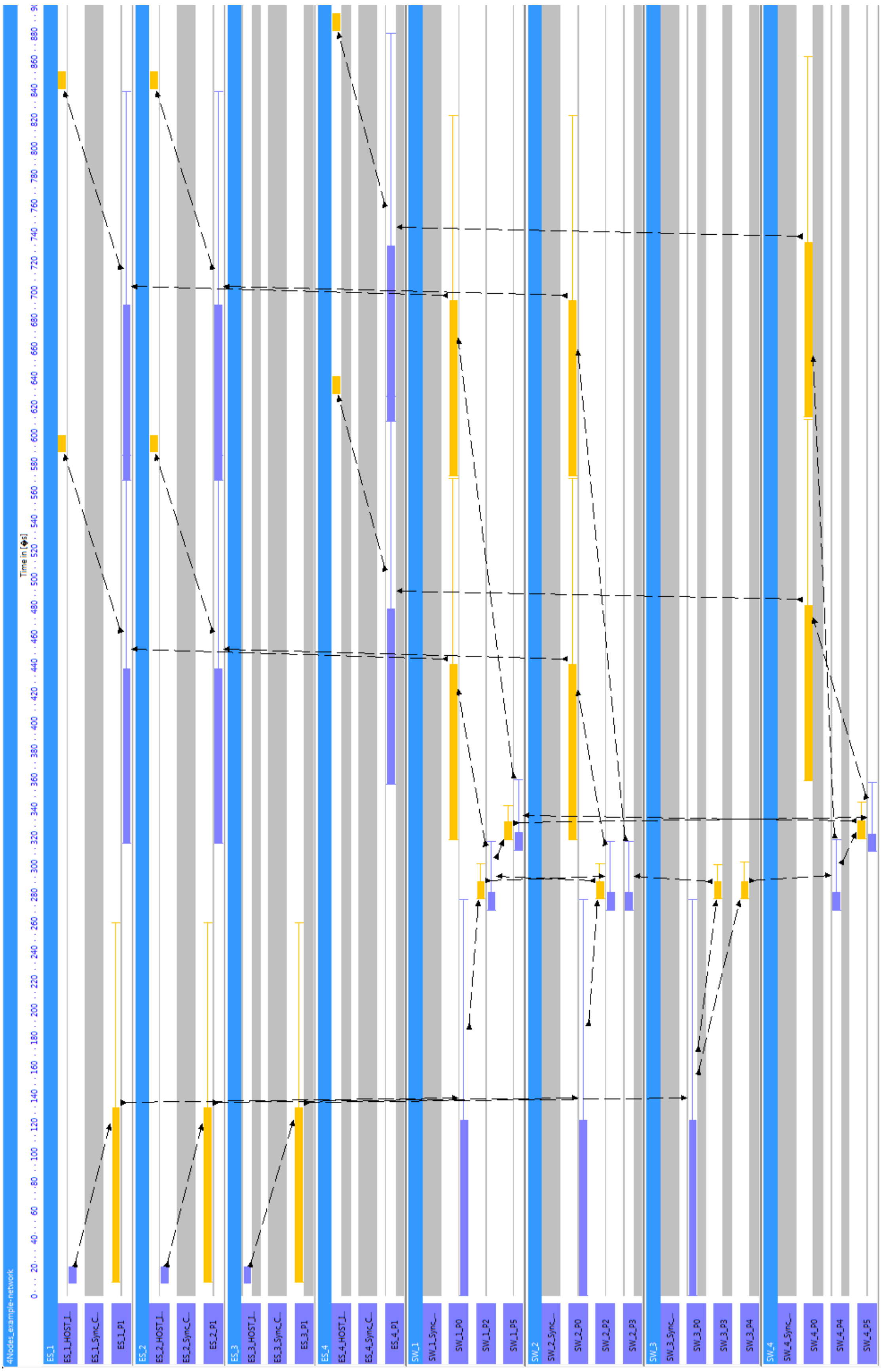


Figure 5.5: Communication schedule of the exemplary network on the 2x2 modular satellite

5.1.2 Loop-handling concept of the communication data network

On a modular satellite, where redundant data paths are given by multiple connection of the modules, it is highly likely that network loops can exist, and broadcast messages routed by the network switches can result in a broadcast storm. Frames could be transmitted in endless loops and overload the network.

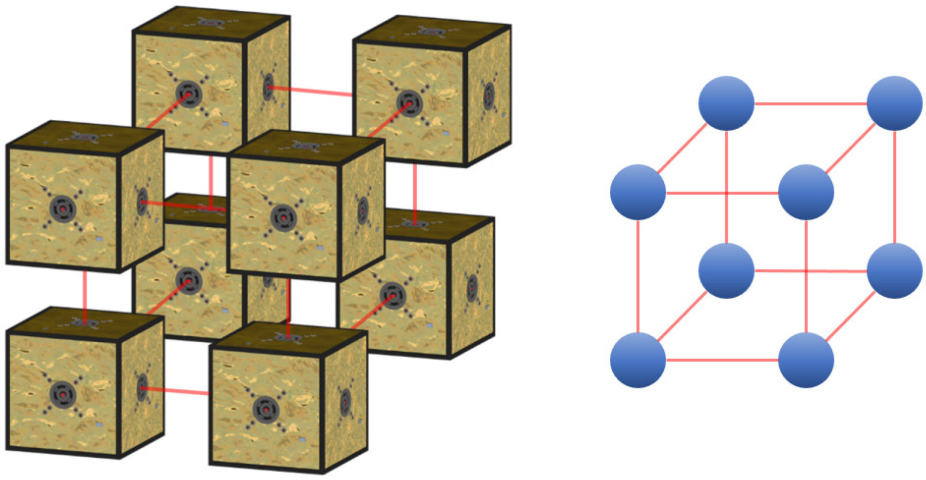


Figure 5.6: Network loops on a 2x2x2 iBOSS modular satellite configuration

In the figure above, multiple network loops exist in a 2x2x2 iBOSS modular satellite configuration. The on-board communication data network has a physical mesh topology. Although the network engineer can remove loops by manually switching off certain communication links between two modules and change the physical network topology into a tree or linear bus topology, but this will reduce the network reliability effectively by creating potential single points of failure in the network and the redundancy of data paths, which is regarded as one of the advantages of the modular structure. Thus, mechanisms to handle network loops on logical topology shall be preferred.

For time-triggered data communication, network loops are less critical since in TTEthernet, time-triggered and rate-constrained traffic belong to the Critical Traffic (CT) and are per design and definition statically configured. The CT communication paths in the network are predefined as CT virtual links within the device configuration and loaded into the devices (TTE-Endsystems and TTE-Switches) at startup. After startup, the critical communication traffic cannot be edited anymore and changing the network configuration inside a device will require that a new configuration file needs to be loaded into the device and a controller reset of the device.

For best-effort data traffic in a data communication network containing loops, TTEthernet provides the dynamic routing and address learning mechanism. With this mechanism the network switches can learn the network addresses of the nodes dynamically and update the routing table periodically. The best-effort routes throughout the network are then statically used to route the best-effort traffic though the network.

However, during startup, the dynamic routing table is empty, so default routes need to be set manually. This feature can be set by setting the following network configuration parameters in the network description file:

```
enableDynamicRouting="true"
createUnknownDefaultRoutes="false"
```

If the parameter createUnknownDefaultRoutes is set "true", the default routes in the switches will be set to all ports except the incoming one and incoming frames will be broadcasted to all other ports of the switch. In a network containing loops, this inevitably leads to a broadcast storm and will crash the network.

Using the mentioned setting of the parameters above, best-effort data traffic can be routed safely through a network containing loops without causing a broadcast storm.

5.2 Reliability enhancement of the on-board data communication system

According to the definition of dependability described in Chapter 2.4, reliability is one attribute of the system dependability and refers to the continuity of the correct system functions which can be disrupted by the occurrence of failures in the system. Failures are caused by errors which have the origin in faults within the system.

To ensure and increase the system dependability, 4 techniques are identified: fault prevention, fault tolerance, fault removal and fault forecasting. Fault forecasting is not regarded in this thesis. Aspects of fault prevention are provided by methods and techniques to enhance the robustness of the network hardware components and have been described in Chapter 4. The ability of fault removal in operation is given by the modular architecture of the satellite bus itself and the on-orbit serviceability of modular satellites.

A fault tolerant system shall still be able to provide its specified functionalities in presence of faults. Therefore, it must integrate mechanisms to detect failures and to recover the system functionalities affected by failures in acceptable manner in the communication protocols. For a data communication system on modular satellites, a failure of any network device including passive components such as cables or internal connections on the OBC-Switch-Board in one building block can impact the communication stability of this module and of the whole system. In order to detect failures in the network, it is important to know firstly which types of failures in the data communication network on modular satellites can occur and what are their impacts on the communication system and the subsystems of the satellite.

5.2.1 Analysis of failures

Fault tolerance is a system property that allows the system to continue operating in acceptable manner in presence of the failure of some of its components. Thus, the first step of designing a fault-tolerant communication architecture is the determination of the types of the failures which can occur within the on-board data communication system of a highly modularized satellite. Figure 5.7 shows the network devices and components inside of a module of such a modular satellite using the TTEthernet as the main architecture for the intermodular data communication and the network hardware presented in Chapter 4.

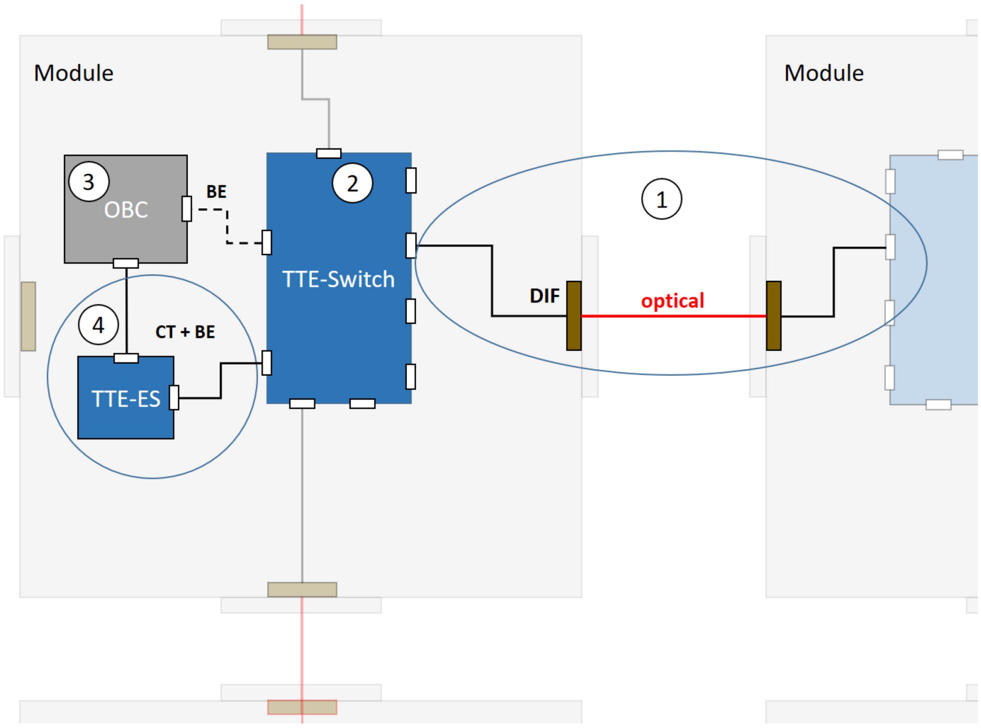


Figure 5.7: Network components and devices of a module regarded as funtional elements

In the figure above, some devices and components can be functionally grouped together since the functionality of a group is given if, and only if, all parts of this group are functional. For example, the optical data transmission using the optical data interface between two building blocks is only a substitution of a physical data bus, in this case, an Ethernet cable. Therefore, all components of the data connection from the port of the TTE-Switch in one module to the port of the TTE-Switch in the neighbouring module are grouped and regarded as one functional element. The failure of one part of this group, whether of the cable between the TTE-Switch and the backend of the data interface or of the backend of the data interface or of another part of the data interface, will result in a failure of this group. In Figure 5.7, this group of components are described as element ①.

The TTE-Switch is marked as element ②, the OBC as element ③ and the TTE-Endsystem is marked as element ④. The OBC and the TTE-Endsystem can theoretically be grouped together as one element since the operability of TTE-Endsystem is obsolete in case of an OBC failure. But according to the hardware concept of the OBC-Switch-

Board described in Chapter 4, an additional Ethernet link between the OBC and the switch is provided. Although only BE traffic can be transmitted via this link, it still represents a fall-back solution for reintegrating of the OBC in the network communication in case of a failure of the TTE-Endsystem or of the connection between the OBC and the TTE-Endsystem. Failures of the numbered elements can lead to limitations of the communication or to complete loss of connection to the module where the elements are allocated.

As mentioned in Chapter 2.4, failures of the data communication system can be categorized **link failure** and **node failure**. While a link failure results in a limitation or loss of some communication functionalities of the module and consequently in a reduced functionality of the module and the satellite, a node failure can result in the loss of the whole module. The loss of a module may have severe impacts on the system functionality, in worst-case it can lead to the failure of the mission or even to the loss of the satellite.

A failure of element ① refers to a **link failure** between two nodes of the network. Since the data network on a modular satellite can provide redundant data paths between two modules, the failure of a single link between two switches may not inevitably lead to a complete module failure but it may cause that some critical data traffic, which are statically scheduled to be transmitted via this link, cannot arrive at the destination anymore. Link failure in a module that has only one neighbour results in severe consequences because the module will be cut off from the network. In this thesis, such modules are called **hermit modules**. If a cluster of modules is connected via one single link to the rest of the network as shown in Figure 5.8b, a failure of this link is critical because a part of the network is then cut off from the rest of the system. Corresponding to hermit modules, clusters of building blocks with a single link to the rest of the system are called **hermit clusters**. Configurations with hermit nodes or hermit clusters are weak designs and must be avoided in the design phase of the modular satellites because they contain SPOFs in the architecture.

The TTE-Switches, represented as element ② in Figure 5.7, are the central elements for the intermodular communication and represent the entrance to the satellite communication network for intramodular devices. The failure of element ② is consequently catastrophic for every module and if critical data traffic to neighbouring modules are routed over the failed TTE-Switch, the failure also has impacts on the communication functionality of a group of nearby modules. In this case, the on-board data communication may be partly suspended and the failure of at least one module is determined. Thus, failure of the element ② must be assessed as a critical **node failure**.

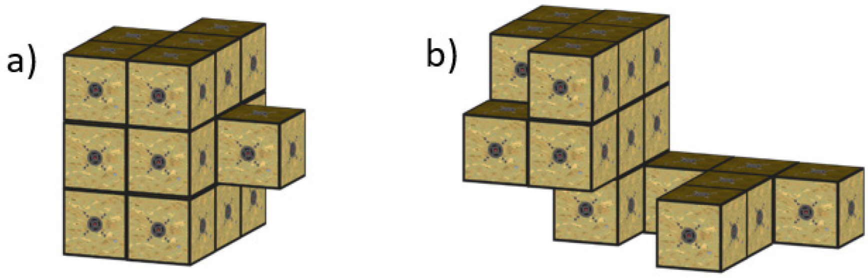


Figure 5.8: Exemplary modular satellite configurations with weak designs: a) A configuration with a hermit module; b) a configuration with a hermit cluster and several hermit modules

The failure of the element ③ occurs when the OBC of the module is out of function or cannot be contacted for some reasons. The communication with the controlling intelligence of the module is cut off. Functions and services provided by this module cannot be attained in acceptable quality anymore since the necessary coordination with the distributed system intelligence to accomplish the functions is missing. This also means that needed information generated by the defective OBC cannot be provided to other modules. Depending on the criticality of these data, this can have high impact on the functionalities of the system. However, if the functionality of the TTE-Switch is still given, data routing over this module is still ensured and other components in this module, which are directly connected to the TTE-Switch, can still be reached and controlled by other OBCs in the satellite. This scenario represents a less severe **node failure**.

The failure of element ④ means the disruption of the data traffic between the OBC and the TTE-Switch and therefore, the OBC is cut off from the data network. As mentioned in Chapter 4.3, the hardware concept of the OBC-Switch-Board provides an additional Ethernet link between the OBC and the switch ASIC. Although only best-effort communication can be transmitted on this link, this still represents a fall-back solution for the failure of the TTE-Endsystem. The mentioned link, represented by the dashed line between OBC and TTE-Switch in Figure 5.7, shall be deactivated on switch side per default and can be reactivated on demand. The failure of element ④ can be regarded as a **link failure** since this failure implies the loss of the TTEthernet data interface of the OBC and inherently the ability to send and receive critical messages.

Table 5.3 summarizes the classification of the failures of the elements and their impacts on the on-board data communication network.

Table 5.3: Categorization of failures of network elements in one module

Failure of element	Failure type	Criticality level	Impacts on the module reliability	Impact on satellite reliability
①	Link failure	average	- Limited communication to network	- physical topology changed

			- critical for hermit modules	- critical data traffic routed over failed link are not available - critical for networks with many hermit nodes or hermit clusters
②	Node failure	very high	- Module is cut off from communication network - Loss of intramodular communication on the Ethernet bus - Needed inputs for accomplish functions are not available	- Loss of time-triggered communication to one or more module - Loss of functionalities of one or more module - Network break-down possible
③	Node failure	average - high	- OBC failure means loss of controlling and monitoring the module - Loss of the main functionalities of module - Module is “brain dead”	- Loss of the module - Inputs for other modules to coordinate the satellite functions are not available - Data routing over the TTE-Switch in the module is still functional - Access to components connected to the Ethernet bus from OBCs in adjacent modules possible
④	Link failure	average - high	- Loss of the ability to send and receive time-triggered and rate-constrained data - Critical data can only be transmitted and received via best-effort channel	- Reliability of the communication of critical data to the module reduced - Functionality and reliability of module reduced - Impact severity depends on the importance of inherent caused failure of components in the module

5.2.2 Detection of failures

To coordinate efficiently the error detection and error handling actions on a modular satellite, two OBCs of the modular satellite shall be predefined as masters of the communication reliability monitoring and recovery process and henceforth named as **health master 1** and **health master 2**.

In order to detect a failure in an operating on-board data communication system of the modular satellite, a periodic routine is implemented on the OBCs of every building block of the modular satellite except the two health masters. During operation, every OBC executes the routine of the begin of a defined duration cycle to check the connection health of the active data links to neighbouring building blocks and so, is able to monitor steadily the communication stability with the neighbours.

If an OBC detects a failure of communication to one neighbour, it will report to the health master. For doing this, firstly, every building block needs to know all neighbouring building blocks, which have an active data connection to it. Obviously, it is assumed that the configuration of any modular satellite shall be known at any time after LEOP. For the case of the reconfiguration of the building blocks on orbit, the final configuration of the modular satellite shall be known. Thus, physical and logical network topology of any satellite configuration are defined statically and persistently by the communication schedule in any mission phase.

Following assumptions are made for the error detection process:

- In order to achieve a 1-error-tolerance of the communication system, it is presumed that only one failure can occur at same time on the modular satellite, either a link failure or a node failure.
- The configuration of the satellite shall have no hermit cluster. For example, a single-line configuration of the modular satellite, where all building blocks are ordered in one single line, is not allowed.
- The health masters shall not be allocated in two directly neighbouring building blocks.
- All building blocks shall know the neighbours with active data link to them all the time.
- After a reconfiguration process or a rescheduling process, the health master shall distribute the new physical and logical network topology to all building blocks of the satellite.
- In order to detect a TTE-Switch failure, it is necessary that all data links in the network need to be active and dynamic routing for BE traffic of the on-board data network is active.

The failure of a TTE-Switch in the network can have catastrophic consequences to the whole network system and the satellite. In the following example, the building block with the TTE-Switch failure is allocated in a central position of the modular satellite. In this case, the failure does not only result in a loss of the critical communication to this building block but even in a collapse of the critical data communication between several building blocks nearby. In this case, some critical functionalities of the satellite may not be guaranteed anymore and an exchange of the building block in an OOS mission must be planed and accomplished within the shortest possible time. Figure 5.9 illustrates such a scenario.

For all VLs (Virtual Links – the logical links for time-critical traffic (TT and RC traffic)), the routes are defined statically by the schedule. Usually, not all physical links in the physical topology need to be used for routing the CT data packets in the network schedule. Possible BE physical links are represented by the dashed yellow lines between the building blocks and the green lines in the picture represent scheduled data paths of CT and BE traffic. The left picture in the figure below shows a possible schedule of the data traffic in the network.

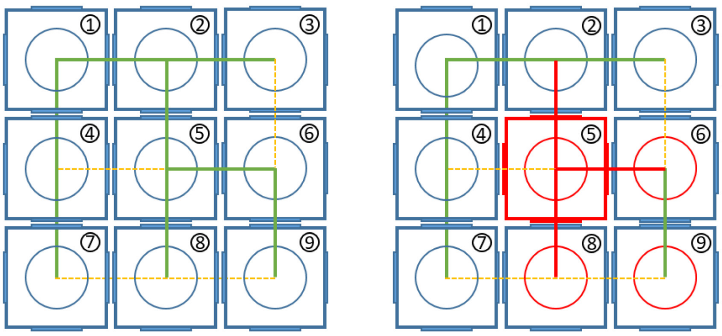


Figure 5.9: The critical TTE-Switch failure in the on-board data network of a modular satellite

In the example shown in the figure above, the TTE-Switch failure occurs in the central building block ⑤ and cuts off the communication of four building blocks from the other five building blocks of the satellite. In the cut off part of the satellite, only one data link of the CT is still functional. If one or more building block in this part contain ACS actuators like propulsion system thrusters or reaction wheels, a failure of the attitude control of the satellite is determined.

Therefore, it is necessary that all physical links are active and the dynamic routing for BE traffic is activated in the network. This ensures the report messages in case of a TTE-Switch failure in a central building block can arrive the health masters. If a BE route is affected by the TTE-Switch failure, the dynamic routing function will find another route throughout the network.

The health check routine implemented and executed on the OBCs utilizes the echo request and echo reply messages of the Internet Control Message Protocol (ICMP) to test the connectivity to the neighbouring OBCs. If no echo reply from one requested OBC is received, it is assumed that a failure occurred on the data link from the TTE-Switch of the requesting building block to the requested OBC (failure of ① or ② or ③ or ④ in Figure 5.7). By using the Simple Network Management Protocol (SNMP), the status of the TTE-Switch of the neighbouring building block can be requested. If this action also failed, it is assumed that the data link to the TTE-Switch of the neighbouring building block is corrupt or the TTE-Switch itself (failure of ① or ②). In this case, the checking OBC cannot determine the type and location of the failure and reports the corrupt link status sending the message connection_fail_status_1 to health master.

If the requested TTE-Switch answers to the SNMP request, then it can be concluded that either an OBC failure or a TTE-Endsystem failure or a failure of the connection between these components or to the TTE-Switch occured (failure of ③ or ④). This status is then sent to the health master in the message connection_fail_status_2.

The algorithm of the health check routine is outlined in Figure 5.10.

```
Algorithm 1 Health check routine in a building block
Load link neighbour list
while health check cycle do
    for all node i in link neighbour lists do
        if node i is health_master_1 then
            set master = health_master_2
        else
            set master = health_master_1
        end if
        send ICMP echo request to node i.OBC
        if no IMCP echo reply from node i.OBC received then
        send SNMP request to node i.switch
            if SNMP request to node i.switch failed then
                send message (connection_fail_status_1[node i]) to master
            else
                send message (connection_fail_status_2[node i]) to master
            end if
        end if
    end for
end while
```

Figure 5.10: Algorithm to check link health to neighbouring modules implemented on the OBCs of the modular satellite

In order to classify the type of the reported failure, another routine is implemented and running on the health masters. If the health master receives the status message about the occurrence of a failure on the data link from OBC A to OBC B, it will first check if building block B is a hermit module. In this case the failure can be directly denoted as node failure of B regardless of the type of the reported fail status because a hermit module can have only one neighbour and if the connection between the hermit module and its neighbour fails, then B is completely cut off from the communication network. The algorithm for the routine on the health master is outlined in Figure 5.11.

```
Algorithm 2  Failure detection and classification routine in the health masters
TRIGGER :   Receiving of a connection_fail_status message
Reporter = sender of connection_fail_status message
BB  = reported Building block
set BB health status as failure
if  BB is a hermit module then
    set BB failure type as node failure
else
    if  received message is connection_fail_status_1 then
        if  more than one connection_fail_status _1(BB) received then
            set BB failure type as node failure
        else
            set BB failure type as link failure
            set Reporter health status as failure
            set Reporter failure type as link failure
        end if
    else
        send command to BB.switch to turn on secondary OBC port
        send ICMP echo request to BB.OBC
        if  no IMCP echo reply from BB.OBC received then
            set BB failure type as node failure
        else
            set BB failure type as link failure
        end if
    end if
end if
```

Figure 5.11: Algorithm for failure detection and classification implemented in the health masters

If OBC B is not a hermit module, the health master needs to check if other reports about fail status of B are received. If the failure is reported by connection_fail_status_1(B) from only one neighbour of B, then a link failure between B and that neighbour (failure of element ① between B and the reporter) is diagnosed. If several messages connection_fail_status_1(B) are received, then a failure of the TTE-Switch (element ②) in building block B occurred and the health master denotes a node failure on B.

If the health master receives the message connection_fail_status_1(B) which indicates a failed connection to the OBC in B while the TTE-Switch in B still works. The health master will send the command to the TTE-Switch to activate the ethernet port for the secondary link to the OBC in B and check the health of the OBC in B via ICMP echo request after link recovery. If the echo request is not answered by the OBC in B, then an OBC failure (failure of element ③) and thus, a node failure in building block B is determined, otherwise an internal link failure (failure of element ④) in building block B is determined.

If a failure of a health master regardless of the failure type is confirmed, the second health master must define a new health master in the network and distribute this information to all OBCs.

5.2.3 Network recovery in case of failures

Error handling mechanisms are needed to recover the network system functionality in the occurrence of failure. In the last section different failures of network devices and components of the on-board data communication system on a modular satellite were identified and classified in two main types: link failure and node failure.

Due to Table 5.3, the impacts of a link failure are generally less critical than the impacts of a node failure on the network system. For the satellite itself, the criticality depends on which functions are allocated to the module with node failure or which critical traffics are scheduled to be routed over the broken link. For example, the failure of a link, via which many VLs with critical data needed for mission-critical functionalities are routed, is obviously more critical than a node failure in a module with no critical applications. Two kinds of link failure are distinguished, the failure of the link between two adjacent building block and the failure of the TTE-Endsystem within a building block.

5.2.3.1 Network recovery in case of a disruptive data link between two building blocks

A disruptive data link between two buildings block can be caused by a defect of any components on the physical connection between two TTE-Switches in two adjacent building blocks. From the point of view of the network communication, this refers to a permanent change in the physical network topology since the connection between two TTE-Switches are interrupted. In Figure 5.12, the recovery mechanism of the communication network for an exemplary 3x3 configuration of building blocks with a link failure is illustrated.

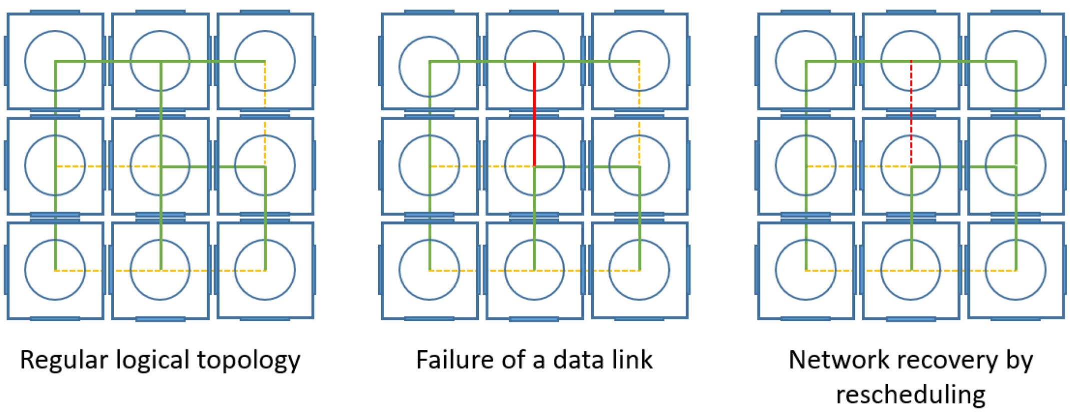


Figure 5.12: Recovery of the communication network with a link failure by rescheduling (Dashed lines – Physical links; full lines – scheduled BE and CT data paths; red line - defective link.

For each active data link between two building blocks in the network a recovery network schedule shall be calculated with the physical topology without this physical link as input. The recovery schedules are stored on the satellite. If a link failure between two modules occurs during the operational phase of the satellite, the health master will start the rescheduling process by loading the appropriate recovery schedule for this data link on all TTE-Switches of the network.

The new physical topology will be sent to the operator via downlink. New recovery schedules need to be calculated and uploaded on the satellite again via uplinks because the recovery schedules stored on the satellite become out-of-date by the resulting change of the physical topology.

This approach is generally applicable for all configurations of the modular satellite. For satellites with no hermit building blocks or hermit clusters as required above, it is even not necessary to store recovery schedules on the satellite because all TTE-Switches of the network are still reachable via redundant data paths. Therefore, the recovery schedule can be calculated by the operator and uploaded to the satellite via uplink in case of the link failure.

5.2.3.2 Network recovery in case of a link failure inside a building block

A link failure within a building block is detected when the TTE interface of the OBC in this building block is not able to send and receive time-triggered and rate-constrained data anymore (this occurs when the TTE-Endsystem failed or when the connection between the TTE-Endsystem and the OBC failed). However, the OBC can still be connected to the network via the secondary Ethernet link to the TTE-Switch. Depending on the criticality of the functions allocated in this building block, in some cases it may be necessary to exchange the building block by a new one with an OOS mission.

If the affected OBC does not host any time critical function and thus, has no need for critical data traffic, for example in a building block with a payload camera directly connected to the TTE-Switch, only the internal routing of the TTE-Switch in this building block needs to be reconfigured. All best-effort messages to and from the OBC will be routed to the secondary Ethernet port.

5.2.3.3 Network recovery in case of an OBC failure inside a building block

In case of an OBC failure inside a building block, the TTE-Switch is still functional. The data traffics routed by this TTE-Switch are not affected. Similar to the case above, the impact of the OBC failure on the satellite functionality depends on the functions hosted in the defective OBC.

If the loss of this OBC is bearable for the satellite, no further action must be taken except that the missing of the contributions of this OBC to the coordinative execution of some satellite functions must be regarded and maybe taken over by other OBCs on the satellite.

5.2.3.4 *Network recovery in case of an TTE-Switch failure inside a building block*

A failure of the TTE-Switch inside a building block of the modular satellite is the most critical failure type. Depending on the allocation of the building block and its functions, the impacts on the on-board data network can be catastrophic and can even lead to a loss of the satellite if no actions are taken. A failure of a TTE-Switch generally requires the exchange of the building block in shortest possible time as consequence. But until the OOS mission to change the defective building block is initiated, the data communication network of the modular satellite must be recovered in the best possible manner to ensure at least the survival of the satellite.

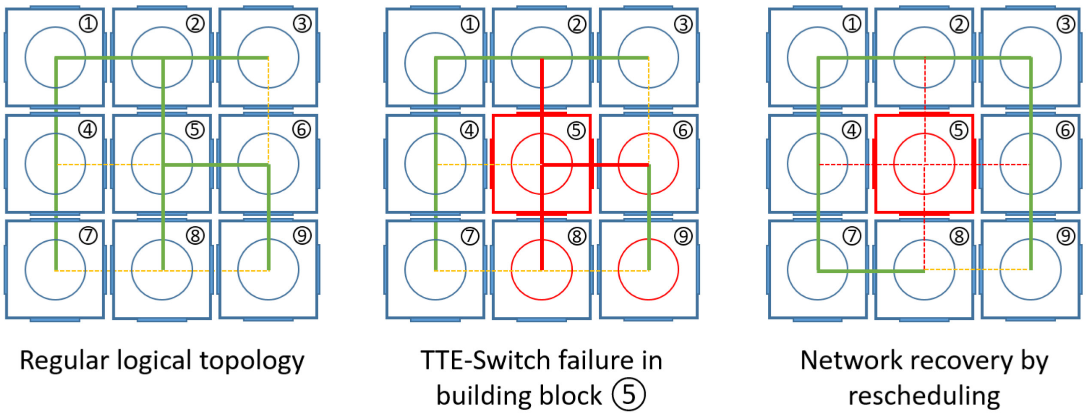


Figure 5.13: Recovery of a network with a TTE-Switch failure

Figure 5.13 illustrates the recovery process of the data network in case of a TTE-Switch failure. When a TTE-Switch failure is detected in the network, the health master must immediately initialize the recovering process by sending the command to reschedule to all building blocks of the satellite. To ensure quick failure handling for this case, the needed recovery schedules shall be stored on every OBC. This also means that each OBC of the network must store a dedicated recovery schedule for every TTE-Switch of the satellite except for the TTE-Switch in its building block. In the example illustrated in Figure 5.13, the modular satellite is made of 9 building blocks and on the OBC of every building block, 8 recovery schedules are stored.

5.3 Reconfigurability of the on-board data networks during OOS scenarios

Different OOS scenarios using a robotic servicer satellite for highly modularized satellites have different complexity with respect to the on-board communication. As mentioned before, time-triggered communication systems aboard the modular satellite are not known for their flexibility and reconfigurability in operation. The lack of reconfigurability of time-triggered systems is due to the need of a precise a priori knowledge about system activities to schedule the time-triggered data traffics.

In OOS missions where the modular satellite can be reconfigured or where building blocks can be removed or added on orbit, the changes of the physical network topology usually require a rescheduling of the on-board data communication.

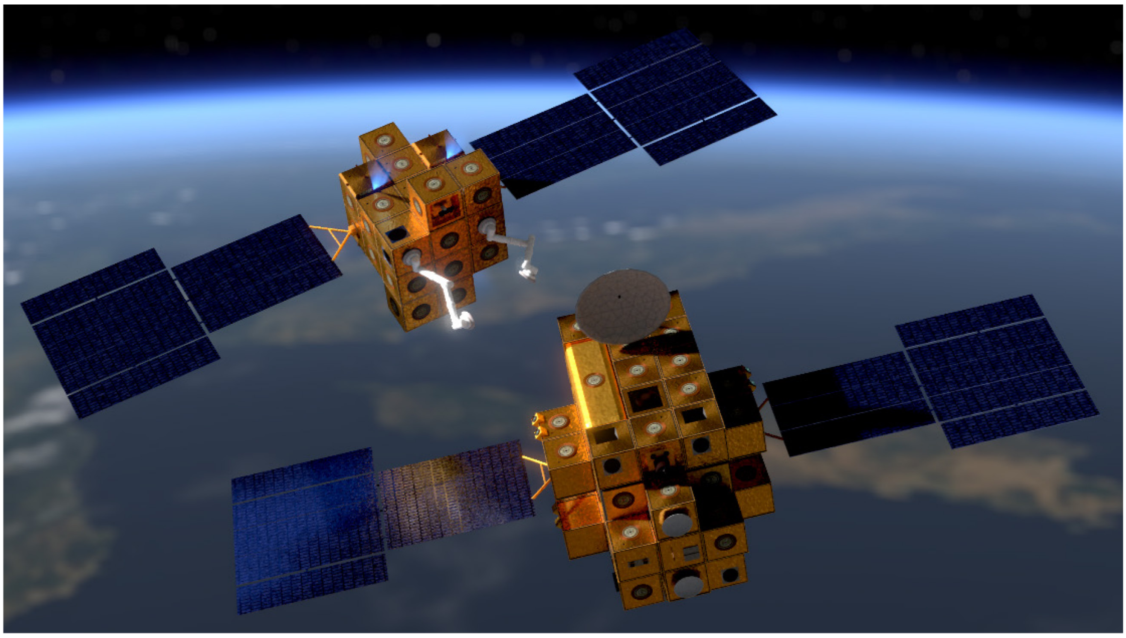


Figure 5.14: Artist's illustration of an on-orbit servicing mission in iBOSS project [10]

To determine which OOS scenario would require a rescheduling of the on-board data network, the following possible OOS use cases for highly modularized satellites comprised of system modules (e.g. the HISats in the PHOENIX program or the iBLOCKs in the iBOSS project) are identified:

- **OOS-Exchange**: Exchange of modules on modular satellites – exemplary scenarios: exchange of modules with hardware failure (failure of the OBC, failure of the TTE-Switches or failure of other critical components), exchange of modules with newer hardware to improve the satellite performance, etc.
- **OOS-Extension**: Adding modules to the modular satellite – exemplary scenarios: modular satellites can be upgraded with new hardware, new payloads or additional propellant tank to prolong the mission lifetime, etc.
- **OOS-Reconfiguration**: Reconfiguration of the building blocks configuration of the modular satellite – exemplary scenarios: the reconfiguration of the modular satellite on orbit is necessary to fulfil the mission objectives, removing weak designed aspects of the actual configuration, etc. Another exemplary application can be found in the definition of new mission objectives after an accomplished satellite mission or even during a mission.
- **OOS-Removal**: Removal of modules – this use case may be rare, but for some cases errors occurring on modules of the operating modular satellite on orbit may have critical impacts on the system performance and so, represent a potential threat to the mission or even to the satellite. For these cases, the

defective modules need to be removed in shortest possible time, even if no appropriate replacement modules are available on orbit.

Not all of the OOS use cases listed above need a rescheduling of the network communication. The flow diagram in Figure 5.15 determines the use cases where a rescheduling of the on-board data network is required.

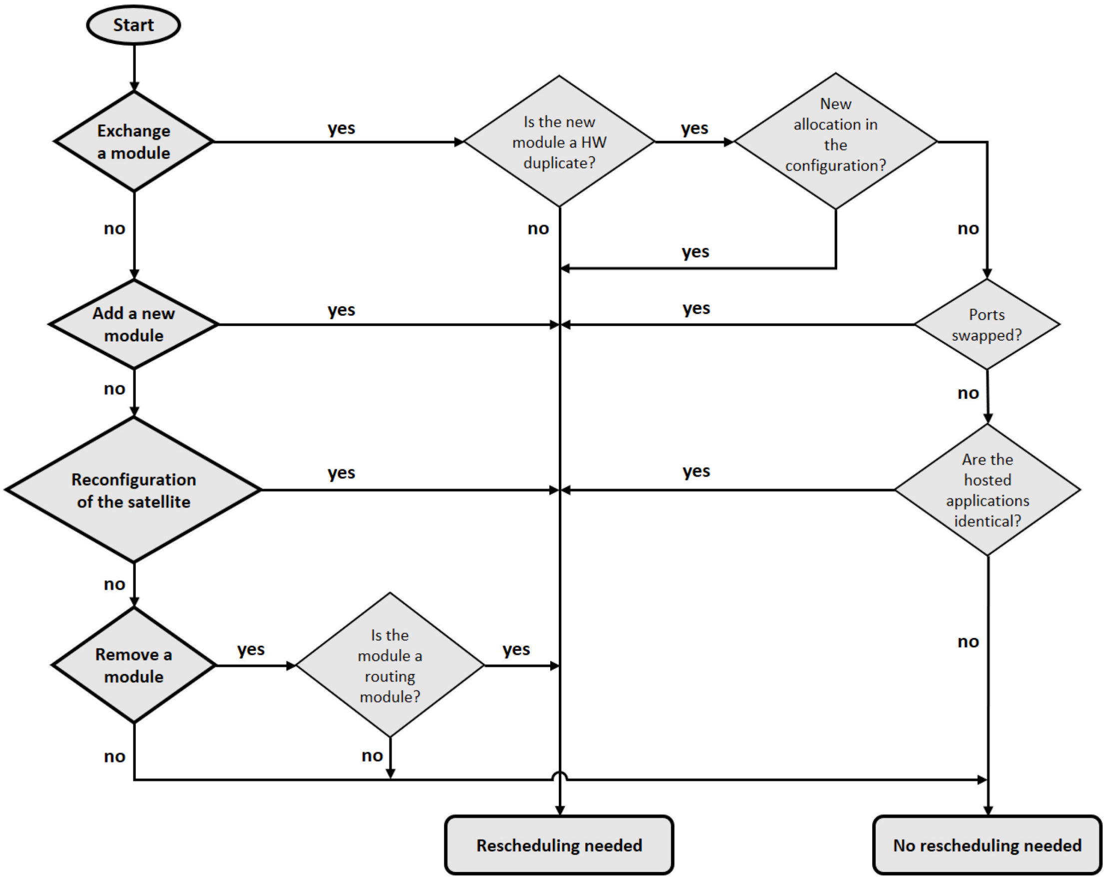


Figure 5.15: Flow diagram to check for rescheduling necessity of OOS use cases

The use case OOS-Exchange does not need a network rescheduling if the replacement module is an exact hardware duplicate of the replaced module and all system configuration parameters like hosted applications, port assignments, MAC and IP addresses of the components, etc. are exactly the same as well as the allocation of the new module on the modular satellite. The use case OOS-Removal also needs no rescheduling if no VLs is routed over its TTE-Switches. In all other use cases a rescheduling of the on-board data network is needed.

In order to coordinate the OOS procedure, two dedicated modules in the satellite need to be defined as the coordinators for the OOS procedures after the Launch and Early Orbit Phase (LEOP), subsequently named as **OOS-masters** (one main master and one redundant master for the case of failure of the main master). It is important that the OOS-masters must provide about telecommunication components implemented within the modules to communicate with the OOS servicer satellite.

Furthermore, it is required that the building blocks except on the OOS-masters shall have a state **OOS-State** defined in their state machine. The OOS-State is similar to the spacecraft safe mode. While in safe mode all non-essential systems are shut down except the essential functions like e.g. TCS, ACS or telecommunication, the OOS-State allows only the OBCs and the on-board data communication network being active and the power subsystem to supply the network devices. In this state, the modules shall only receive or route messages and must not actively send messages to the network. The satellite functions shall be deactivated. The OBCs shall be set in an idle state which can only be left by commands of the OOS masters. Generally, all OOS missions must be announced to the modular satellite by the ground station ahead of the mission initiation. The final configuration of the satellite after OOS shall be planned, simulated and tested meticulously before sending the calculated communication schedule and the necessary configuration files to the modular satellite in operation via uplink.

In following, different OOS use cases will be proceeded sequentially from the point of view of the on-board data network. For the OOS missions of modular satellites using a robotic servicer satellite, RVD manoeuvres of the robotic servicer satellite and the client satellite must be performed. The RVD manoeuvres of the OOS participants are not regarded in this work. Further, the modular satellite is assumed as cooperative. OOS of uncooperative client satellites are not regarded in this thesis. It seems reasonable to consider that in docked formation of the OOS servicer and the modular satellite, the servicer shall overtake the attitude control function to stabilize the formation since two simultaneously and independently operating attitude control systems on a coupled structure may result in instability and an uncontrollable state.

5.3.1 OOS-Reconfiguration

After the robotic servicer successfully docked on the client satellite, it can changes the configuration of the modular satellite by decoupling one or more building blocks of the satellite and move them sequentially to another position on the modular satellite until the planned configuration of the modular satellite is achieved.

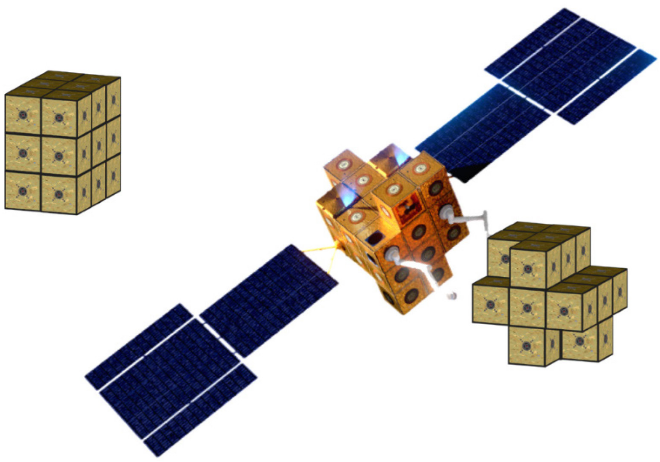


Figure 5.16: Exemplary reconfiguration of a modular satellite

An exemplary OOS-Reconfiguration scenario is shown in Figure 5.16. The procedure of the OOS-Reconfiguration will be described in follow:

Phase 1: Prearrangements

- OOS plan and network configuration files for the final configuration of the modular satellite are sent to the satellite as software upload.
- The network configuration files for every TTE-Switches and TTE-Endsystems are stored in the OBCs within the same building blocks.
- The satellite sends an acceptance message for the planned OOS operation to ground operator via downlink.

Phase 2: Initialisation

- After the servicer docks on the client satellite, it will establish the intersatellite communication to the modular satellite to send the command to start the OOS operation to the OOS master module.
- The OOS master sends a command to all OBCs via a broadcast RC message to initiate the OOS process.
- All OBCs send acknowledgment back to the OOS-master, load the local stored network configuration files on the TTE-Switches and TTE-Endsystems, then restart the network devices and change to OOS-State.
- OOS-master loads the new network configuration files on the TTE-Switch and TTE-Endsystem inside its own building block and restart the network devices to apply the new configuration and sends a message via RF link to the servicer to release the OOS operation.

Phase 3: Reconfiguration of the satellite

- The servicer starts the reconfiguration of the modular satellite.
- After the successful reconfiguration process, the servicer sends a message to the OOS-master to confirm the end of the reconfiguration.

Phase 4: Wake up

- The OOS-master sends a broadcast message to all OBCs to change back to the normal operation mode.

Figure 5.17 illustrates the sequence diagram of the on-board data communication during the OOS-Reconfiguration process.

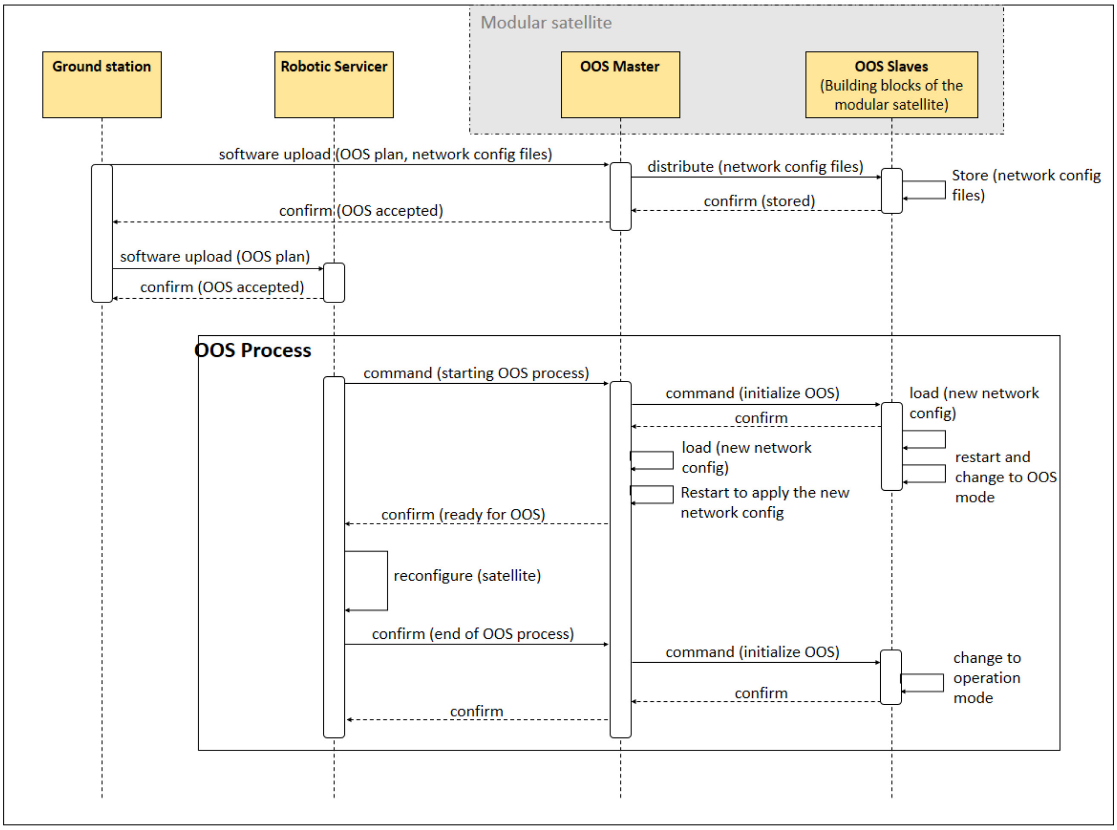


Figure 5.17: Sequence diagram of the OOS-reconfiguration

5.3.2 OOS-Exchange and OOS-Removal

In the use cases OOS-Exchange and OOS-Removal, it must be distinguished between two cases. In the first case, the building block which will be replaced or removed is still functional and operative. This case is the simpler to handle. The procedure is the same as in the OOS-Reconfiguration or OOS-Extension. If the building block is not functional anymore and also not cooperative, for example due to a failure of the TTE-Switch in it, the following design specification must be regarded in designing the mechanical interface coupling the building blocks: Since no communication can be established to this building block, it is required that the coupling mechanism of the interconnection between the building blocks must allow a decoupling from any side of the interconnection of two building blocks regardless which mechanical interface was the active one for the mechanical coupling. The mechanical interface as part of the iSSI in the project iBOSS provides such feature [48, 49]. Only then, the decoupling of a non-cooperative building block can be carried out. According to Table 5.3, a rescheduling of the on-board network is not always necessary for.

Phase 1: Prearrangements

- The OOS plan and, if rescheduling is required, the network configuration files for the final configuration of the modular satellite are sent to the satellite as software upload.

- The network configuration files for every TTE-Switches and TTE-Endsystem are stored on the OBCs within the same building blocks, if rescheduling is needed.
- The satellite sends an acceptance message for the planned OOS operation to ground operator via downlink.

Phase 2: Initialisation

- After the servicer docked on the client satellite, it will establish the intersatellite communication to the modular satellite to send the start command for the OOS operation to the OOS-master.
- OOS-master sends a message to all neighbours of the building block which need be exchanged or removed to command the decoupling of the mechanical interconnections and request for confirmation if the release of the interconnection is successfully completed.
- After receiving the confirmation of all requested building blocks, OOS-master sends a command to all OBCs via a broadcast message to initiate the OOS process.
- All OBCs send acknowledgment back to the OOS-master, load the local stored network configuration files on the TTE-Switches and TTE-Endsystems, restart then the network devices, if rescheduling is needed, and change to OOS-State.
- OOS-master, if needed, loads the new network configuration files on the TTE-Switch and TTE-Endsystem inside its own building block and restart the network devices to apply the new configuration and sends a message via RF link to the servicer to release the OOS operation.

Phase 3: Exchange/Removal of the modules

- The servicer starts the replacement or removing of the building blocks.
- After the successful reconfiguration process, the servicer sends a message via RF communication to the OOS-master to confirm the end of the OOS process.

Phase 4: Wake up

- The OOS-master sends a message to all OBCs via broadcast to change back to the normal operation mode.

5.3.3 OOS-Extension

In an OOS-Extension mission of a modular satellite, the robotic servicer satellite needs to carry the new building blocks on it. The new building blocks shall already have the new network configuration implemented on their TTE-Switches and TTE-Endsystems and already be set in the described OOS-State. The procedure of the OOS-Extension is then the same as for the OOS-Reconfiguration. Such an exemplary OOS-Extension scenario is shown in Figure 5.18.

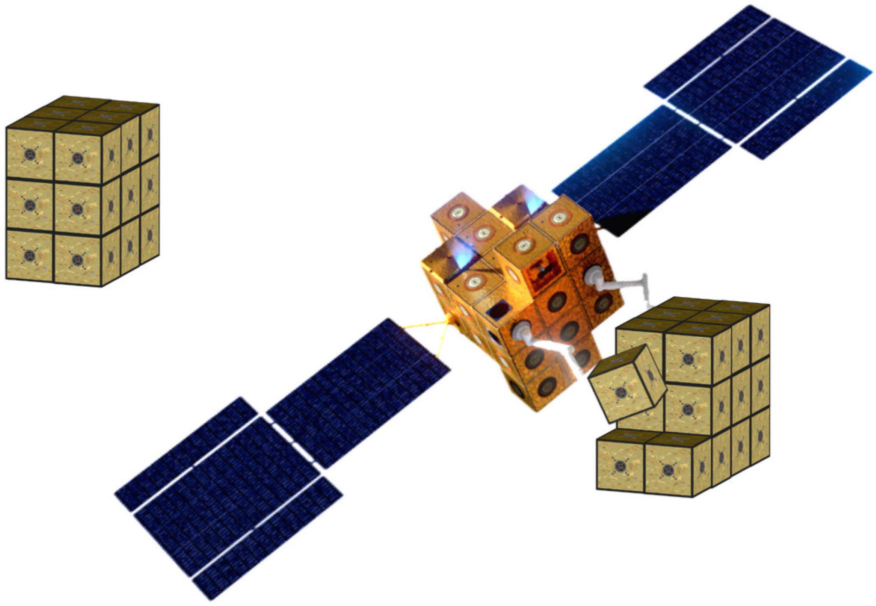


Figure 5.18: Exemplary scenario of an OOS-Extension of a modular satellite

5.4 Concluding remarks

In this chapter, methods and concepts for engineering the data communication network of highly modularized satellite systems are presented. For calculating a feasible communication schedule on-board a modular satellite, the system parameter and communication requirements of the distributed applications on the satellite are summarized coherently in a logical and a physical network model. This approach for calculate a communication schedule is demonstrated for an exemplary 2x2 modular satellite by using the ^{TTE}Tools provided by the company TTTECH.

Since a highly modularized satellite with redundant data paths contains network loops, a method to handle this problem is proposed. In Section 5.2, methods to enhance the network reliability are presented. Herein, a deep insight in the different types of failure in the data network of a modular satellite was given as well as methods to detect and to categorize these failures followed by concepts for network recovery for each failure type. The presented methods can be implemented as additional communication protocols to the protocol stack of TTEthernet to complement the applicability of the proposed data communication system on a modular satellite. In order to meet the requirements of reconfigurability in operation, different scenarios of OOS are discussed and sequential communication procedures for each OOS use cases are developed.

The evaluation of the proposed approaches and methods is performed in the following chapter with an experimental setup of the on-board data communication network for a modular satellite.

6 EVALUATION OF THE PROPOSED CONCEPTS

In this chapter, the concept for the on-board data communication system including the methods and approaches proposed in this thesis is evaluated on an experimental model of the data network of the iBOSS IOD satellite, a small modular satellite comprised of four iBLOCKs in the iBOSS project. The iBOSS IOD satellite is presented in Section 6.1 as well as the experimental setup for the on-board data communication network with four OBC-Switch-Boards forming a network loop. In Section 6.2, the scheduling approach is applied to calculate a network schedule for the communication of on-board applications and functions. The results of the network scheduling are analysed, discussed and assessed in the same section. In Section 6.3, the dependability of the communication network are approved by the failure detection methods and the network recovery mechanisms implemented on the OBC-Switch-Boards. Lastly, the procedures of the data communication system in different OOS use cases are realised on the experimental setup and verify the reconfigurability of the proposed data communication system for modular satellites in Section 6.4. The results of the experimental verification are summarized in the concluding remarks in 6.5.

6.1 Experimental model of the data network aboard a small modular satellite

For the modular satellite concept developed in the project iBOSS, an IOD mission was planned. The mission goal was to launch a small modularized satellite comprised of four fully integrated iBLOCKs to LEO as a technology demonstration of the iBOSS concept and to qualify the components of the modular satellite for space applications.

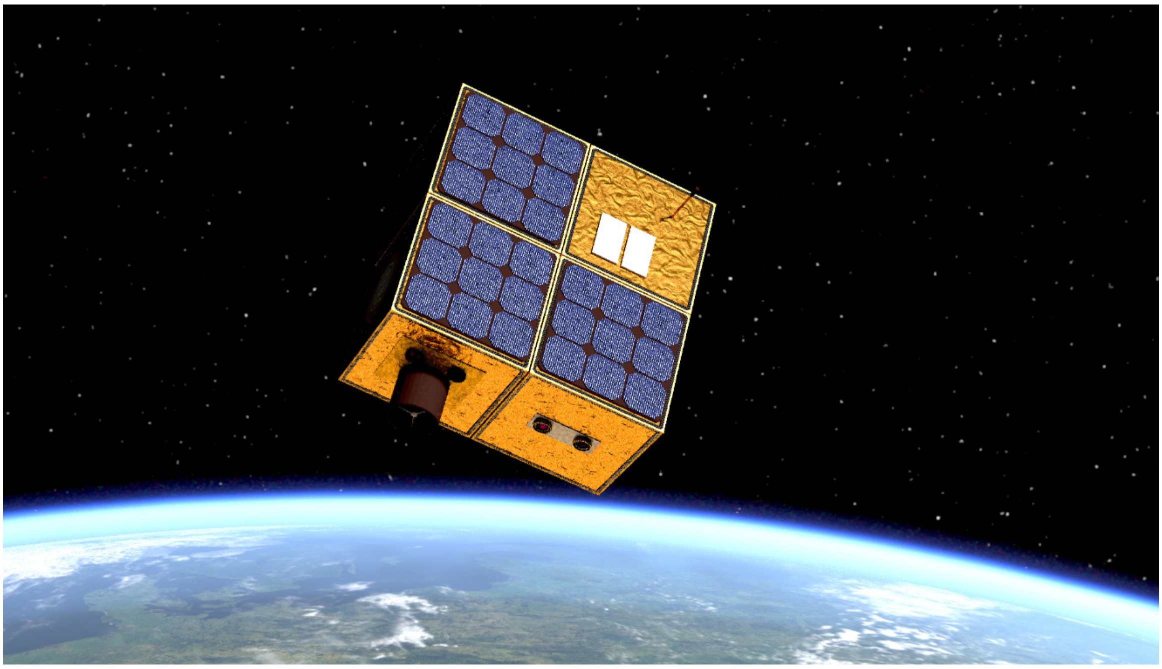


Figure 6.1: Planned iBOSS IOD mission with 4 iBLOCKs (Artist's depiction) [10]

In the IOD satellite, the developed network hardware (OBC-Switch-Boards and data interfaces) should be integrated and the concept for the on-board data communication network proposed in this thesis should be applied. The left part in Figure 6.2 shows the planned allocation of the OBC-Switch-Boards in the iBLOCKs and the right part illustrates the block diagram of the on-board data network. As a demonstration of the loop-tolerance of the data network concept, the four OBC-Switch-Boards will be connected and form a physical network loop as shown in the block diagram of the on-board data network in Figure 6.2.

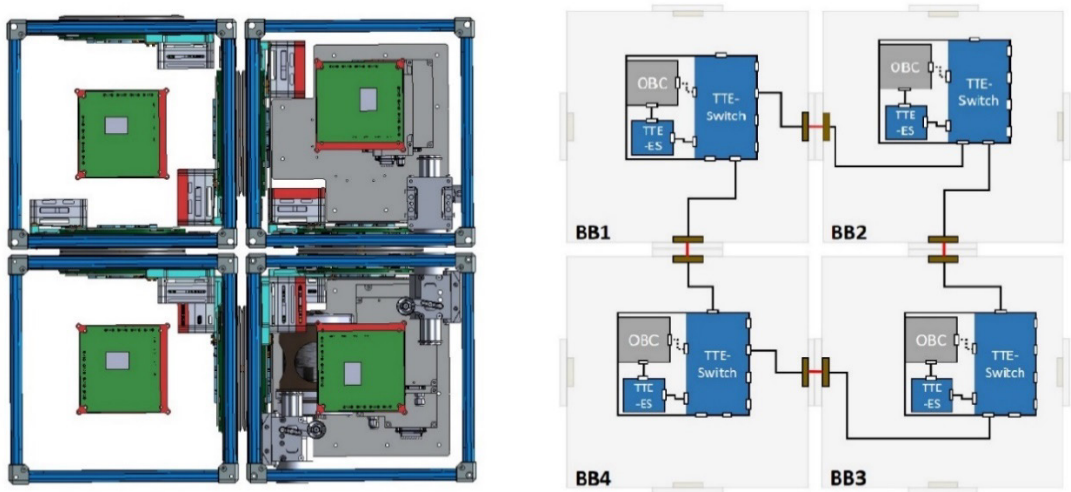


Figure 6.2: CAD model of the EQM and block diagram of the on-board data network

The physical topology of the on-board data network on the iBOSS IOD satellite and the planned on-board data communication can be emulated nearly exactly and tested in terrestrial environment by using the experimental setup with the developed network hardware as shown in Figure 6.3. Network communication schedules for various scenarios of the mission can be calculated, analysed and optimised with different space components or payload implemented.

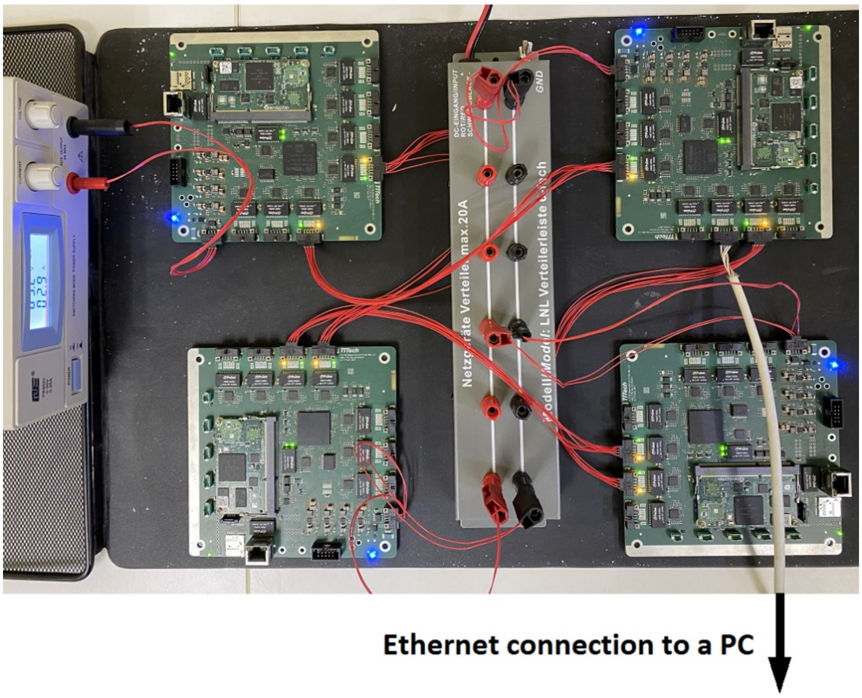


Figure 6.3: Experimental setup of the data communication network on the iBOSS IOD satellite

Since the functionality of the data interface was already shown by functional tests, measurements and the space qualifying process described in Chapter 4, for simplification, the data interface is not applied in the experimental setup. The physical connection between two OBC-Switch-Boards on the satellite is substituted by a network cable, as shown in Figure 6.3. Also, the RF connection of the satellite to the ground station or to a robotic servicer in an OOS scenario is simply simulated by an Ethernet connection of a OBC-Switch-Board to a PC in the experimental setup. New network configuration files in case of rescheduling can easily be loaded on the 4 OBC-Switch-Boards via this Ethernet link and the network performance can be monitored on the connected PC. Further, an application to send and receive time-triggered and best-effort messages in the network is implemented on every OBC to simulate the satellite on-board data communication between the building blocks.

6.2 Evaluation of the communication scheduling approach

The needed inputs for calculating the schedule for the network communication are determined by the physical and logical network model of the data communication system as described in Section 5.1.1 of the previous chapter. While the physical model is given by the physical topology and the specifications of the OBC-Switch-Boards, building an exact logical network model needs the detailed knowledge about all integrated components on the satellite, the communication demands of the subsystems on the satellite and consequently, the detailed communication plan between the building blocks. Unfortunately, at the time when this thesis is finished, the concept for the iBOSS IOD satellite has not been consolidated yet. Hence, a communication demands plan for the distributed applications of the satellite is assumed based on the following integrations concept for the iBOSS IOD satellite.

In the first and third build block (marked as **BB1** and **BB3**in Figure 6.2) sensors and actuators of the ACS are integrated. Since the typical control rate of small satellites is 2 Hz and can be up to 20 Hz [99], 2 CT communication jobs with a maximum payload size of 200 Bytes and a period of 50ms are assumed for the exchange of ACS data in order to coordinate the attitude control.

The RF antenna and the RF communication module are integrated in **BB2**. On this building block, 3 CT jobs to the other building blocks with a maximum frame size of 500 Bytes and a period of 100ms. Further, assuming telecommands need to be executed time-critically in all building blocks, a broadcast CT job to **BB1**, **BB3** and **BB4** is carried out by **BB2** with a maximum payload size of 1000 Bytes and a period of 500ms.

In the last building block **BB4**, a small payload component is integrated. Although data from payload can be send via BE traffic to **BB2** to store and transmit subsequently to earth via downlink, **BB4** still carries out a CT job with the maximum frame size of 500 Bytes periodically every 100ms.

Lastly, to grant the possibilities of the communication of telemetry between the building blocks for coordinating other critical applications like the control of the power transmission between the building blocks or the thermal control, every building block provide a CT broadcast job to all the others with the generous frame size of 1000 Bytes and a period of 500ms. Table 6.1 summarizes the assumed communication jobs of the modular satellite.

Table 6.1: CT jobs of the four building blocks of the modular IOD satellite

CT Job	Sender	Destination	Maximum frame size	Period
1	**BB1**	**BB3**	200 Bytes	50ms
2	**BB1**	**BB2**	500 Bytes	100ms
3	**BB1**	**BB2**, **BB3**, **BB4**	1000 Bytes	500ms
4	**BB2**	**BB1**	500 Bytes	100ms
5	**BB2**	**BB3**	500 Bytes	100ms
6	**BB2**	**BB4**	500 Bytes	100ms
7	**BB2**	**BB1**, **BB3**, **BB4**	1000 Bytes	500ms
8	**BB3**	**BB1**	200 Bytes	50ms
9	**BB3**	**BB2**	500 Bytes	100ms
10	**BB3**	**BB1**, **BB2**, **BB4**	1000 Bytes	500ms
11	**BB4**	**BB2**	500 Bytes	100ms
12	**BB4**	**BB1**, **BB2**, **BB3**	1000 Bytes	500ms

As mentioned, since payload data traffic is not time critical, data from the payload component integrated in **BB4** shall be transmitted via BE traffic to **BB2**. Generally, BE messages can always be transmitted throughout the network when no CT traffics are proceeding.

The assumed communication plan for the IOD mission is used to build the logical network model. The logical and physical network model are described in a XML network description file and the network schedule can be calculated by using the **TTETools** package.

The total calculation duration of the on-board data network schedule is 3.22s. The resulted network configuration file is used to build the binary image files for the TTE-Switches and TTE-Endsystems on the OBC-Switch-Boards. The size of binary files is listed in Table 6.2.

Table 6.2: File sizes of the binary files for network devices

Building block	Devices	Binary file size
BB1	ES	2,816 Bytes
	SW	8,936 Bytes
BB2	ES	2,924 Bytes
	SW	8,828 Bytes
BB3	ES	2,816 Bytes
	SW	8,752 Bytes
BB4	ES	2,707 Bytes
	SW	8 748 Bytes
Total Size		46,527 Bytes

The sizes of the binary files indicate a modest memory consumption for the recovery schedules on the OBCs needed for the network recovery mechanism in case of failure occurrences in the network. For the failure of each TTE-Switch and the failure of each link between two neighbouring building blocks, a recovery schedule is calculated and stored on the OBCs. In summary, every OBC stores 7 recovery schedules containing the binary files for the TTE-Switch and the TTE-ES in its building blocks and needs totally around 80 kB memory.

The binary files can be transmitted to the OBC-Switch-Boards via the Ethernet connection to the PC by using the Trivial File Transfer Protocol (TFTP). By allowing the dynamic routing ability of the TTE-Switches, as described in Section 5.1.2, the network communication is not affected by the network loop in the physical topology. According to the decision parameters defined in Section 5.1.1.2, results of the network scheduling approach are listed in Table 6.3.

Table 6.3: Results of the network scheduling for the IOD satellite

Job	Sender	Receivers	(Longest) Path	Max Latency
1	**BB1**	**BB3**	ES1 – SW1 – SW4 – SW3 – ES3	1092 µs
2	**BB1**	**BB2**	ES1 – SW1 – SW2 – ES2	809 µs
3	**BB1**	**BB2, BB3, BB4**	ES1 – SW1 – SW4 – SW3 – ES3	1315 µs
4	**BB2**	**BB1**	ES2 – SW2 – SW1 – ES1	816 µs
5	**BB2**	**BB3**	ES2 – SW2 – SW3 – ES3	1092 µs
6	**BB2**	**BB4**	ES2 – SW2 – SW3 – SW4 – ES4	1085 µs
7	**BB2**	**BB1, BB3, BB4**	ES2 – SW2 – SW3 – SW4 – SW1 – ES1	1368 µs
8	**BB3**	**BB1**	ES3 – SW3 – SW4 – SW1 – ES1	1092 µs
9	**BB3**	**BB2**	ES1 – SW1 – SW2 – ES2	1062 µs
10	**BB3**	**BB1, BB2, BB4**	ES3 – SW3 – SW4 – SW1 – SW2 – ES2	1591 µs
11	**BB4**	**BB2**	ES4 – SW4 – SW1 – SW2 – ES2	1568 µs
12	**BB4**	**BB1, BB2, BB3**	ES4 – SW4 – SW1 – SW2 – ES2	2074 µs

The four first jobs, which are started synchronously, are job 1 on **BB1**, job 2 on **BB2**, job 4 on **BB3** and job 11 on **BB4**. The frame, which arrives on the destination OBC (**BB2**) at the latest, belongs to job 12. Further, following interesting observations can be made by analysing the calculated network schedule:

- The job with most hops (job 7) is not the job with maximum latency (job 12) – latency also depends on the port relays determined for the TTE-Switches in the schedule. If necessary, using the constraint *E2ELatencyConstraint* can restrict the end-to-end latency for a certain job.
- In job 7 and job 10, the longest path through the network refers to the transmission of a frame from sender (**BB3** in job 10) to its neighbouring building block (**BB2** in job 10) by taking a detour instead of using the direct link between them. This behaviour is due to several effects, for example the direct route is occupied in the time windows, taking the detour may help to keep the deadline of the job. If necessary, using the constraint *BlacklistRoutingConstraint* can exclude some device ports from the routing solution space and so force the scheduling algorithm to find a better path for the job.

The last job is finished at around 2.35ms. Even for the minimum defined period of 50ms of the logical model, there is enough time buffer remained. The bandwidth utilization on every link of the scheduled network communication is shown in Figure 6.4.

PHYSICAL LINK NAME	SOURCE PORT	DESTINATION PORT	MEDIA SPEED	PCF USAGE [%]	RC USAGE [%]	TT USAGE [%]	AGGREGATED BANDWIDTH UTILIZATION [%]
ES_1_P1_SW_1_P0_left_ES_1_P1_TO_SW_1_P0	ES_1_P1	SW_1_P0	100Mbps	0,3072	0	0,3937	0,7009
ES_1_P1_SW_1_P0_right_SW_1_P0_TO_ES_1_P1	SW_1_P0	ES_1_P1	100Mbps	0,1024	0	0,4429	0,5453
ES_2_P1_SW_2_P0_left_ES_2_P1_TO_SW_2_P0	ES_2_P1	SW_2_P0	100Mbps	0,3072	0	0,3937	0,7009
ES_2_P1_SW_2_P0_right_SW_2_P0_TO_ES_2_P1	SW_2_P0	ES_2_P1	100Mbps	0,1024	0	0,4429	0,5453
ES_3_P1_SW_3_P0_left_ES_3_P1_TO_SW_3_P0	ES_3_P1	SW_3_P0	100Mbps	0,3072	0	0,3937	0,7009
ES_3_P1_SW_3_P0_right_SW_3_P0_TO_ES_3_P1	SW_3_P0	ES_3_P1	100Mbps	0,1024	0	0,4429	0,5453
ES_4_P1_SW_4_P0_left_ES_4_P1_TO_SW_4_P0	ES_4_P1	SW_4_P0	100Mbps	0,3072	0	0,1476	0,4548
ES_4_P1_SW_4_P0_right_SW_4_P0_TO_ES_4_P1	SW_4_P0	ES_4_P1	100Mbps	0,1024	0	0,1969	0,2993
SW_1_P2_SW_2_P2_left_SW_1_P2_TO_SW_2_P2	SW_1_P2	SW_2_P2	100Mbps	0	0	0,3199	0,3199
SW_1_P2_SW_2_P2_right_SW_2_P2_TO_SW_1_P2	SW_2_P2	SW_1_P2	100Mbps	0,3072	0	0,3691	0,6763
SW_1_P5_SW_4_P5_left_SW_1_P5_TO_SW_4_P5	SW_1_P5	SW_4_P5	100Mbps	0,1024	0	0,2707	0,3731
SW_1_P5_SW_4_P5_right_SW_4_P5_TO_SW_1_P5	SW_4_P5	SW_1_P5	100Mbps	0,6144	0	0,1969	0,8113
SW_2_P3_SW_3_P3_left_SW_2_P3_TO_SW_3_P3	SW_2_P3	SW_3_P3	100Mbps	0	0	0,2707	0,2707
SW_2_P3_SW_3_P3_right_SW_3_P3_TO_SW_2_P3	SW_3_P3	SW_2_P3	100Mbps	0,1024	0	0,3691	0,4715
SW_3_P4_SW_4_P4_left_SW_3_P4_TO_SW_4_P4	SW_3_P4	SW_4_P4	100Mbps	0,3072	0	0,1723	0,4795
SW_3_P4_SW_4_P4_right_SW_4_P4_TO_SW_3_P4	SW_4_P4	SW_3_P4	100Mbps	0,1024	0	0,2953	0,3977

Figure 6.4: Bandwidth utilization for each physical link of the on-board data network of the iBOSS IOD satellite

The aggregated bandwidth utilization of the CT traffic is lower than 1% for all physical links of the network. Hence, it can be concluded that with the calculated schedule for the data communication system of the iBOSS IOD satellite, there is sufficient bandwidth remained for payload data communication using BE traffic.

The calculated schedule for the experimental setup of the on-board data communication network of the iBOSS IOD modular satellite is illustrated in Figure 6.5.

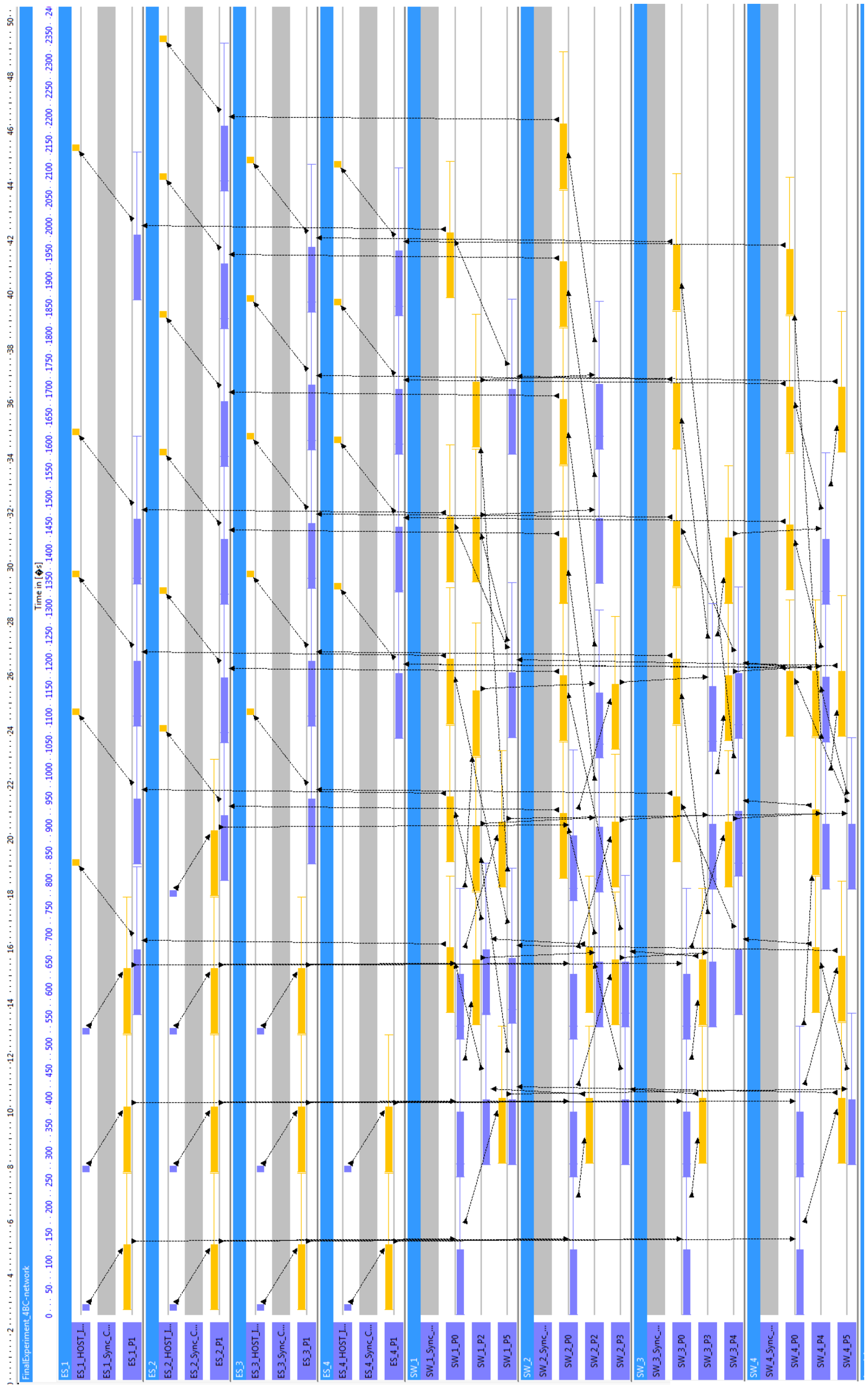


Figure 6.5:Network schedule for the on-board data communication of the iBOSS IOD satellite

6.3 Evaluation of the concept for network recovery in case of failures

In order to evaluate the proposed network recovery concept with the experimental setup, firstly, occurrences of failure must be produced. Although failure of the OBC, failure of the TTE-ES or failure of the TTE-Switch within a building block can be individually simulated by switching off certain ports to cut off the communication to the component but doing this during operation is actually not possible with the test hardware.

Therefore, instead of simulating of the failure of single component on the OBC-Switch-Board, a complete failure of an OBC-Switch-Board will be simulated by manually switching off the power of the OBC-Switch-Board. The so simulated failure of the OBC-Switch-Board is the most critical failure of the failures described in Section 5.2.1 since it represents a failure of the integrated TTE-Switch. In this case, not only the communication to the OBC on the building block is disrupted, no routing of VLs can be carried out by the TTE-Switch.

A link failure between two building blocks can simply be simulated by removing the connecting data cable between two OBC-Switch-Boards. The OBC in **BB2** is defined as the health master. The health check cycle is set to 2s since the maximum response time of the OBC-Switch-Board for SNMP request or ICMP request is set to 500ms.

Seven recovery schedules need to be calculated and stored on each OBC, four for the failures of the links between the building blocks and three for the failure of the blocks (a OBC does not need to store the recovery schedule for the failure of the own building block). The average needed memory size for storing the recovery schedules is 325.4 kB.

6.3.1 Network recovery in case of a link failure between two building blocks

To simulate a link failure between two building blocks during operation, the connecting cable between two OBC-Switch-Boards (**BB1** and **BB4**) in the experimental setup is removed, as shown in Figure 6.3.

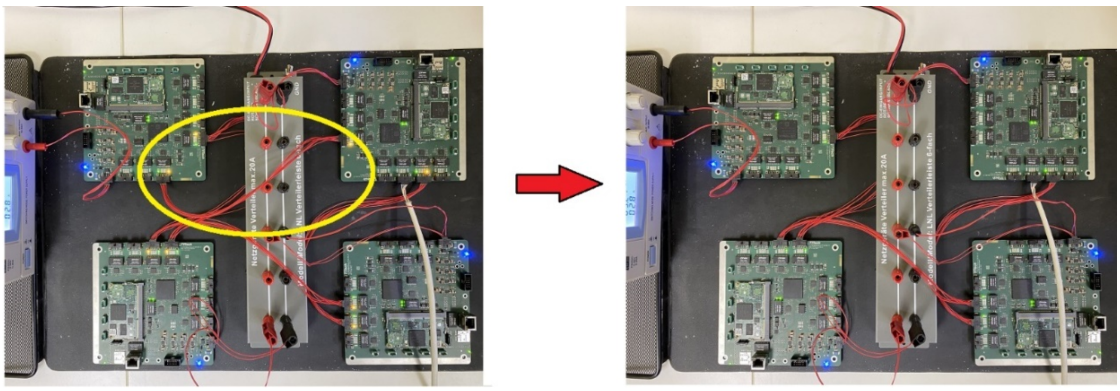


Figure 6.6: Simulation of a link failure in the experimental setup by removing a connecting cable

As a consequence, the VLs over this path are immediately disable. This failure affects job 1, job 3, job7, job 8, job 10, job 11 and job 12 in the original communication schedule in Table 6.3. For the iBOSS IOD satellite, the impact of this failure would be catastrophic since several critical functionalities would be disrupted, inter alia the attitude control. The CT communication between **BB1** and **BB3,** where ACS sensors and actuators are allocated, is disable.

For the resulted physical topology (no physical connection between **BB1** and **BB4**), a recovery network schedule as calculated and stored in all OBCs of the network beforehand. The involved OBCs (OBC1 and OBC4) determine the link failure in a health check cycle and report the failure to the health master in **BB2**. The health master registers the link failure between **BB1** and **BB4** and sends a BE message to all building blocks to load the recovery schedule for this failure on all TTE-Switches and all TTE-Endsystems. The process of failure detection and network recovery is monitored for each OBC and illustrated in Figure 6.7.

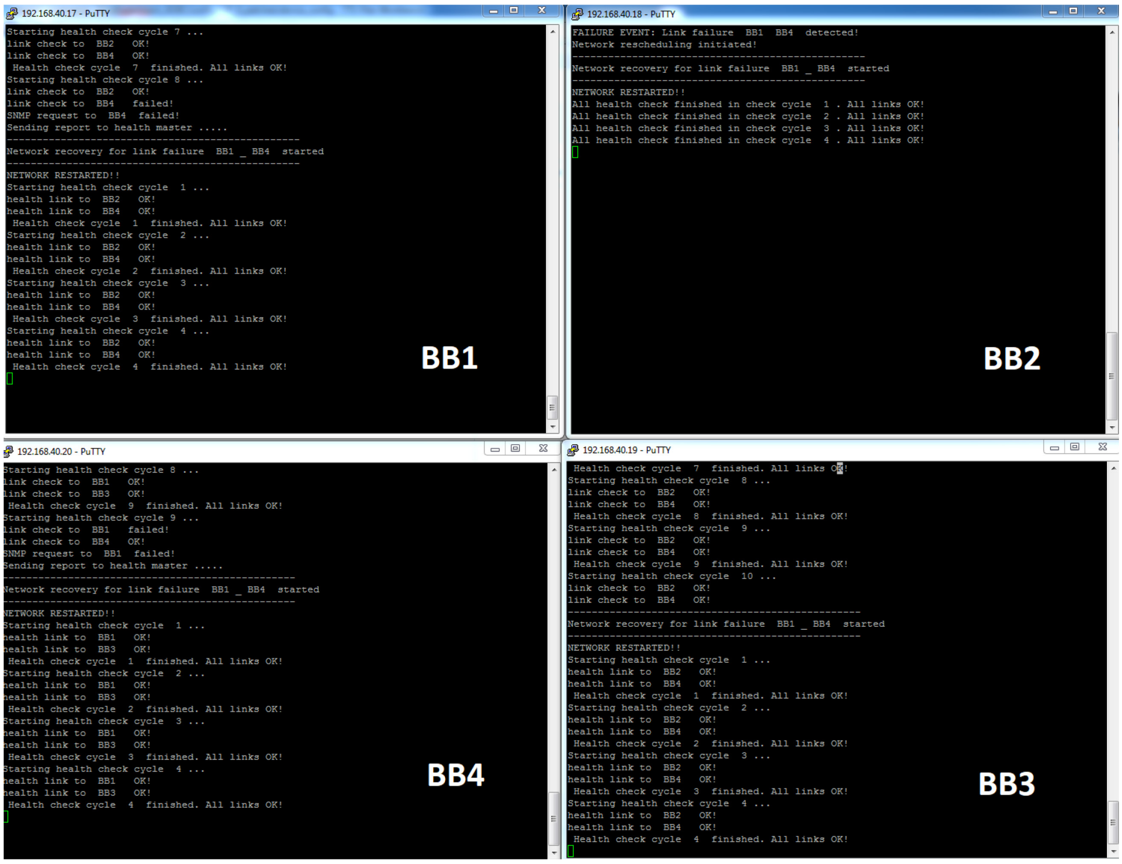


Figure 6.7: Network recovery after a link failure between BB1 and BB4

After the autonomous rescheduling and restart of all OBCs, the network communication is restored. VLs, which were routed over the link **BB1-BB4**, are now with the new schedule rerouted over other links. The total time of the failure detection process from the time point of removing the connection until the initiating of the network recovery is 1.46s. The network recovery process needs nearly no time but the restarting duration of the OBC-Switch-Boards is around 9.8s. The total duration from failure occurrence till the fully restoration of the network communication is 10.96s. The time needed for the network recovery process can be significantly reduced in further works since an unknown waiting timeslot between rescheduling and restart of the OBC-Switch-Boards is coded in the actual firmware.

6.3.2 Network recovery in case of a link failure between two building blocks

A node failure in the experimental setup of the IOD satellite can be simulated by switching of the power of one OBC-Switch-Board during operation. To demonstrate the network recovery in case of a node failure, the power supply of **BB4** was switched off as shown in Figure.6.8.

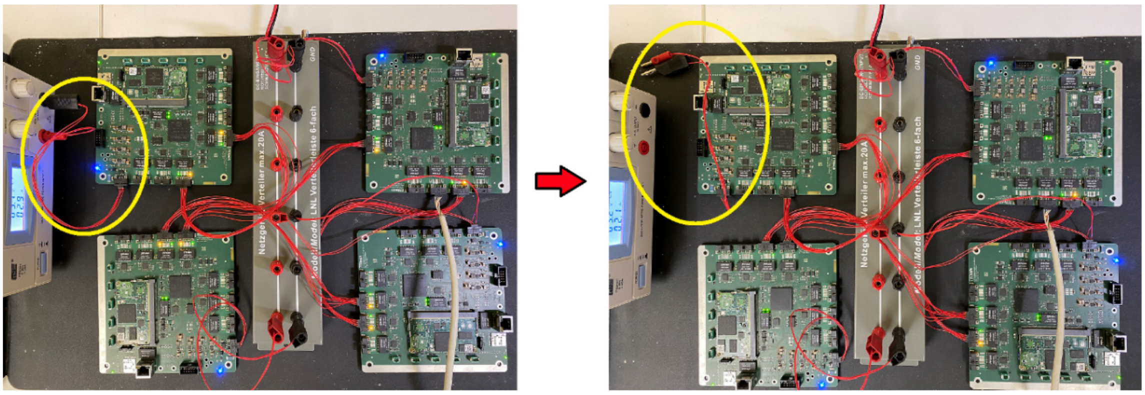


Figure.6.8: Experimental simulation of a failure on BB4

The network recovery approach works similar as for link failure. Outputs of the 4 OBC-Switch-Boards are illustrated in the figure below.

Figure 6.9: Network recovery after a node failure on BB4

The total duration for the network recovery process is with 11.08s comparable to the network recovery in case of link failure described in the previous section and can also be optimized by reducing the waiting timeslot between loading the recovery network and restarting the OBC-Switch-Boards.

6.4 Network reconfiguration for OOS use cases

OOS use cases are simply to simulate with the experimental setup of the on-board data communication network. New configurations of the satellite can be simulated by manually changing the ports of certain OBC-Switch-Board. The data cables are unplugged and reconnected to other ports of the boards. As described before, one building block needs to be defined as the OOS-master to coordinate and monitor the OOS process. For the following experimental verifications of the reconfiguration approach, **BB2** is defined as the OOS-master. The RF connection to the ground station is simulated by the Ethernet link to the PC via the RJ-45 interface of the OBC as shown in Figure 6.10.

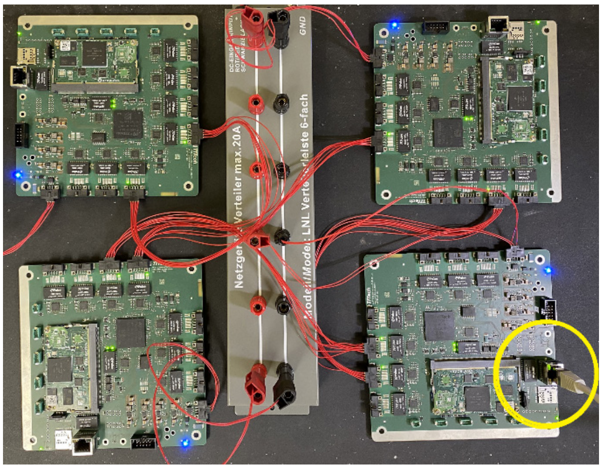


Figure 6.10: Ethernet link to the PC via RJ45 interface of the OBC on BB2

The Ethernet communication to the PC represents the communication to ground station and the intersatellite communication to the servicer for the simulation of the OOS use cases. The network schedules needed for the final configurations of the satellite were loaded over this link and stored on every OBCs in the prearrangement phase as described in Section 5.3.

It is important that the health check application on the OBCs and the network recovery function on the health master must be stopped before reconfiguration. The stored recovery schedules must also be updated after reconfiguration due to the new physical network topology of the satellite.

6.4.1 OOS-Reconfiguration

Figure 6.11 illustrates the simulation of an OOS-Reconfiguration for the experimental setup of the satellite communication network. By unplugging of **BB1** from **BB2** and **BB4** and plugging it to **BB3**, a new physical network topology was created. A network schedule calculated based on this configuration was already distributed and stored on all OBCs. After the manual reconfiguration process (unplugging the **BB1** and connect it to **BB3**) had finished, a message about the succeeded reconfiguration process was sent to OOS-master on **BB2** via Ethernet link and the OOS-master woke up all OBCs in the network and set them back to operation mode.

The communication on the network was restored nearly immediately since all OBCs loaded the new schedule and restarted before they were set in OOS mode.

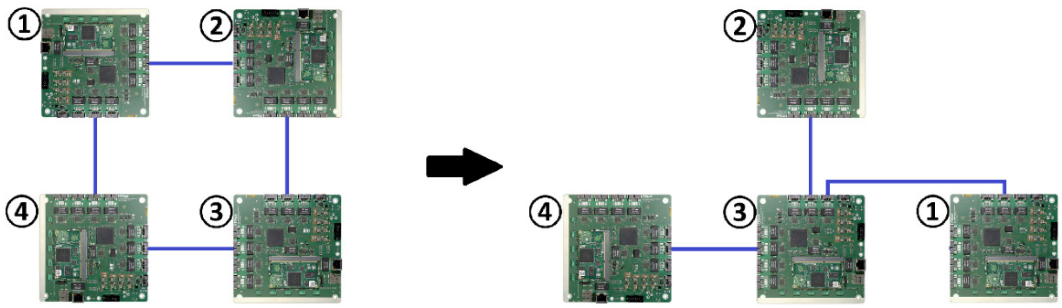


Figure 6.11: Reconfiguration of the exemplary network configuration of the IOD satellite

6.4.2 OOS-Extension

The OOS-Extension use case was realized with the experimental setup by extending the experimental data network of the IOD satellite with the fifth OBC-Switch-Board, which represented the fifth building block added to the satellite by the robotic servicer. As mentioned in Section 5.3.3, the new building block must have the network configuration loaded, which was calculated for the final configuration with 5 OBC-Switch-Boards before it was manually added to the network.

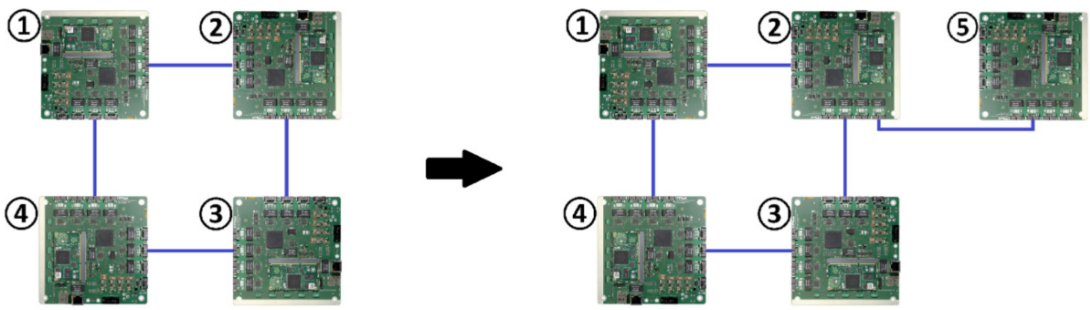


Figure 6.12: Extension of the exemplary network of the IOD satellite with a further building block

Setting the building blocks in the OOS-state in the experiment was not necessary since the extension of the network can be accomplished in very short time. Via the Ethernet link to the PC, a message to command the rescheduling process on all OBCs was sent to the OOS-master on **BB2** which subsequently initiated this process on all OBCs. After the network restart, network communication was immediately restored.

In a real OOS-Extension scenario, it is recommended to set the OBCs into the OOS-state since the extension of the modular satellite with a new building block using a robotic service on orbit may need more time.

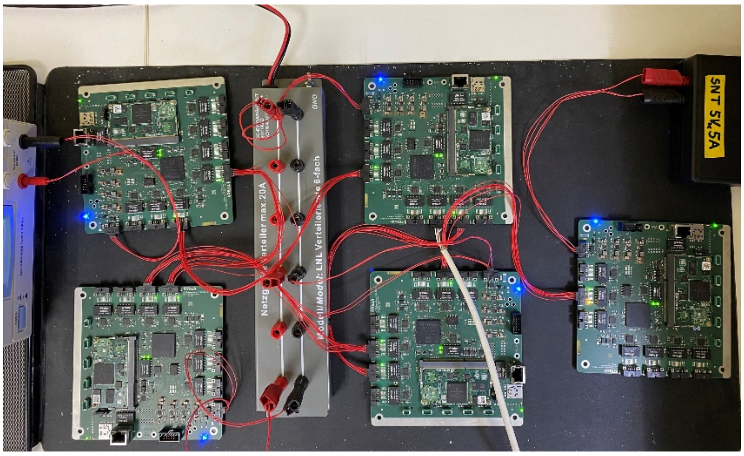


Figure 6.13: Experimental setup with 5 OBC-Switch-Boards

6.4.3 OOS-Exchange and OOS-Remove

The procedures for OOS-Exchange and OOS-Remove use cases are nearly the same as the procedure for reconfiguration if the exchanged or removed building block is cooperative. The OOS-Exchange of a cooperative building block can be simply simulated as by unplugging and reconnecting of an OBC-Switch-Board due to the assumption that the new building block is the exact duplicate of the exchanged one. The OOS-Removal use case is even easier to simulate by removing some connecting data cables. Thus, a verification of these two use cases is trivial.

6.5 Concluding remarks

In this chapter, proposed methods and approaches of the concept of a dependable and reconfigurable on-board data communication system for highly modularized satellites could be applied and proven by experiments using an experimental setup for the on-board data network based on the planned iBOSS IOD satellite.

For the experimental data network, the physical and logical network model were built and used as inputs to calculate the network schedule for the time-triggered communication aboard the modular satellite by using the TTETool of TTTECH. While the physical network model was defined by the given configuration of the building blocks and the hardware parameters, the logical network model had to be built on realistic assumptions of the distribution of space components on the modular IOD satellite and the resultant deterministic and real-time communication demands. The calculated schedule was assessed by decision parameters and it could be shown that the aggregated bandwidth utilization of the time-critical communication was lower than 1% for all physical links in the network. Sufficient bandwidth would remain for payload data communication using BE traffic.

In order to prove the methods to enhance the dependability of the on-board communication system, failures on the data network of a modular satellite were simulated by different experiments. Functionality of the proposed link detection method and the network recovery mechanism was successfully proven. Finally, the proposed procedures for various OOS use cases were applied to the network of the experimental setup and approved the reconfigurability of the on-board data communication concept proposed in the previous chapters of this thesis.

7 CONCLUSIONS AND OUTLOOK

The development of highly modularized satellites with on-orbit serviceability and reconfigurability poses a great challenge to space engineers and researchers. Generally, modular approaches begin with the basic idea of the decomposition of traditional, monolithic spacecraft on subsystem or component level. The spatial distribution of components of various subsystems in several building blocks is a consequence of the modular structure of the system. But, the higher the modularity level, the more complex and more demanding becomes the design of the OBDH and data communication system of such modular satellites. Spatial distribution of components on different building blocks which shall be standalone systems results in the need for a system intelligence in every building block for locally monitoring and controlling of the integrated components on one hand and on the other hand, for globally coordinating with other building blocks to perform the critical functions and applications of the satellite. Generally, on-board data communication system belongs to the mission-critical elements of a spacecraft. Dependability and reliability form here the most important design specifications, especially for highly modularized satellites. The reported communication issue aboard the modular **eXITe** satellite in the DARPA's PHOENIX program, as described in Section 2.2.4, is a cynical confirmation for this statement. Thus, ones can draw the following conclusion:

A highly modularized satellite designed as a heterogeneous system of building blocks with local intelligence is a higcomplex, distributed system with decentral control architecture where the system performance and system dependability are indicated by the performance and reliability of the interprocess and interprocessor and, consequently, the intermodular communication.

On top of all this, the OOS capability of modular satellites in operation extends the already very challenging requirements to the on-board communication system with further demands for flexibility and reconfigurability. Therefore, spacecraft engineers designing a highly modularized satellite need to face the challenge of developing and applying an appropriate on-board data communication system which must provide on one side, high reliability and fault tolerance and on the other side, the needed flexibility and reconfigurability of the communication network. At present, there is no known concept for an on-board data communication system that provides the combination of the required reliability for space applications and reconfigurability for modular satellites in one communication system.

The contributions of this thesis fill up the gap in the state of the art with the concept of an on-board data communication system based on the standard TTEthernet and various methods and mechanisms which are needed to make the proposed data communication concept applicable on highly modularized reconfigurable satellites with the capability of on orbit servicing. Aspects of the proposed concept were also realized in network hardware developments to fulfil the mentioned requirements of the modular satellite designs.

Hereby, the hardware concept encompasses the design, development and qualifying of the optical data interface to enable a wireless near-field data transfer between building blocks of a modular satellite. The multifunctional OBC-Switch-Board unifies the functionalities of an OBC, a TTE-Endsystem and a TTE-Switch in one single component. The design of the OBC-Switch-Board meets the hardware constraints and functional requirements of the on-board communication system and is deployable in the modules of a highly modularized satellite. Finally, the concept for the on-board data communication system is applied and verified with an experimental model of the modular iBOSS IOD satellite to demonstrate its applicability in real satellite missions in the near future.

7.1 Conclusions regarding the objectives of the thesis

The objectives set out at the beginning have been achieved successively during the dissertation. By analysing different modular satellite systems with on-orbit servicing capabilities, several conceptual considerations have been made and discussed in detail and so, a profound understanding of the demands of the on-board data communication system on highly modularized satellites can be gained. This understanding is then formulated in requirements to the data communication system in Chapter 3. Hereby, the first objective is achieved, and the first research question can be answered as follow.

Answer to the first research question: The highly modularized satellite design defines several requirements and constraints to the on-board data communication system. Especially, the spatial distribution of components of the same critical subsystems in different building blocks of the satellite demands a highly reliable and dependable communication which shall provide determinism and real-time capability and is even more critical than in classical monolithic satellites. On same time, caused by the desired OOS capability of modular satellite design, the on-board data communication system must provide reconfigurability and extensibility. The maximum network node number shall not be limited to a low number with regard to larger modular space systems in the future. Since satellite control data and housekeeping data as well as payload data shall be transferred on the same physical layer, the communication architecture shall provide several communication modes with different criticality levels and enough bandwidth for larger payload data traffic. Not every physical topology is appropriate for the modular satellite design. The result of the analysis of several network topology is that the mesh topology is the most suitable topology because it allows redundant data paths and minimizes single points of failure. But this topology also contains network loops and therefore, a network loop handling concept must be regarded.

By attaining the second objective of the thesis, the second research question can be answered. In Chapter 3, a selection of well-known and proven communication architectures was presented which had been successfully applied on space missions or at least, have the potential to be applied in future spacecraft. The chosen communication architectures are studied and compared as to their applicability on highly modularized satellites. For this comparison, several decision parameters were determined by the requirements and the conceptual considerations beforehand.

Answer to the second research question: As a result of the comparison, the most suitable communication standard is determined as TTEthernet since it provides many features especially the three parallel communication modes for applications with different criticality levels. The deterministic and real-time capable time-triggered mode can be used for the critical data traffic of satellite monitoring and control data, the rate-constrained mode for critical event-triggered communication and the best-effort for not critical application like the transmission of payload data. However, time-triggered architectures are known for the lack of flexibility and reconfigurability. Therefore, further protocols, methods and concepts need to be developed to make the on-board data communication system based on TTEthernet standard to fulfil the other requirements of the modular satellite design.

Answer to the third research question: For a precise scheduled communication, all activities on the network including the physical network topology must be known in detail a priori. In order to close this gap, additional methods and approaches have been developed to enable the application of the time-triggered communication for the distributed components aboard the modular satellite. Further, modular satellites with spatially distributed components in different building blocks have particularly high requirements to dependability since the critical functions and applications are distributed on different building blocks of the satellite. The resulted stringent coordination of the decentral intelligence to ensure the performance of these applications and functions on a modular satellite needs higher reliability and fault tolerance of the on-board data communication than on a monolithic satellite, as described in Section 2.4. In order to enhance the dependability of the on-board data network, beside fault prevention methods which were applied in the design and development of the network hardware, error detection and system recovery methods have been developed and verified. Procedures of the on-board communication system for different OOS scenarios have also been elaborated and presented to complete the concept for the on-board data communication system for reconfigurable modular satellites. The presented methods and functions can be implemented as communication protocols.

With the answer of the last research question, the third objective is achieved. The last objective of the thesis is fulfilled by applying and evaluation of the concept on the experimental model of the iBOSS IOD modular satellite and the verification of the proposed approaches in various failure and OOS use cases. Thus, it could be shown successfully and completely, that the proposed concept for the on-board data communication system can be applied on a real modular satellite with OOS capabilities.

7.2 Contributions to iBOSS

As mentioned before, the idea of this thesis was raised in the iBOSS project and knowledge gained during the project are used in this thesis. The verification of the proposed concept was realised by using iBOSS hardware. The contribution to the iBOSS project comprises the comparison of the communication architectures, the application of TTEthernet as the main global bus in iBOSS satellite concept and the design concepts for the network hardware including the optical data interface and the OBC-Switch-Board.

The proposed concept for the on-board data communication system can be used in iBOSS missions in future with the procedures proposed for OOS use cases.

7.3 Contribution outside of iBOSS

7.3.1 Application of the proposed data communication system in terrestrial industry e.g. the automotive industry

The proposed data communication system in this thesis can also be applied in terrestrial applications. TTEthernet is one of the chosen data network system in many terrestrial systems since this network standard provides the union of high data rate, deterministic and reliable communication and the capability to integrate units with classical Ethernet interface into the network and also supports other industrial Ethernet protocols like PROFINET and EtherNet/IP. Especially in the automotive area, TTEthernet may have the chance to become a leading role in near future and because it provides higher data rate than the known FlexRay and higher reliability than the CAN bus [100]. Although reconfigurations of the data network in operation is an uncommon use case in automotive applications and other terrestrial applications, several aspects in this work still contribute knowledge to develop a highly dependable data network system with reconfigurability.

7.3.2 Application of the data communication system in modular space robots and space rovers

An application of modular robot system in space missions and planetary exploration missions was proposed in [54, 101, 102]. However, as mentioned in Section 2.3, at the best knowledge of the author, no data communication concepts deployed on modular robots would provide the needed reliability and fault tolerance for space applications. Reconfigurable modular robots and reconfigurable modular satellites show a high conceptual similarity and the proposed concept in this work will meet the requirements of modular robot systems for space application.

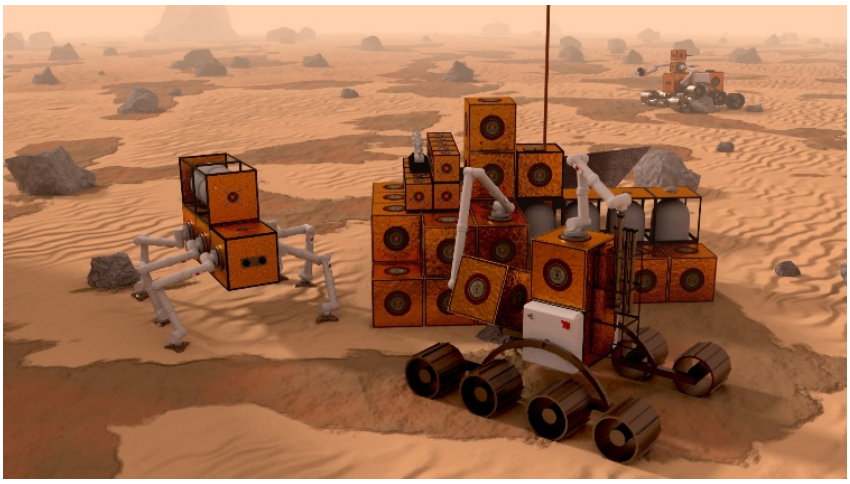


Figure 7.1: Artist's conception of space rover built of iBOSS modules [10]

During the iBOSS project there was ideas of using the building blocks on space rovers for planetary exploration missions. If a space rover provides iBOSS interfaces or even consists of iBOSS building blocks, an application of the proposed data network system is conceivable.

7.4 Outlook and Future works

The field of highly modularized spacecraft is still in the very early stages of its development. The contribution of this thesis is only one of the first steps on this long path in the future of modular space architectures. There are still many possibilities to refine and to optimize the concept in the future works.

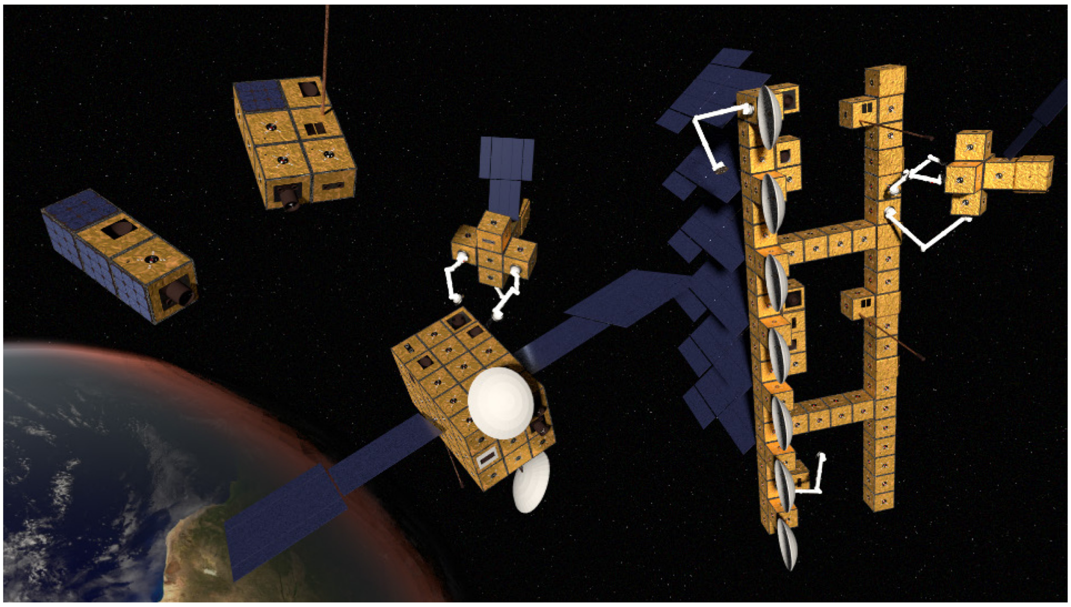


Figure 7.2: Artist's conception of iBOSS' future vision [10]

In following, various promising research directions are listed which can be built on the results of this work:

- **On-board network scheduling and network topology discovery**

  In this work, communication schedules for a data network aboard modular satellites need to be calculated beforehand and loaded on the modular satellite in the prearrangement phase of OOS use cases or stored on the satellite as recovery schedules for every failure point of the network. With the ability to calculate new schedules on-board using the OBCs in combination with autonomous network topology discovery methods by using known network protocols like LLDP, SNMP, etc., the future modular satellite can autonomously reconfigure the network or can recover the data network in case of failure more effectively. Further, the network recovery methods in this thesis are implemented as protocols in the application layer of the OSI model. In the next versions, this functionality shall be implemented in the MAC layer so that the recovery time can be effectively shortened.

- **Partial rescheduling**

  The idea of partial rescheduling contains a network in which some building blocks with shared functions and applications are clustered in a local network within the global satellite data network. These local networks are regarded as an enclosed systems inside the global network and provide an effective possibility to contain failures inside them and therefore, to limit the impacts of failures on the global network.

- **Security network protection methods against intrusion**

  Confidentiality and integrity are the two dependability attributes which are not regarded in this work. A modular satellite with many exposed data interfaces which can be misused for intrusion in the network needs protection methods to prevent unauthorized access.

- **Radiation hardening of OBC-Switch-Board and further space qualifying steps**

  Finally, although the OBC-Switch-Board was developed with a high reliable design and the integrated TTE-SIC should be radiation tolerant, a fully qualification for space applications is still outstanding. Therefore, the next necessary step for qualifying the OBC-Switch-Board for a space mission is completing the space qualification process on different levels.

Appendix

Comparison of communication architectures to their applicability on modular satellites

Features / bus standards	MIL-STD-1553	TTP/C	TTCAN	SpaceWire	IEEE-1394B	TTEthernet	EtherCat
Short Description	- Military standard with defined electrical and protocol characteristics for a data bus for the use in avionic and aerospace area - Most adapted standard in spacecraft - Very high safety, full deterministic and real-time capable	- Fault tolerant time-triggered communication protocol for safety-critical, real-time distributed systems - Developed by the University of Vienna and TTTECH	- Time-triggered and deterministic extension to the CAN bus protocol - High-precise, global network time synchronisation - Low data rate	- Event-triggered, high reliable communication standard developed for space application with high data rate - Developed in collaboration by ESA, NASA, JAXA and RKA for use in spacecraft	- Simple, low-cost interface for peripheral communication in computers and electronics with high bandwidth - Not adapted in spacecrafts yet	- Time-triggered extension of the Ethernet standard to meet time-critical, deterministic or safety-relevant conditions - Full deterministic - High fault tolerance - High safety and high data rate	- Real-time expansion of Ethernet as a fieldbus in control automation - Not adapted in spacecrafts yet - Low communication jitters and low hardware cost - Low flexibility
Application fields and systems	Numerous military and civil spacecraft and avionics	Airbus 380 cabin pressure control, civil and military aircraft and railway	Mostly automotive industry	Several satellite missions of NASA, ESA, JAXA, etc.	Joint Strike Fighter (F-35), Space Shuttle	NASA's ORION MPCV and ESA's ORION's service module	Control automation technology

Features / bus standards	MIL-STD-1553	TTP/C	TTCAN	SpaceWire	IEEE-1394B	TTEthernet	EtherCat
Communication Control Scheme	Event-triggered, master-slave principle	Time-triggered	Time-triggered and event-triggered by using arbitrating frames	Event-triggered	Isochronous data transmission in broadcast mode controlled by a cycle master; asynchronous data transmission can be arbitrated by cycle master in the left time	Time-triggered and event-triggered, three operation modes: TT, RC and BE	Event-triggered, master/slave relationship (1-n)
Multimaster /masterless	No (redundant bus master)	masterless	multimaster	masterless	No	masterless	No
Maximum Data Rate	1 Mbps	5 Mbps using RS-485 physical layer, 25 Mbps using Ethernet physical layer	1 Mbps (125 Kbps in fault-tolerant mode)	400 Mbps	800 Mbps	1000 Mbps	100 Mbps
Maximum node number	31	64	120	224	63 on a bus with up to 1023 buses	Depends on the number of	65535

Features / bus standards	MIL-STD-1553	TTP/C	TTCAN	SpaceWire	IEEE-1394B	TTEthernet	EtherCat
						available switch ports	
Clock synchronization	Not required, since controlled by bus master	Fault tolerant synchronization	Global synched clock using data in reference message	No	Synchronized by cycle master	Distributed synchronization, fault tolerant	Distributed synchronization, initialized by master per broadcast, slaves latch internal clock
Global time base	No	Yes	Yes	No	No	Yes	No
Topology	Multidrop	Multidrop, point-to-point, star	Multidrop	Point-to-point and switched	Peer-to-peer, tree by using bridges	Point-to-point and switched, mesh	Bus, tree and star (by using switches)
Duplex	Half	Half	Half	Full	Full	Full	Full
Physical layer	Twisted pairs or Coaxial, approach for optical links presented	RS-485 or Ethernet physical layer, optical fiber	CAN controller, twisted pairs, Optical CAN Interface (iBOSS)	LVDS, approach for optical links presented in iBOSS	1394 physical layer	Ethernet physical layer, optical fiber	Ethernet physical layer (100BASE-TX), optical fiber by using media converter
Determinism	Yes	Yes	Yes for time-triggered mode	No	No	Yes for time-triggered mode	Yes

Features / bus standards	MIL-STD-1553	TTP/C	TTCAN	SpaceWire	IEEE-1394B	TTEthernet	EtherCat
Fault Tolerance (Fail-operation/Fail-Safe) Mode	Yes	Yes	No	No	No	Yes	No
Fault Containment	Yes by using redundant bus	Yes by using dual-redundant bus	Yes, if CAN controller is not faulty	No	No	Yes by using redundant data paths	No
Babbling idiot Avoidance	No, but can be defined at application level	Yes	Yes, if the CAN controller is not faulty or the transceiver can be switched off	No	No	Yes	No
Network loop tolerance	No	No	No	No	Yes by disabling ports to break loop	Yes	No
Message CRC	No	Yes	Yes	No	Yes	Yes	Yes

Features / bus standards	MIL-STD-1553	TTP/C	TTCAN	SpaceWire	IEEE-1394B	TTEthernet	EtherCat
System tolerance in case of node failure	No in case of failure of master node; yes, in case of failure of slave node	Yes	Limited, since node failure can have impacts on communication	Yes	Depends on topology	Yes	Yes
Extensibility	Yes, up to maximum node number	Yes, up to maximum node number	Yes, up to maximum node number	Yes, up to maximum node number	Yes, up to maximum node number	Yes	Yes, up to maximum node number
Hot Swap Ability	Depends on implementation	Yes, but require network breaking	Yes, but require network breaking	Yes	Yes, swapping requires network breaking	Yes, but require network breaking	Yes
COTS Availability	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Rad-hard hardware Availability	Yes	Yes	Yes	Yes	No	Yes	No

Features / bus standards	MIL-STD-1553	TTP/C	TTCAN	SpaceWire	IEEE-1394B	TTEthernet	EtherCat
Composability	No	Yes	Yes	No	No	Yes	No
Designed for safety critical systems	Yes	Yes	Yes	No	No	Yes	No
Applicability for modular satellites	Although the proven high safety and fault tolerance, MIL-STD-1553 does not provide the flexibility and reconfigurability needed for an application on modular satellites especially regarding to its master-slave-scheme. Furthermore, the low data rate, the missing	TTP/C is a fault tolerant fieldbus with high reliability and several features which would meet many requirements of the modular spacecraft architecture. Especially the capability to handle two data traffic types (time-triggered and event-triggered) in the same network	TTCAN can be used on a modular satellite due to its reliability and determinism, but its low data rate only allows an application as housekeeping and satellite control data bus, a secondary bus with high data rate will additionally be needed for the P/L data traffic.	SpaceWire is a space-qualified data bus standard with high data rate but it is primarily developed as a P/L bus. Missing determinism and fault containment make it only be adaptable on modular satellites in combination with a reliable housekeeping and	FireWire is obviously not suitable for modular satellites since several critical requirements to the on-board data communication system are not met by this standard.	TTEthernet seems to be the most suitable solution for an application on modular spacecraft, since it meets several critical requirements of the modular systems. However, additional complex concepts to make a time-triggered communication system reconfigurable	EtherCat is primarily developed as Ethernet-based fieldbus in the process und automation area and provides many advantages due to jitters, synchronization and low costs compared to classical fieldbus. Due to the missing flexibility and reconfigurability, it

Features / bus standards	MIL-STD-1553	TTP/C	TTCAN	SpaceWire	IEEE-1394B	TTEthernet	EtherCat
	extendibility and limited node number of the network are additional counterarguments of this standard.	is a great advantage. However, the defined maximum node number of 64 nodes is a limit for large modular space systems.		satellite control data bus.		need to be developed and implemented on top	is not suitable to be deployed on modular satellites.

References

[1] G. P. Sudhakar, “ISRO: 104 Satellites in 1 go,” in *Vidyaniketan Journal of Management Research*, pp. 74–94.

[2] A. Nazeer, “Research on PSLV-C37 Launcher by ISRO,” *International Journal of Science and Research (IJSR)*, Volume 7 Issue 11, pp. 17–21, 2018, doi: 10.21275/ART20192415.

[3] G. Curzi, D. Modenini, and P. Tortora, “Large Constellations of Small Satellites: A Survey of Near Future Challenges and Missions,” *Aerospace*, vol. 7, no. 9, p. 133, 2020, doi: 10.3390/aerospace7090133.

[4] C. C. Helms, “A Survey of Launch Services 2016-2020,” in *AIAA Propulsion and Energy 2020*, Aug. 24, 2020.

[5] G. E. Musgrave, Ed., *Safety design for space systems*. Oxford: Elsevier, 2009. [Online]. Available: http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10286088

[6] *Computer generated model of the object population in the geosynchronous region.* [Online]. Available: https://www.orbitaldebris.jsc.nasa.gov/photo-gallery/ (accessed: Oct. 4, 2020).

[7] D. Tham, “Reconfiguration of time triggered data network on a modular satellite during run-time,” presented at Small Satellites, System and Services Symposium (4S). Sorrento, Italia, May. 28, 2018.

[8] D. Tham and F. Baumann, “Eine rekonfigurierbare Datenbusarchitektur für modulare Satelliten,” presented at Deutsche Luft- und Raumfahrtkongress 2018. Friedrichshafen, Germany, Sep. 4, 2018.

[9] D. Tham and P. Wüstenberg, “A reconfigurable communication architecture for modular satellites,” Presented at the 69th International Astronautical Congress. Bremen, Germany, Oct. 1, 2018.

[10] T. Meschede, *iBOSS Pictures Collection*. Used with the friendly permission from the author.

[11] *Onboard Computers and Data Handling.* [Online]. Available: https://www.esa.int/Enabling_Support/Space_Engineering_Technology/Onboard_Computers_and_Data_Handling/Onboard_Computers_and_Data_Handling (accessed: Oct. 4, 2020).

[12] *Architectures of Onboard Data Systems.* [Online]. Available: http://www.esa.int/Enabling_Support/Space_Engineering_Technology/Onboard_Computers_and_Data_Handling/Architectures_of_Onboard_Data_Systems (accessed: Oct. 4, 2020).

[13] D. Tham, “iBOSS - Intelligentes Baukastensystem für das On-Orbit-Satelliten Servicing: Satelliteninterne Kommunikation,” presented at Kolloquium Raumfahrttechnik TU Berlin 2013. Berlin, Germany, 2013.

[14] The Consultative Committee for Space Data Systems (CCSDS), *Wireless Network Communications Overview for Space Mission Operations*. Informational Report.

[15] R. Amini, “Wireless Communication onboard Spacecraft,” Ph.D. Dissertation, Fac. of Aerospace Eng., Delft Univ. of Techn., Delft, the Netherlands, 2016.

[16] B. Palmintier, C. Kitts, P. Stang, and M. Swartwout, “A Distributed Computing Architecture for Small Satellite and Multi-Spacecraft Missions,” presented at the 16th Annual AIAA/USU Conference on Small Satellites. Logan, Utah, USA, 2002.

[17] F. Scheler and W. Schroeder-Preikschat, “Time-Triggered vs. Event-Triggered: A matter of configuration?,” in *ITG FA 6.2 Workshop on Model-Based Testing, GI/ITG Workshop on Non-Functional Properties of Embedded Systems, 13th GI/ITG Conference Measuring, Modelling, and Evaluation of Computer and Communications*, 2006, pp. 1–6.

[18] A. Murshed, “Scheduling Event-Triggered and Time-Triggered Applications with Optimal Reliability and Predictability on Networked Multi-Core Chips,” PH.D. Dissertation, Naturwissenschaftlich-Technischen Fakultät, Universität Siegen, 2018.

[19] A. Albert, “Comparison of Event-Triggered and Time-Triggered Concepts with Regard to Distributed Control Systems,” *Embedded World*, vol. 2004, pp. 235–252, 2004.

[20] D.A. Gwaltney and J. M. Briscoe, “Comparison of Communication Architectures for Spacecraft Modular Avionics Systems: NASA Technical Memorandium,” NASA, Mashall Space Flight Center, Alabama, USA.

[21] T. Sauter, “The Three Generations of Field-Level Networks—Evolution and Compatibility Issues,” *IEEE Transactions on Industrial Electronics*, vol. 57, no. 11, pp. 3585–3595, 2010, doi: 10.1109/TIE.2010.2062473.

[22] G. S. Hura and M. Singhal, *Data and computer communications: Networking and internetworking*. Boca Raton, FL: CRC Press, 2001. [Online]. Available: http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10142637

[23] NASA, Goddard Space Flight Center, “On-Orbit Satellite Servicing Study: Project Report”, 2010.

[24] D. Rossetti, B. Keer, J. Panek, B. Ritter, B. B. Reed, and F. Cepollina, “Spacecraft Modularity for Serviceable Satellites,” in *AIAA SPACE 2015 Conference*, Pasadena, California, USA, Aug. 31, 2015.

[25] J. Esper, “Modular, Adaptive, Reconfigurable Systems: Technology for Sustainable, Reliable, Effective, and Affordable Space Exploration,” in *AIP Conference Proceedings*, Albuquerque, New Mexico (USA), Feb. 2005, pp. 1033–1043.

[26] S. J. Leete, “Satellite servicing in mission design studies at the NASA GSFC,” in *2003 IEEE Aerospace Conference Proceedings (Cat. No.03TH8652)*, 2003, 8_3957-8_3966.

[27] A. Siddiqi, “Reconfigurability in Space Systems: Architecting Framework and Case Studies: Ph. D. Dissertation,” Dep. of Aeronautics and Astronautics, MIT, Cambridge, USA, May 2006.

[28] S. Speretta, L. M. Reyneri, C. Sansoé, M. Tranchero, C. Passerone, and D. Del Corso, “Modular architecture for satellites,” in *58th International Astronautical Congress 2007, Hyderabad, India, Sept. 24-28, 2007*, pp. 24–28.

[29] L. M. Reyneri *et al.,* “Design Solutions for Modular Satellite Architectures,” in *Aerospace Technologies Advancements*, T. T., Ed.: InTech, 2010.

[30] M. R. Mughal, D. Roascio, and L. M. Reyneri, "Data communication bus interface for modular architecture of small satellites," in *2012 IEEE First AESS European Conference on Satellite Telecommunications (ESTEL)*, 2012, pp. 1–6.

[31] M. R. Mughal, L. M. Reyneri, J. C. De Los Rios, and H. Ali, "Development of optical wireless transceiver for data communication in small satellites," in *2012 IEEE First AESS European Conference on Satellite Telecommunications (ESTEL)*, 2012, pp. 1–6.

[32] M. R. Mughal, L. M. Reyneri, and D. Del Corso, "Implementation of Inter and Intra Tile Optical Data Communication for NanoSatellites," in *Proceeding International Conference on Space Optical Systems and Applications (ICSOS) 2012,* Corsica, France, Oct. 9-12, 2012.

[33] L. Hill *et al.,* "The Market for Satellite Cellularization: A historical view of the impact of the Satlet morphology on the space industry," in *AIAA SPACE 2013 Conference and Exposition*, San Diego, CA, Oct. 9, 2013.

[34] D. Barnhart *et al.,* "A Market for Satellite Cellularization: A First Look at the Implementation and Potential Impact of Satlets," presented at AIAA Space 2013 Conference & Exposition. San Diego, California, USA, Sep. 2013.

[35] D. Barnhart, L. Hill, M. Turnbull, and P. Will, "Changing Satellite Morphology through Cellularization," in *AIAA SPACE 2012 Conference & Exposition*, Pasadena, California, 2012.

[36] A. A. Kerzhner *et al.,* "Architecting Cellularized Space Systems using Model-Based Design Exploration," in *AIAA SPACE 2013 Conference and Exposition*, San Diego, CA, Oct. 9, 2013.

[37] D. Barnhart *et al.,* "Phoenix Program Status - 2013," in *AIAA SPACE 2013 Conference and Exposition*, San Diego, CA, Oct. 9, 2013.

[38] T. Jaeger, W. Mirczak, and B. Crandall, "Cellularized Satellites - Initial Experiments and the Path forward," 31st Space Symposium. Colorado Springs, Colorado, USA, Apr. 13, 2016.

[39] T. Jaeger and W. Mirczak, "Phoenix and the New Satellite Paradigm Created by HISat," presented at the 28th Annual AIAA/USU Conference on Small Satellites. Logan, Utah, USA, Aug. 5, 2014.

[40] T. Jaeger and W. Mirczak, "Satlets - The Building Blocks of Future Satellites - And Which Mold Do You Use ?," in *AIAA SPACE 2013 Conference, September 10-12, 2013, San Diego, California, USA*.

[41] T. Jaeger, W. Mirczak, and B. Crandall, "Cellularized Satellites - A Small Satellite Instantiation that Provides Mission and Space Access Adaptability," presented at 2016 SmallSat Conference. Logan, Utah, USA, Aug. 9, 2016.

[42] *Satlets: crazy idea or ingenious concept? This week's test on ISS will offer clues.* [Online]. Available: https://spacenews.com/Satlets-crazy-idea-or-ingenious-concept-this-weeks-test-on-iss-will-offer-clues/ (accessed: Oct. 4, 2020).

[43] L. Hill *et al.,* "A Further Look at Potential Impact of Satlets on Design, Production, and Cost of Satellite Systems," presented at 28th Annual AIAA/USU Small Satellite Conference 2014. Logan, Utah, USA, Aug. 4 2014.

[44] S. Scholes, *Now Entering Orbit: Tiny Lego-like Modular Satellites: Space is getting closer, thanks to small, cheap "Satlets" that network themselves to solve problems in flight.* [Online]. Available: https://www.wired.com/story/now-entering-orbit-tiny-lego-like-modular-satellites/ (accessed: Oct. 4, 2020).

[45] G. Krebs, *PODSAT 1.* [Online]. Available: https://space.skyrocket.de/doc_sdat/podsat.htm (accessed: Oct. 4, 2020).

[46] G. Krebs, *eXCITe (PTB 1).* [Online]. Available: https://space.skyrocket.de/doc_sdat/excite.htm (accessed: Oct. 4, 2020).

[47] G. Krebs, *SIMPL (HISat).* [Online]. Available: https://space.skyrocket.de/doc_sdat/simpl.htm (accessed: Oct. 4, 2020).

[48] iBOSS Consortium, "Final Technical Report of the Cooperation Project iBOSS-3," 2018.

[49] iBOSS Consortium, "Final Technical Report of the Cooperation Project iBOSS-2," Jan. 20, 2016.

[50] iBOSS Consortium, "Abschlussbericht der TU Berlin zum Kooperationsvorhaben iBOSS - Intelligenter Baukasten für das On-Orbit-Satelliten Servicing -," Aug. 31, 2012.

[51] iBOSS Consortium, *iSSI – intelligent Space System Interface.* [Online]. Available: https://www.iboss.space/wp-content/uploads/2018/09/iSSI-Brochure2017-DLR-SpaceAdministration.pdf (accessed: Oct. 4, 2020).

[52] P. Moubarak and P. Ben-Tzvi, "Modular and reconfigurable mobile robotics," *Robotics and Autonomous Systems*, vol. 60, no. 12, pp. 1648–1663, 2012, doi:10.1016/j.robot.2012.09.002.

[53] A. Brunete, A. Ranganath, S. Segovia, J. P. de Frutos, M. Hernando, and E. Gambao, "Current trends in reconfigurable modular robots design," *International Journal of Advanced Robotic Systems*, vol. 14, no. 3, 172988141771045, 2017, doi: 10.1177/1729881417710457.

[54] M. Yim, K. Roufas, D. Duff, Y. Zhang, C. Eldershaw, and S. Homans, "Modular Reconfigurable Robots in Space Applications," *Autonomous Robots*, vol. 14, 2/3, pp. 225–237, 2003, doi: 10.1023/A:1022287820808.

[55] A. Avižienis, J.-C. Laprie, and B. Randell, "Fundamental Concepts of Dependability," Computer Science, University of Newcastle upon Tyne, 2001.

[56] R. F. Mendoza Garcia, K. Stoy, D. J. Christensen, and A. Lyder, "A Self-Reconfigurable Communication Network for Modular Robots," in *The Proceedings of First International Conference on Robot Communication and Coordination*, Athens, Greece, Oct. 2007.

[57] E. Yoshida, T. Arai, M. Yamamoto, J. Ota, and D. Kurabayashi, "Evaluating the efficiency of local and global communication in distributed mobile robotic systems," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS '96*, Osaka, Japan, Nov. 1996, pp. 1661–1666.

[58] T. M. Nguyen, "Future Satellite System Architectures and Practical Design Issues: An Overview," in *Satellite Systems - Design, Modeling, Simulation and Analysis [Working Title]*: IntechOpen, 2020.

[59] H. J. Kramer, *EnMAP (Environmental Monitoring and Analysis Program).* [Online]. Available: https://directory.eoportal.org/web/eoportal/satellite-missions/e/enmap (accessed: Oct. 5, 2020).

[60] H. J. Kramer, *SmallGeo.* [Online]. Available: https://directory.eoportal.org/web/eoportal/satellite-missions/s/smallgeo (accessed: Oct. 4, 2020).

[61] OHB, *Satellite Platforms: Medium and Large Platforms.* [Online]. Available: https://www.ohb.de/fileadmin/ohb/Downloads/190603_OHB-System_Satellite_Platforms_2019-05.pdf (accessed: Oct. 4, 2020).

[62] D. Tham, "Verteilte Datenkommunikation in modularen Satelliten," presented Kolloquium Raumfahrttechnik TU Berlin 2017, July, 2017.

[63] J. Rushby, "Bus Architectures for Safety-Critical Embedded Systems," in *Lecture Notes in Computer Science, Embedded Software*, G. Goos, J. Hartmanis, J. van Leeuwen, T. A. Henzinger, and C. M. Kirsch, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 306–323.

[64] I. Arruego *et al.,* "OWLS: a ten-year history in optical wireless links for intra-satellite communications," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 9, pp. 1599–1611, 2009.

[65] J. Kim and J. Han, "Development of a distributed OBDH system with MIL-STD-1553B and its application," in *2000 IEEE Aerospace Conference. Proceedings (Cat. No.00TH8484)*, 2000, 243-248 vol.5.

[66] R. Killough, "Integrating CCSDS and MIL-STD-1553: What you should know," in *Proceedings, IEEE Aerospace Conference*, 2002, p. 4.

[67] *ECSS-E-ST-50-13C - Interface and communication protocol for MIL-STD-1553B data bus onboard spacecraft*, ECSS-E-ST-50-13C, Nov. 15, 2008.

[68] Phoenix Logistics LLC, *Data bus overview.* [Online]. Available: http://www.phxlogistics.com/data-bus/overview (accessed: Oct. 4, 2020).

[69] W. Elemenreich and R. Ipp, "Introduction to TTP/C and TTP/A," presented at Proceedings of the Workshop on Time-Triggered and Real-Time Communication Systems, January, 2003. Accessed: Oct. 5, 2020. [Online]. Available: https://www.researchgate.net/publication/215499228_Introduction_to_TTPC_and_TTPA

[70] H. Kopetz and G. Bauer, "The time-triggered architecture," *Proceedings of the IEEE*, vol. 91, no. 1, pp. 112–126, 2003.

[71] *Time-Triggered Protocol TTP/C*, D–032–S–10–028, Nov. 11, 2003.

[72] S. Weiß, S. Kapitola, N. Korn, S. Trowitzsch, and K. Brieß, Eds., *BEESAT-2 to BEESAT-4: In flight demonstration of attitude control to GPS based positioning and orbit determination*. presented at 4S Symposium 2016. Valleta, Malta, May. 30 - Jun. 3, 2016.

[73] A. Emrich, "CAN application in avionics," Final Report of Omnisys Instruments AB, Jul. 2001.

[74] ESA, *ESA's SMART-1: testing solar electric propulsion and studying the Moon.* [Online]. Available: https://www.esa.int/ESA_Multimedia/Images/2002/04/ESA_s_SMART-1_testing_solar_electric_propulsion_and_studying_the_Moon (accessed: Oct. 5, 2020).

[75] T. Führer, B. Müller, F. Hartwich, and R. Hugel, “Time Triggered CAN (TTCAN),” in *SAE Technical Paper Series*, 2001.

[76] M. Short and M. J. Pont, “Fault-Tolerant Time-Triggered Communication Using CAN,” *IEEE Trans. Ind. Inf.*, vol. 3, no. 2, pp. 131–142, 2007, doi: 10.1109/TII.2007.898477.

[77] Stock Flight Systems, *Interface specification for airborne CAN applications; Stock Flight Systems.* [Online]. Available: https://www.stockflightsystems.com/tl_files/downloads/canaerospace/canas_17.pdf (accessed: Oct. 5, 2020).

[78] RS Flightsystems: *ARINC-825 CANaerospace Interface (CANflight).* [Online]. Available: https://www.rs-flightsystems.com/product-page/arinc-825-canaerospace-interface-canflight (accessed: Oct. 5, 2020).

[79] *ECSS-E-ST-50-12C – SpaceWire – Links, nodes, routers and networks*, ECSS-E-ST-50-12C, May. 15, 2019.

[80] *ECSS-E-ST-50-11C – SpaceFibre – Very high-speed serial link*, ECSS-E-ST-50-11C, ECSS, May. 15, 2019.

[81] T. Takashima, E. Ogawa, K. Asamura, and M. Hikishima, “Design of a mission network system using SpaceWire for scientific payloads onboard the Arase spacecraft,” *Earth Planets Space*, vol. 70, no. 1, 2018, doi: 10.1186/s40623-018-0839-z.

[82] *Space Cables for SpaceWire Applications.* [Online]. Available: https://www.gore.com/products/space-cables-for-spacewire-applications (accessed: Oct. 5, 2020).

[83] S. Parkes, A. F. Florit, A. G. Villafranca, D. McLaren, and C. McClements, “SpaceFibre,” presented at Flight Software Workshop 2015, 2015.

[84] H. Kopetz, A. Ademaj, P. Grillinger, and K. Steinhammer, “The time-triggered Ethernet (TTE) design,” in *Eighth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC'05)*, 2005, pp. 22–33.

[85] A. T. Loveless, “On TTEthernet for Integrated Fault-Tolerant Spacecraft Networks,” in *AIAA SPACE 2015 Conference and Exposition*, Pasadena, California, Aug. 2015.

[86] TTTech, *Flight & Rugged Hardware for Space.* [Online]. Available: https://www.tttech.com/products/aerospace-space/flight-rugged-hardware/space-ceramic/ (accessed: Oct. 6 2020).

[87] *NASA's Orion Spacecraft.* [Online]. Available: https://www.esa.int/ESA_Multimedia/Images/2014/12/Orion_NASA_Grafik (accessed: Oct. 6 2020).

[88] H. Kopetz, *Real-time systems: Design principles for distributed embedded applications,* 2nd ed. New York, Dordrecht, Heidelberg: Springer, 2011.

[89] C.-J. Hou and K. G. Shin, “Allocation of periodic task modules with precedence and deadline constraints in distributed real-time systems,” *IEEE Trans. Comput.*, vol. 46, no. 12, pp. 1338–1356, 1997, doi: 10.1109/12.641934.

[90] M. Abuteir and R. Obermaisser, “Scheduling of rate-constrained and time-triggered traffic in multi-cluster TTEthernet systems,” in *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)*, Cambridge, United Kingdom, Jul. 2015 - Jul. 2015, pp. 239–245.

[91] G. C. Buttazzo, *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications,* 3rd ed. Boston, MA: Springer US, 2011.

[92] L. Wisniewski, “New methods to engineer and seamlessly reconfigure time triggered Ethernet based systems during runtime based on the PROFINET IRT example,” Dissertation, Fakultät für Elektrotechnik und Informationstechnik, Otto-von-Guericke-Universität Magdeburg.

[93] P. Eles, A. Doboli, P. Pop, and Z. Peng, “Scheduling with bus access optimization for distributed embedded systems,” *IEEE Trans. VLSI Syst.*, vol. 8, no. 5, pp. 472–491, 2000, doi: 10.1109/92.894152.

[94] R. Tyagi and S. K. Gupta, “A Survey on Scheduling Algorithms for Parallel and Distributed Systems,” in *Advances in Intelligent Systems and Computing, Silicon Photonics & High Performance Computing*, A. Mishra, A. Basu, and V. Tyagi, Eds., Singapore: Springer Singapore, 2018, pp. 51–64.

[95] *Wireless Microsystems.* [Online]. Available: https://www.ipms.fraunhofer.de/en/research-development/wireless-microsystems.html (accessed: Oct. 5, 2020).

[96] TTTech Computertechnik AG, *TTESwitch OBC HiRel User Manual*, 2018.

[97] *ECSS-M-ST-10C - Space project management*, Mar. 6, 2009.

[98] TTTech Computertechnik AG, *TTE-Plan User Manual*. Version 5.2.6, 2018.

[99] “Attitude and Orbit Control Systems (AOCS),” in *Lecture Notes in Computer Science, Software Frameworks and Embedded Control Systems*, G. Goos, J. Hartmanis, J. van Leeuwen, and A. Pasetti, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 17–28.

[100] T. Steinbach, F. Korf, and T. C. Schmidt, “Comparing time-triggered Ethernet with FlexRay: An evaluation of competing approaches to real-time for in-vehicle networks,” in *2010 IEEE International Workshop on Factory Communication Systems Proceedings*, Nancy, France, May. 2010, pp. 199–202.

[101] S. Farritor and J. Zhang, “A modular robotic infrastructure to support planetary surface operations,” in *Space 2000 Conference and Exposition*, Long Beach,CA,U.S.A, Sep. 19,2000.

[102] G. S. H. Matthew D. Hancher, “A Modular Robotic System with Applications to Space Exploration,” in *2nd IEEE International Conference on Space Mission Challenges for Information Technology (SMC-IT'06)*, Pasadena, CA, USA, Jul. 2006, pp. 125–132.