Study on real-time industrial control networks

Wu, Xuepei

2019

Wu, X. (2019). Study on real-time industrial control networks. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/90139>

<https://doi.org/10.32657/10220/48429>

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

SINGAPORE

Study on Real-Time Industrial Control Networks

Wu Xuepei

School of Electrical & Electronic Engineering

2019

Study on Real-Time Industrial Control Networks

Wu Xuepei

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Doctor of Philosophy

2019

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

7 Jan 2019

DATE

WU

WU XUEPEI

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

6 Jan. 2019

DATE

PROF. XIE LIHUA

Authorship Attribution Statement

This thesis contains material from six papers published in the following peer-reviewed journals and conference proceedings where I was the first and/or corresponding author.

The work in Chapter 3 is published as

- X. Wu and L. Xie, "Towards an IIoT-based architecture for baggage handling systems", *Journal of Communications*, vol. 12, no. 8, pp. 475-481, 2017. DOI: 10.12720/jcm.12.8.475-481.
- X. Wu, L. Xie and F. Lim, "EtherCAT-enabled next generation baggage handling systems", in *Proceedings of the 18th IEEE Conference on Emerging Technologies & Factory Automation*, Cagliari, Italy, pp. 1-6, 2013. DOI: 10.1109/ETFA.2013.6648067.

The work in Chapters 4 and 5 is published as

- X. Wu and L. Xie, "Performance evaluation of industrial Ethernet protocols for networked control application", *Control Engineering Practice*, vol. 84, pp. 208-217, 2019. DOI: 10.1016/j.conengprac.2018.11.022.

- X. Wu and L. Xie, “End-to-end delay evaluation of industrial automation systems based on EtherCAT”, in *Proceedings of the 42nd IEEE Conference on Local Computer Networks (LCN)*, Singapore, pp. 70-77, 2017. DOI: 10.1109/LCN.2017.14.
- X. Wu and L. Xie, “On the wireless extension of EtherCAT networks”, in *Proceedings of the 42nd IEEE Conference on Local Computer Networks (LCN)*, Singapore, pp. 235-238, 2017. DOI: 10.1109/LCN.2017.15.
- X. Wu, L. Xie and F. Lim, “Network delay analysis of EtherCAT and PROFINET IRT protocols”, in *Proceedings of the 40th Annual Conference of the IEEE Industrial Electronics Society (IECON)*, Dallas, TX, USA, pp. 2597-2603, 2014. DOI: 10.1109/IECON.2014.7048872.

For the publications “EtherCAT-enabled next generation baggage handling systems” and “Network delay analysis of EtherCAT and PROFINET IRT protocols”, the contributions of the co-authors are as follows:

- I prepared the manuscript drafts.
- Prof. L. Xie identified the research scope and direction for the work in the publications. Additionally, Prof. L. Xie provided guidance on all manuscripts including reviews and revisions.

- Mr. F. Lim provided the initial direction and the business context of the research project.

For the publications “Towards an IIoT-based architecture for baggage handling systems”, “Performance evaluation of industrial Ethernet protocols for networked control application”, “End-to-end delay evaluation of industrial automation systems based on EtherCAT” and “On the wireless extension of EtherCAT networks”, the contributions of the co-authors are as follows:

- I prepared the manuscript drafts.
- Prof. L. Xie identified the research scope and direction for the work in the publications. Additionally, Prof. L. Xie provided guidance on all manuscripts including reviews and revisions.

15 May 2019

DATE

WU XUEPEI

Acknowledgments

Undertaking doctoral study has been a truly life-changing experience for me and it would never have been possible to take this work to completion without the guidance and support that I received from many people.

Firstly, I would like to express my sincere appreciation and respect to my supervisor, Professor Xie Lihua, for his professional guidance and valuable suggestions throughout my time as his student. I could not imagine having had a better advisor and mentor for my research work.

I owe the deepest gratitude to my family for their encouragement and love. My wife and my parents have been extremely supportive of me throughout the entire study and have made countless sacrifices to help me get to this point.

I also thank the staff and students in the Internet of Things Laboratory of Nanyang Technological University for their help.

Lastly, I gratefully acknowledge the funding received from the Industrial Postgraduate Programme of the Singapore Economic Development Board (grant number S11-1669-IPP).

Abstract

Boosted by business trends such as Industry 4.0, Industrial Internet of Things (IIoT) solutions such as real-time Ethernet and wireless technologies have been increasingly deployed in the industrial automation sector. Higher fluctuation and less predictability in terms of customer demand promoted by Industry 4.0 can be overcome by improving the flexibility of the system via network-based control. However, closing control loops over a shared communication network means that the system performance can be highly dependent on the quality of the communication services which transfer control variables (e.g., set-point and measurement) between controllers, sensors and actuators. It is of great importance to design a common networking solution to meet real-time (and wireless) requirements in networked control systems, and maintain the backward compatibility to distributed control systems where soft real-time communication is sufficient.

This thesis firstly reviews existing Ethernet-like IIoT technologies and suggests a generic communication model with multiple data channels. The scheduling of data transmissions in different channels is presented, and wireless extension scheme is suggested to integrate wireless segments into the network. The communication model is defined so that a communication backbone can be established for all types of network-based control applications, such as motion control and path tracking of automated guided vehicles (AGVs). Next, the quality of service (QoS) of the hybrid network, including cycle time, end-to-end delay and communication reliability is analyzed. Finally, a typical industrial automation system with motion control and remote AGV tracking applications is presented and discussed to demonstrate the performance of the networking scheme proposed in the thesis where a hybrid network possessing various real-time characteristics is able to satisfy multiple types of control applications described in the case study. The benefits of the solution, i.e., flexibility and scalability, can be significant to fulfil dynamic market demands in the next generation automation systems powered by Industry 4.0.

Contents

Acknowledgments	i
Abstract	ii
List of Figures	vii
List of Tables	ix
List of Abbreviations	x
1 Introduction	1
1.1 Motivation and Objectives	1
1.2 Major Contribution of the Thesis	3
1.3 Organization of the Thesis	4
2 Literature Review	6
2.1 Real-time Ethernet	6
2.2 Wireless Extension of Real-time Ethernet	10
2.3 Performance Evaluation on IIoT Networks	11
2.4 Network-based Control over IIoT Networks	13
3 Networked Control System Overview	15
3.1 System Architecture and Configuration	15
3.1.1 System Architecture and Components	15
3.1.2 System Configurations	18
3.2 Communication Network and Topology	19
3.2.1 Overview of Existing Ethernet-based Technologies	19
3.2.2 Comparison between Industrial Ethernet Networks	22
3.2.3 Industrial Wireless Networks	24

3.2.4	Topology in Hybrid Networks	26
3.3	Task Model	26
3.4	Problem Statement	27
4	Design of Real-time Hybrid IIoT Networks	30
4.1	Conceptual Design	30
4.2	Frame Structure	31
4.2.1	Non-real-time Frame	32
4.2.2	Real-time Frame	32
4.2.3	Hard Real-time Frame	34
4.2.4	IEEE 802.11	35
4.3	Communication Scheduling	36
4.3.1	Bus Cycle	36
4.3.2	Guaranteed Channel	36
4.3.3	Best-effort Channel	37
4.3.4	Reserved Channel	39
4.4	Wireless Extension	39
4.4.1	Distributed Coordination Function Mode	39
4.4.2	Point Coordination Function Mode	40
4.4.3	Extension at Physical Layer	41
4.4.4	Extension at Data-link Layer	42
4.4.5	Extension at Application Layer	45
4.5	Conclusion	47
5	Analysis of Communication Quality of Service	48
5.1	Cycle Time	48
5.1.1	Guaranteed Phase	49
5.1.2	Best-effort Phase	50
5.1.3	Reserved Phase	52
5.2	End-to-end Delay on OSI Model Layers	52
5.2.1	Network Delay	53
5.2.2	Stack Delay	53

5.2.3	Application Delay	56
5.3	Sensor-to-Controller Network Delay	56
5.3.1	Network Delay of Wired Network	56
5.3.2	Network Delay of Wireless Network	58
5.4	Controller-to-Actuator Network Delay	59
5.4.1	Network Delay of Wired Network	59
5.4.2	Network Delay of Wireless Network	60
5.5	Component Delay	61
5.5.1	Controller Delay	61
5.5.2	Bridge Delay	62
5.5.3	Gateway Delay	62
5.6	Delay Model	62
5.7	Communication Reliability	63
5.7.1	Reliability on Wired Network	63
5.7.2	Reliability on Wireless Network	64
5.8	Conclusion	65
6	Evaluation on Communication Quality of Service	66
6.1	Cycle Time	66
6.1.1	Performance Comparison between SFP and IFP	66
6.1.2	Cycle Time of Wireless Segment	68
6.2	Network Delay	68
6.2.1	Performance Comparison between SFP and IFP	68
6.2.2	Network Delay of Wireless Segment	72
6.3	Reliability	75
6.4	Conclusion	75
7	Integrated Network-based Control — A Case Study	77
7.1	System Integration	77
7.2	Evaluation Strategy	79
7.3	Communication Quality of Service	81
7.4	Networked Control via Guaranteed Channel	82

7.4.1	Servo Control Application	82
7.4.2	Cost Functions	86
7.4.3	Quality of Control	86
7.5	Remote Tracking over Hybrid Network	88
7.5.1	AGV Kinematic Model	88
7.5.2	Local Path Tracking Control	89
7.5.3	Remote Path Tracking Control	94
7.5.4	Application Parameters	94
7.5.5	Cost Functions	95
7.5.6	Quality of Control	96
7.6	Conclusion	99
8	Conclusion and Recommendations	100
8.1	Conclusion	100
8.2	Recommendations for Further Research	101
	List of Publication	103
	References	105

List of Figures

1.1	A typical industrial automation system powered by Industrial IoT	2
2.1	Layered communication architecture for performance class (a) NRT; (b) RT; (c) HRT	8
3.1	System architecture of industrial control systems	17
3.2	An Ethernet POWERLINK cycle	20
3.3	Schematic of an EtherCAT slave device	22
3.4	Real-time classes offered by PROFINET	23
3.5	Network topology of a typical NCS	26
4.1	Data frame structure for real-time class (a) NRT; (b) RT; (c) HRT; . . .	33
4.2	IEEE 802.11 DATA frame structure	36
4.3	Bus cycle incorporating guaranteed, best effort and reserved phases . . .	36
4.4	Space-time diagram of transmissions using (a) IFP (with slipstreaming effect); (b) SFP.	38
4.5	Medium access of DCF mode	41
4.6	Medium access of PCF mode	41
4.7	System structure of bridge-based wireless extension	42
4.8	Space-time diagram of DCF-based wireless extension	44
4.9	Space-time diagram of PCF-based wireless extension	45
4.10	System structure of gateway-based wireless extension	46
5.1	Transmissions in wireless segment during BE phase	51
5.2	Gantt chart of a wired network using (a) IFP; (b) SFP	54
5.3	Gantt chart of a hybrid network	55

6.1	Efficiency factor of SFP relative to IFP as a function of f_p and N_{dev} . . .	67
6.2	Cycle time of wireless segment as a function of f_p and N_{wln}	69
6.3	Efficiency factor of SFP relative to IFP on S-C path as a function of f_p and n	70
6.4	Efficiency factor of SFP relative to IFP on C-A path as a function of f_p and n	71
6.5	Network delay of S-C link as a function of f_p and N_{wln}	73
6.6	Network delay of C-A link as a function of f_p and N_{wln}	74
6.7	Efficiency factor of SFP relative to IFP as a function of f_p and N_{dev} . . .	76
7.1	System topology	78
7.2	Evaluation process on system performance	80
7.3	Probability distribution of end-to-end delays for HRT_IFP: (a) τ_{sc} ; (b) τ_{ca} . . .	83
7.4	Probability distribution of end-to-end delays for HRT_HFP: (a) τ_{sc} ; (b) τ_{ca} . .	84
7.5	Probability distribution of end-to-end delays for NRT: (a) τ_{sc} ; (b) τ_{ca} . .	85
7.6	System response of the servo system when $N_{dev} = 50$	87
7.7	AGV kinematics	89
7.8	Local path tracking control flow	89
7.9	Algorithm of computing $s(k)$	91
7.10	Remote path tracking control flow	94
7.11	Actual path of the AGV	97
7.12	Closest distance between desired and actual path	98

List of Tables

2.1	Comparison on top-5 industrial Ethernet.	10
3.1	Comparison among system configurations.	18
3.2	Comparison between wired IIoT technologies.	24
3.3	Comparison between wireless IIoT networks.	25
5.1	Communication parameters in IEEE 802.11g	52
7.1	Simulation parameters	78
7.2	Cycle time performance ($N_{dev} = 10, N_{wln} = 10$)	82
7.3	Comparison on end-to-end delays ($N_{dev} = 10, N_{wln} = 0$)	82
7.4	QoC results of the motion control system	87
7.5	Simulation parameters for the AGV application	95
7.6	Comparison on QoC between local and remote tracking	97

List of Abbreviation

AC	Access Category
ACK	Acknowledgement
AGV	Automated Guided Vehicle
AP	Access Point
ASIC	Application-Specific Integrated Circuit
BE	Best-Effort
BER	Bit Error Rate
C-A	Controller-to-Actuator
CAN	Controller Area Network
CAP	Contention Access Period
CFP	Contention-Free Period
CIP	Common Industrial Protocol
CNC	Computerized Numerical Control
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CSMA/CA	Carrier-Sense Multiple Access with Collision Avoidance
CSMA/CD	Carrier-Sense Multiple Access with Collision Detection
DC	Distributed Clock
DCF	Distributed Coordination Function
DCS	Distributed Control System
DFP	Dynamic Frame Packing
DIFS	DCF Inter-Frame Spacing
DPRAM	Dual-Port Random-Access Memory
DSSS	Direct Sequence Spread Spectrum
EDF	Earliest Deadline First
FCS	Frame Check Sequence
FDL	Fieldbus Data Link
FMMU	Fieldbus Memory Management Unit
FPGA	Field-Programmable Gate Array
GTD	Guaranteed
GTS	Guaranteed Time Slot
HRT	Hard Real-Time

HTTP	HyperText Transfer Protocol
IAE	Integral Absolute Error
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IFP	Individual Frame Protocol
IIoT	Industrial Internet of Things
IP	Internet Protocol
IRT	Isochronous Real-Time
ISM	Industrial, Scientific, and Medical
LR-WPAN	Low-Rate Wireless Personal Area Network
MAC	Media Access Control
MPC	Model Predictive Control
MTU	Maximum Transfer Unit
NCS	Networked Control System
NRT	Non-Real-Time
OS	Operating System
PCF	Point Coordination Function
PDU	Protocol Data Unit
PHY	Physical Layer
PID	Proportional-Integral-Derivative
PLC	Programmable Logic Controller
PTCP	Precision Transparent Clock Protocol
PTP	Precision Time Protocol
QoC	Quality of Control
QoS	Quality of Service
RF	Radio Frequency
RT	Real-Time
S-C	Sensor-to-Controller
SD	Standard Deviation
SFP	Summation-Frame Protocol
SIFS	Short Inter-Frame Spacing
SoC	System on Chip
TCP	Transmission Control Protocol
TDMA	Time-Division Multiple Access
TSN	Time-Sensitive Networking
UDP	User Datagram Protocol
VLAN	Virtual Local Area Network
WISA	Wireless Interface for Sensors and Actuators
WLAN	Wireless Local Area Network

Chapter 1

Introduction

1.1 Motivation and Objectives

With recent initiatives from various organizations and governments such as Industry 4.0 [1], Made in China 2025 and Cloud Manufacturing [2], industrial automation sector is moving towards product customization and personalization. Business parameters are of more fluctuation and less predictability as compared to the current ones. Factory automation systems are required to be self-configurable to adapt to fast-changing market needs. Consequently, there is increasing demand for better operational flexibility and efficiency on control systems, without increasing system complexity. An example of an Industry 4.0-powered system is illustrated in Figure 1.1 (Source: [3]).

Controlling these automation processes using state-of-the-art techniques such as *Big Data* and artificial intelligence over Industrial IoT networks can be one of the solutions. Networked control, where the control loops are closed over a shared communication channel rather than dedicated and independent connections, offers a number of unique advantages in comparison with conventional distributed systems. Firstly, connecting sensors, actuators and controllers via data networks allows information to be shared among devices over a large physical space to enable distributed or centralized intelligence [4]. Secondly, a networked system can be easily modified or expanded by replacing or adding components onto the network without changing the overall structure of the system. The scalability and reconfigurability offered by networked automation systems is helpful to reduce the complexity of the system caused by fluctuating market [5]. Lastly, controllers

Figure 1.1 and then verify the effectiveness of the designed network by assessing the resulted communication QoS and system QoC. This has motivated the research works in this thesis. The objective of the thesis is to find an IIoT solution which offers not only various real-time characteristics needed by different control applications but also the possibility to integrate wireless technologies into the network. The thesis also aims to formulate and evaluate communication QoS of the solution, and eventually study the impact of the network QoS on the control performance at the operational level.

1.2 Major Contribution of the Thesis

The challenge of this thesis is to design an industrial communication network in a holistic way so that it caters for various possible applications in factory automation powered by Industry 4.0, ranging from classic I/O control, motion control, to remote control of the AGVs. The main contributions of the thesis can be summarized as follows:

- A generic solution is proposed for control networks in automation systems, supporting communication channels for guaranteed and best-effort QoS. The proposed scheme not only provides a possibility for hard real-time networked control on the wired segment through a guaranteed channel, it also enables interconnection to wireless segments via a best-effort channel for mobile applications such as the AGV. With advantages over existing standard and proprietary technologies mentioned in [10–12], the solution can be a better candidate for future advanced factory automation.
- The real-time capability of a communication network is normally analyzed through performance evaluation, and recently, much attention has been paid to the mathematical modelling of cycle time and end-to-end delay as they are the most important performance indicators for a communication network [13–18]. This thesis studies these parameters comprehensively for both wired and wireless segments of a hybrid network, which is not considered in these works. A virtual time-stamping approach is developed in the thesis to study the delay which is critical to the NCS design. The result is summarized with a generic model which can be imported by simulation environment for controller design and validation. Additionally, the reliability

of the design is verified as well in the thesis as it might degrade the performance of networked control.

- As the communication system is designed to serve networked control applications in industrial automation, it is not sufficient to analyze the communication performance without any NCS context as in [19–22]. In the thesis, the network design is verified in a case study with industrial network-based control applications including motion control over a wired sub-network and AGV tracking over a wireless sub-network. Moreover, a verification process is introduced to overcome the limitations of existing simulation tools so that the communication QoS and the system QoC can be investigated harmoniously. The effectiveness of the suggested networking solution is demonstrated by simulations.

1.3 Organization of the Thesis

The remainder of the thesis is organized as follows:

Chapter 2 gives a comprehensive literature review on *control of network*, including research works on Ethernet-based industrial communication and wireless extension of its connectivity. The chapter also covers current research in NCS such as networked controller design and tracking control of AGVs.

Chapter 3 introduces the overall architecture and the control components of a typical NCS and existing IIoT communication protocols which can be deployed in the automation system. A task model is introduced which will be further applied in Chapter 5 to study end-to-end delay.

Chapter 4 defines a generic network model which is designed to offer both guaranteed and best-effort services. Wireless extension opportunities on different OSI layers are conceptually discussed and the corresponding scheme to interconnect wired and wireless segments through the best-effort channel is presented.

Chapter 5 suggests a virtual time-stamping approach to analyzing system behaviours, and formulates key QoS parameters (i.e., bus cycle time, controller-to-actuator (C-A) delay, sensor-to-controller (S-C) delay, and communication reliability) that are important to the networked controller design.

Chapter 6 performs evaluation on the network defined and analyzed in Chapters 4 and 5. The results on cycle time, end-to-end delay and reliability are graphically presented and discussed.

Chapter 7 describes an integrated case study with networked control applications over a hybrid network. Particularly, a high-level tracking control algorithm of AGVs is presented. The evaluation strategy is explained, and the system performance is evaluated to illustrate the effectiveness of the designed network.

Chapter 8 concludes the thesis and suggests potential topics for future study.

Chapter 2

Literature Review

As suggested in [23], the NCS research can be classified into two parts. The first part is *control of network* where research works are carried out to design network protocols and services in a way that they are suitable for networked control. This is also to facilitate the second part, *control over network*, which deals with controller design to mitigate the negative impact of the imperfect communication network on control performance. This chapter aims at reviewing existing works, especially in the area of *control of network*, to motivate the study in this thesis.

2.1 Real-time Ethernet

A thorough study of Ethernet-based data-link protocols was carried out in the industrial context by Decotignie in [10]. The study pointed out the problems with conventional Ethernet where the carrier-sense multiple access with collision detection (CSMA/CD) scheme is used as the media access control (MAC) algorithm and stated the evolutionary history of Ethernet from a hub-based architecture to a full-duplex switch-based solution. The reference also suggested a number of schemes to improve the real-time performance, such as time-division multiple access (TDMA), master-slave structure, token passing and bandwidth reservation. These improvement proposals in [10] were adopted in communication networks of industrial automation sector due to increasing demand for real-time communication. Several industrial Ethernet specifications were developed by different market players in the last decade and standardized in IEC 61158. As suggested in

[24–27], EtherNet/IP, EtherCAT, PROFINET, Modbus TCP/IP and Ethernet POWERLINK are the dominant Ethernet-like standard on the market in recent years. These technologies can be classified into three categories as suggested in [11, 12, 28] — on top of Internet protocol (IP), standard Ethernet, and modified Ethernet as presented in Figure 2.1.a, 2.1.b and 2.1.c, respectively.

First, non-real-time (NRT) protocols EtherNet/IP and Modbus TCP/IP are completely built on top of IP. EtherNet/IP utilizes UDP to realize a producer-consumer model for I/O data exchange while Modbus TCP/IP devices transmit data with a TCP-based client-server architecture [29, 30]. Second, Ethernet POWERLINK and PROFINET RT are capable of delivering better real-time performance compared to EtherNet/IP and Modbus TCP/IP while staying fully compliant with standard Ethernet. Seno *et al.* described in [31] the Ethernet POWERLINK data-link layer protocol as polling mechanism is used to grant each device with exclusive access to the network. On the other hand, PROFINET RT establishes firstly a connection between the sender and receiver using TCP and then transmits real-time data over UDP with a producer-consumer model [32]. Last, data-link layers of EtherCAT and PROFINET IRT are completely modified from standard Ethernet to achieve hard real-time (HRT) performance. The difference between EtherCAT and PROFINET IRT on the layer-2 was discussed in [15, 16, 33].

Standard built completely on top of IP is normally not ideal for real-time NCSs. Lian *et al.* established a delay model in [13] for EtherNet/IP and it concluded that EtherNet/IP is not ideally suited to networked control because of stochastic and unbounded end-to-end delays. This is in line with the experimental results available in [34] as the round-trip time of an EtherNet/IP network is considerably large and non-deterministic.

These results motivated researchers to seek better networking solutions over IP-based standard. The performance comparison between EtherNet/IP and one of the real-time (RT) technologies — Ethernet POWERLINK was carried out in [35]. The paper suggested that EtherNet/IP is faster than Ethernet POWERLINK in terms of configuration time due to the intuitive IP-based addressing scheme. This conclusion is in line with re-configurability analysis provided in [36]. It was also suggested that, in terms of real-time performance, Ethernet POWERLINK has less jitter in reaction time than EtherNet/IP and is therefore more suitable for real-time control relying on deterministic delay.

2.1.a:

2.1.b:

2.1.c:

Figure 2.1: Layered communication architecture for performance class (a) NRT; (b) RT; (c) HRT

However, Ethernet POWERLINK is developed on top of standard Ethernet and it is still not optimized for hard real-time applications such as motion control where guaranteed communication service is mandatory. Similarly, PROFINET RT is based on the best-effort paradigm, and the real-time performance of a PROFINET RT network has dependency on network parameters such as load condition [37]. Modified Ethernet is the solution for fast and guaranteed data transfer. Vitturi *et al.* evaluated both Ethernet POWERLINK and EtherCAT in a coordinated motion control application of two independent controlled axes in [38]. The comparison study demonstrated that the HRT standard EtherCAT is advantageous over Ethernet POWERLINK in terms of cycle time, RT throughput and non-RT throughput. Recently, the top-2 modified Ethernet technologies EtherCAT and PROFINET IRT attracted much attention. Performance comparison between two protocols was carried out in many works [15, 16, 33, 39]. It can be concluded from these literatures that EtherCAT provides better real-time capability compared to PROFINET IRT, when frames transmitted on the network is cyclic and of low data volume. There are also research works which attempted to improve the real-time performance of PROFINET IRT. These include the enhancement on message scheduling [40–42], stack architecture [43, 44] and data packing using the dynamic frame packing (DFP) technique [45, 46].

On the other hand, EtherCAT also has drawbacks. Firstly, reliability issue was addressed in [15] as EtherCAT may encounter more transmission errors than PROFINET IRT due to longer transmission time resulted from the summation-frame principle. Moreover, EtherCAT is not as flexible as PROFINET in handling on-demand communication request in event-driven NCS. Researchers have devoted effort to reduce or eliminate this limitation from EtherCAT. Cena *et al.* proposed a controller area network (CAN)-like arbitration scheme for asynchronous data transmission over EtherCAT [47, 48]. Bello *et al.* pointed out that, as the arbitration scheme is based on statically configured priority, asynchronous data with lowest priority may experience long delays when the load of asynchronous data is high. An earliest deadline first (EDF) approach where priority of asynchronous data frame is dynamically assigned was developed in [49–51] to avoid such long delays.

It can be noticed from this review that each standard has its strength and weakness as presented in Table 2.1, and this motivates the thesis to design a network in a way that

Standard	Real-time performance	Special hardware required	Self-configuration	Wireless support	Market share
EtherNet/IP	-	+	+	+	+
Modbus TCP/IP	-	+	+	+	-
Ethernet POWERLINK	o	o	o	o	-
EtherCAT	+	-	+	-	o
PROFINET (IRT)	(+)	-	-	(-)	+

Table 2.1: Comparison on top-5 industrial Ethernet.

it can combine benefits of all these existing technologies and is well-suited to speed up business transition to Industry 4.0. The details of the design can be found in Chapter 4 of this thesis.

2.2 Wireless Extension of Real-time Ethernet

There is a growing trend in industrial automation to deploy wireless communication in a hybrid network as an enabler for certain control applications [52–56]. To realize these applications, Ethernet-based wired sub-network has to be interconnected with a wireless segment at the data-link or the application layer [57–59].

The wireless extension of real-time Ethernet has attracted considerable interest from academic community. Cena *et al.* explained the possibilities of extending EtherNet/IP and PROFINET IO wirelessly over IEEE 802.11 in [57]. EtherNet/IP can be transparently forwarded to the 802.11-based wireless portion via an access point (AP) at the data-link layer. However, it was highlighted that special technique has to be implemented in the AP so that the mapping between IEEE 802.1Q traffic classes and IEEE 802.11e access categories (ACs) is possible. In case of PROFINET IRT, it was concluded that the extension is only possible on the application layer as the access methods employed by PROFINET IRT and 802.11 distributed coordination function (DCF) are totally different (TDMA v.s. carrier-sense multiple access with collision avoidance (CSMA/CA)). The wireless extension of PROFINET IO based on IEEE 802.1, wireless interface for sensors and actuators (WISA), WirelessHART, and ISA100.11a technologies was also discussed in terms of system integration in [52], [60], [61], and [62], respectively. Similarly, it is only possible to extend EtherCAT to a wireless network via the application layer [63–65].

The most inspiring work in this area was presented by Seno *et al.* on a wireless extension scheme at the data-link or the application layer for Ethernet POWERLINK [20, 21]. The literature stated and discussed how polling mechanism originated from Ethernet POWERLINK can be further adopted in the bridge-based extension so that the wireless cycle can be included in the overall bus cycle. A gateway-based extension was also introduced for data exchange between managing nodes and wireless node at the application layer. Other wireless topics related to Ethernet POWERLINK such as the implementation and the integration of gateway device were covered in [66, 67].

It is worth mentioning that many existing studies on this topic do not provide sufficient performance analysis of a hybrid network. For example, only cycle time was evaluated in [65] without any statement for end-to-end delay. The delivery time of the Ethernet POWERLINK network with wireless communication extended at the data-link layer was not provided in [20, 21]. These performance indicators, however, are critical to a NCS. Hence, without any comprehensive performance evaluation, it is difficult to justify the effectiveness of the extension concepts in industrial control applications, especially networked control where fast and predicable communication is desired.

2.3 Performance Evaluation on IIoT Networks

The evaluation on the real-time performance of industrial Ethernet networks is essential in the design of a NCS. As suggested by Seno *et al.* in [68], IEC 61784-2 [69] defines performance indicators (i.e., QoS parameters), such as throughput, bandwidth, synchronization accuracy, and delivery time, for industrial Ethernet specifications. Research attention has been paid to the performance analysis of cycle time [15, 16, 40, 41, 70–74] and end-to-end delay [14, 17, 33, 34, 75–78], mathematically or experimentally. Cycle time refers to the communication time required by the controller to collect and update data memories of all controlled devices and it reflects the real-time capability of the network to refresh data among the devices [72]. It is one of the important parameters in the conventional DCS where processes are not synchronized. On the other hand, end-to-end delay between controllers, sensors and actuators can be fatal in networked control applications.

IEC standard 61784-2 provided mathematical calculations of end-to-end delays (or delivery time as defined in the standard) of protocols including EtherNet/IP, Ethernet POWERLINK, EtherCAT and PROFINET [69]. The delays in an EtherNet/IP network were given as the sum of cable delay, transmission time, switch delay and stack latency in the IEC standard. The end-to-end delays of modified Ethernet technologies including EtherCAT and PROFINET were also presented in IEC standard [69], however, with inconsistent consideration on stack and application delays. Firstly, the stack delay was neglected in the case of EtherCAT as the calculation which can be true only with an application-specific integrated circuit (ASIC) based implementation. On the other hand, this was improved when calculating delays of PROFINET as the stack latencies were considered. Next, the application cycle of sender and receiver was counted as part of the total end-to-end delay in the case of PROFINET. However, this was ignored in the computation of EtherCAT-induced delays in the standard.

The end-to-end delay of EtherCAT networks was investigated by Sung *et al.* in [14]. However, the literature did not provide any probability model and therefore it is not possible to directly apply the results to analyze the performance of the NCS. This problem did not recur when Höme *et al.* evaluated the end-to-end delay of PROFINET in [17], as the probability distributions of network delays were provided. However, the results of [17] were obtained based on Siemens products and therefore were not applicable to control systems using products from other companies than Siemens.

On the other hand, not much focus has been made on the evaluation of hybrid networks based on industrial Ethernet. The end-to-end delay of a hybrid EtherCAT network has not been discussed by any research paper due to the inability of EtherCAT to be extended wirelessly at the data-link layer [48, 65]. This is caused by the fact that the *processing on-the-fly* medium access scheme and the summation-frame scheme adopted by EtherCAT behave very differently from any existing wireless standard. Seno *et al.* presented the wireless extension scheme of Ethernet POWERLINK using a bridge or gateway and further investigated the communication performance of the hybrid network in [20, 21]. The duration of the isochronous phase of a bridge-based hybrid network and the delivery time of a gateway-powered hybrid network were formulated. However, the work did not discuss the delivery time under the bridge-based extension scheme which

can be more practical in the NCS as wireless nodes communicate with the managing node using isochronous period. A complete analysis (cycle time and end-to-end delay) on industrial hybrid networks has not, to the best of the author knowledge, yet comprehensively covered by any academic literature. The limitations from the literature drive this thesis to make holistic performance analysis on the proposed network so that the results can be useful for the NCS design and verification. See Chapter 5 and 6 for contributions to this topic.

2.4 Network-based Control over IIoT Networks

Before real-time standards such as EtherCAT and PROFINET IRT were available, industrial NCSs were derived mostly based on standard computer protocols such as UDP [79–81] or HTTP [82]. One of the hot topics in the field of the Ethernet-enabled NCS is the controller design to tackle problems induced from imperfect networks, such as link delay/jitter [83–87] and packet loss [80, 88]. The thesis is more interested in reviewing works which describe networked control applications over HRT and hybrid networks so that the case study to be presented in Chapter 7 can be generic and relevant.

The possibility of implementing EtherCAT-powered advanced control techniques was conceptually discussed in [89]. Delgadol *et al.* presented an EtherCAT-based networked motion control system with one controller and two servo drives installed on a mobile robot. It was demonstrated via experimental study that the servo motors controlling left and right wheels were able to follow the desired velocities sent from the controller over EtherCAT and consequently the robot could track the reference trajectory. Erwinski *et al.* presented an open architecture for computerized numerical control (CNC) systems and the proposal was verified by measuring the cycle/jitter of Ethernet POWERLINK network in the CNC system [90]. A stepper motor drive controlled over EtherCAT was presented in [91]. A modified PI controller with feed-forward scheme was introduced to overcome the problem with reverse motor rotation. Zheng *et al.* investigated experimentally the real-time EtherCAT-powered networked PI controller in multi-inverter parallel systems [92]. However, these works did not consider network delays in the controller design. Reference [93] discussed several algorithms which compensate constant

time-delay in order to achieve improved performance of a motion control application and the proposed compensation schemes were verified by observing the disturbance and step response from a Yaskawa servo drive with EtherCAT interface. The probability distribution of the delays caused by PROFINET networks was investigated in [17] and the control performance of the NCS was obtained using a Monte Carlo simulation based on the delay models. Velagic *et al.* described a control server which remotely controls an induction motor over the PROFINET network [94]. The experiment study showed satisfactory results of the NCS, despite the time-varying delays induced from the network.

Similar to the case of control over wired networks, the outcome of networked applications over wireless data networks such as mobile robot control can be also impacted by the network. The robot path-tracking can become unstable when the network-induced delay is not tolerable by the control system. A typical application for wireless communication in automation systems is mobile robots such as the AGVs. Tipsuwan *et al.* proposed a methodology named as GSM to remotely control mobile robots [95, 96]. The advantage of the suggested scheme over the optimal gain scheduling technique is the portability of the scheme, as the tracking controller and GSM are completely separated. Lozoya *et al.*, on the other hand, adopted Kalman filter to recursively estimate the wireless network delays. The simulation results showed that the AGV path-tracking can be improved considerably with the help of the Kalman filter [97]. Kumagai *et al.* developed an adaptive remote control strategy where the moving speed of the mobile robot is dynamically adjusted according to a delay estimation [98]. In principle, the robot moves slower under this strategy when the RTT is large. This design was verified via simulation studies. Santos *et al.* suggested a filtered Smith predictor for dead-time compensation in order to mitigate the negative impact of the network delays on the remote control of the mobile robots and the effectiveness of the strategy was demonstrated through experimental tests [99].

Chapter 3

Networked Control System Overview

One of the prerequisites to study the QoS of an IIoT-enabled network is the understanding of control systems at the component level as well as the architectural level. A generic schematic of an industrial network-based control system is shown in Figure 3.1. The system is typically made up of controllers, controlled devices on the wired sub-network, wireless nodes on the wireless sub-network, and additionally linking components such as switches, bridges, and gateways. The communication network functions as a shared memory region which allows industrial processes to be executed on the components to exchange information. This chapter firstly describes a NCS at both the system and component levels. The existing networking technologies are then revisited to facilitate the network design in Chapter 4 by comparing the pros and cons of each protocol. Lastly, a task model is established to mathematically describe the attributes of industrial processes and the interaction between processes.

3.1 System Architecture and Configuration

3.1.1 System Architecture and Components

As presented in Figure 3.1, the DCS/NCS is normally controlled by a central controller such as programmable logic controller (PLC) or industrial PC [6]. The controller is responsible for writing output to the network based on the execution results of application programs which scan input data from the network. One controller can be used to control

multiple devices as they are connected with a shared network. Nowadays, controller normally integrates a very powerful central processing unit (CPU) in order to perform fast computations of advanced algorithms such as the model predictive control (MPC) [100], multi-axis motion control [101–103], etc. On the other hand, intelligent field devices can be equipped with interfaces to I/O modules, sensors and actuators which do not need network connectivity. For example, the variable-frequency drive designed in patent [104] can read output signals directly from incremental and absolute encoders and use the feedback to perform closed-loop control of position or speed of an induction motor. The CPU on such an intelligent device are multi-tasking, e.g., one thread runs with a local timer to periodically perform measurement activities from a sensor and another thread is ready to actuate upon receiving any set-point from the network. Intelligent devices or nodes have field connectivities which allow them to communicate through network [104]. Switches, which link controllers to devices on wired sub-network, manage the flow of data frames across the sub-network. Bridges or gateways are deployed to interconnect the wired and the wireless parts of the network via either data-link or application layer.

Typically, each component is composed of an application CPU, a network controller and a physical layer (PHY) transceiver. The transceiver circuits carry out transformation of data bits between electrical and physical signals that are compliant with Ethernet or radio frequency (RF) standard. Then, a network controller such as standard or proprietary field-programmable gate array (FPGA)/ASIC/system on chip (SoC) is needed to implement data encoding/decoding, memory mapping, error handling, and other functionalities required at the data-link layer. For Ethernet-based networks, it can generate hardware signals or interrupts to the application CPU for critical usage such as exception handling (alarm signal) and clock synchronization (*sync* signal). A piece of dual-port random-access memory (DPRAM) is normally integrated on the network controller to facilitate cyclic and acyclic data exchange between the CPU and the network. Finally, application-specific tasks, including sensing, actuating and control (e.g., proportional-integral-derivative (PID) control, MPC or remote path tracking), are implemented on the application CPU, denoted as J_s , J_a , and J_c , respectively, as illustrated in Figure 3.1. These industrial processes can be executed synchronously or asynchronously, depending on the system configuration [105].

Figure 3.1: System architecture of industrial control systems

Configuration	Sensor	Controller	Actuator
1	Free-running	Free-running	Free-running
2	Free-running	Frame-driven	Frame-driven
3	Clock-driven	Frame-driven	Frame-driven

Table 3.1: Comparison among system configurations.

3.1.2 System Configurations

There are several possible system configurations in industrial automation systems as summarized in Table 3.1. The first scenario is that all processes $\{J_s, J_a, J_c\}$ run independently, i.e., free-running. Processes under the free-running mode are asynchronous. In the free-running mode, process is triggered once the current execution is finished (event-triggered) or cyclically by a device-local timer (time-triggered). It should be noted that the local timer usually leads to relatively large jitter compared with global clock. This kind of configuration is not common in industrial automation but can be used in control systems with slow dynamics.

In the second scenario, an event-driven approach is used to achieve some degree of synchronization on the outbound link. The event is dispatched from the networked controller to the CPU whenever a frame is received. Actuator task J_a (and controller task J_c) can be activated upon reception of the event. On the other hand, sensor device is not synchronized via the data frame due to its nature being an input device. Instead, it remains to operate in the free-running mode using local timer to ensure the freshness of sampled data.

Lastly, the determinism of the inbound link can be improved if the sensors are synchronized with a global clock. Clock synchronization enables networked components to share the same system time. The sensor task J_s is executed based on the event dispatched at the time instances pre-determined to avoid oversampling. This is normally applied in real-time systems such as NCSs where determinism is required. The synchronization accuracy can be far below $1 \mu s$ to achieve equidistant sampling [106, 107]. In this thesis, the last scenario with time-triggered (via local timer or clock) sensors and event-triggered (frame-driven) controllers/actuators is considered due to the fact that it is commonly applied in industrial DCS or NCS [85, 108].

3.2 Communication Network and Topology

Once the system configuration is determined, the improvement can be made on networked components, e.g., in terms of execution efficiency, to improve the performance of the NCS. At the same time, the performance of the communication network which acts as the data interface between components can be improved as well [23]. The communication network performs cyclic and acyclic data exchange task denoted as J_e in Figure 3.1. Next, the existing IIoT specifications are to be revisited. The IIoT is part of a larger concept known as the IoT. The IIoT networking protocols can be mainly divided into three categories, i.e. fieldbus, industrial Ethernet and industrial wireless. Accordingly, a communication network can be wired, wireless or hybrid in principle.

3.2.1 Overview of Existing Ethernet-based Technologies

Ethernet-based protocols have been boosted by the trend of IIoT and Industry 4.0 to achieve faster growth than traditional fieldbus. From year 2018, industrial Ethernet has overtaken fieldbus to become the most popular category in terms of the number of new device installations in factory automation [26, 27]. The top-5 dominant industrial Ethernet technologies in the market are EtherNet/IP, PROFINET, EtherCAT, Modbus TCP/IP and Ethernet POWERLINK. These protocols were developed and supported by different companies and organizations and are capable of various real-time characteristics. Wireless technologies, on the other hand, is of the highest annual growth rate among all technologies, although the current market share is still significantly smaller than fieldbus and industrial Ethernet.

In recent years, real-time Ethernet has been promoted by big players in industrial automation. With improved data-link protocols and services, these technologies aim at delivering deterministic communication as guaranteed in traditional fieldbus systems. With controlled medium access, queues can be eliminated to minimize delays and jitters. The candidates of real-time Ethernet include Ethernet POWERLINK, EtherCAT and PROFINET IRT.

Figure 3.2: An Ethernet POWERLINK cycle

3.2.1.1 Ethernet POWERLINK

Ethernet POWERLINK was firstly developed by company B&R GmbH in 2011 to replace CSMA/CD in legacy Ethernet by polling [68]. There are two types of network entities, managing node and controlled node. The managing node is normally integrated as part of the controller in a DCS or NCS, and is responsible for system control logics and network management. Only one managing node is allowed per network. On the other hand, the controlled nodes can be sensor or actuator devices with communication capabilities, and up to 240 controlled nodes can be supported per network [90].

Ethernet POWERLINK allows both real-time and acyclic communication traffic on the network by specifying an isochronous period and an asynchronous period as shown in Figure 3.2. During the isochronous phase, the managing node polls each controlled node by a unicast poll request (PReq) frame (unicast). After some response time, the pulled controlled node responds by broadcasting a poll response (PRes) frame to enable slave-to-slave communication. During this phase, the managing node also collects information of asynchronous transmission requests from devices to facilitate the asynchronous phase where selected devices may transmit acyclic frames. Lastly, an idle phase is defined to ensure that all transmissions are concluded before starting a new cycle.

3.2.1.2 EtherCAT

The scheduling procedure of EtherCAT MAC layer is completely modified from standard Ethernet. The principles adopted by EtherCAT such as logical addressing, summation-frame structure, and *processing on-the-fly* mechanism were completely developed from

scratch. For example, logical addressing and summation-frame structure allows EtherCAT to encapsulate payload of slave devices into possibly one single Ethernet frame so that the same overhead is shared by all devices [109]. This significantly enhances the competitiveness of EtherCAT over PROFINET and Ethernet POWERLINK in terms of bandwidth utilization (possibly higher than 90%) [110]. As these features are not at all compatible with IEEE 802.3, EtherCAT can be only realized by special hardware, e.g., Beckhoff ASIC ET1100.

Figure 3.3 shows the internal schematic of an ET1100-based EtherCAT device which supports up to 4 ports. As a core element of the ASIC, EtherCAT processing unit performs error checking on incoming frame and forwards it to the next open port. In other words, EtherCAT devices also operate as switches which forward layer-2 frames to the destination port. An EtherCAT frame is processed on-the-fly by the processing unit with constant and minimal latency when it passes through a slave device. The frame is finally redirected to the master after it reaches the last slave on the ring. This results in short and deterministic cycle time. The distributed clock (DC) can keep EtherCAT devices synchronized precisely and accurately, with a tolerance of less than 100 ns [107]. It should be noted that the DC is optional as it is up to the device manufacturers to support it or not.

3.2.1.3 PROFINET

In contrast to EtherCAT, PROFINET is treated as an IFP as the controller issues individual frames to devices. PROFINET IRT can also function as a SFP via the DFP scheme. However, the DFP has practical limitation as it only works with a bus topology [111]. Consequently, this feature is not considered in the thesis as it disables complicated topologies. As illustrated in Figure 3.4, PROFINET covers full range of control applications in industrial automation as it provides communication services with multiple real-time classes.

Firstly, PROFINET NRT supports data transmissions based on transmission control protocol (TCP) or user datagram protocol (UDP). This real-time class can be applied for non-time-critical usage such as device configuration, firmware update, and status monitoring. With PROFINET RT, data frames are treated with priority according to

Figure 3.3: Schematic of an EtherCAT slave device

IEEE 802.1Q. It is suitable for cyclic data and event-driven (interrupt) transmissions for time-critical processes in the conventional DCS. Lastly, PROFINET IRT is capable of deterministic data delivery by amending IEEE 802.3 on the data-link layer. As a result, FPGA, ASIC or SoC is required for the implementation of the modified MAC (for example ERTEC 200P from Siemens). Transmissions orders are determined by a scheduling algorithm which can be found in detail in literatures [112, 113]. At initialization phase, transmission sequences are downloaded to controllers and devices by an engineering tool. Transmission requests are raised at time instances statically based upon the predefined schedule. PROFINET IRT offers improved real-time performance by reserving bandwidth for clock-driven data transmissions and is most ideal for time-critical applications in the NCS such as motion control.

3.2.2 Comparison between Industrial Ethernet Networks

One possible way to categorize these technologies can be based on real-time performance [11, 111]. The first class refers to NRT technologies built on top of TCP/IP suite and therefore is capable of data delivery with best effort. The technologies belonging to

Figure 3.4: Real-time classes offered by PROFINET

this class, including EtherNet/IP and Modbus-TCP/IP, can be directly implemented using standard Ethernet hardware (Ethernet controller, hub, switch, etc.) together with standard TCP/IP stack which is commercially available [114, 115]. The response time of a few milliseconds can be achieved and is highly dependent on the implementation of TCP/IP stack. In most cases, the operating system (OS) processes the TCP/IP threads with low priority and the stack processing time can be quite variable. Real-time specifications in the second class such as PROFINET RT and Ethernet POWERLINK still use unmodified hardware but the TCP/IP protocol is abandoned to improve real-time capability from the first class. The protocol data units (PDUs) are directly transported in the Ethernet frame. The third class aims at the top performance among all categories with guaranteed data delivery. To achieve this goal, technologies in this category adopt only the physical layer defined in IEEE 802.3. The data-link layers of PROFINET IRT and EtherCAT are modified completely from IEEE 802.3 to achieve HRT services and consequently dedicated hardware is required. The characteristics of these protocols are summarized with Table 3.2.

The technologies listed in Table 3.2 use global clock for synchronization purpose. For example, EtherNet/IP *CIP Sync* and PROFINET IO, respectively, adopt the original (IEEE 1588-2002) and a modified version (based on IEEE 1588-2008) of precision time protocol (PTP) [116, 117] — precision transparent clock protocol (PTCP). On the other hand, EtherCAT implements the DC — a simpler clock synchronization mechanism than PTP, due to its logic ring topology [107]. All these schemes can achieve sub- μ s accuracy which is sufficient for most industrial NCSs.

Real-time class	NRT	RT	HRT
Technology	EtherNet/IP Modbus-TCP/IP PROFINET NRT	Ethernet POWERLINK PROFINET RT	EtherCAT PROFINET IRT
Hardware	Standard	Standard	Proprietary
Gigabyte readiness	Yes	Yes	No
Data-link layer	IEEE 802.3	IEEE 802.3	Proprietary
Addressing	IP	MAC	Proprietary
Frame type	Individual	Individual	Individual/Summation

Table 3.2: Comparison between wired IIoT technologies.

Another important point is the Gigabyte readiness from the candidates. Since the physical layers of all protocols listed above are derived from standard IEEE 802.3, both 100BASE-TX (copper) and 100BASE-FX (fiber-optic) cables are supported. It should be noticed that real-time Ethernet will sooner or later enter the Gigabyte era (1000BASE-TX, 1000BASE-SX and 1000BASE-LX) as a couple of protocols have already been released (e.g., EtherNet/IP and CC-Link IE Field), or will be released (e.g., Ethernet POWERLINK, EtherCAT G/G10, etc.) specification of transmission speed at 1 Gbps and higher [118, 119]. However, EtherCAT and PROFINET IRT requires redesign on the hardware to support Gigabyte Ethernet. It should be highlighted that the IFP such as PROFINET IRT will enjoy more advantages from the transmission speed-up as compared with the SFP such as EtherCAT [15].

3.2.3 Industrial Wireless Networks

Wireless technologies are growing even faster than industrial Ethernet [26, 27]. IEEE 802.11 standard was originally developed for data communication in wireless local area networks. On the physical layer, wireless nodes may work in either 2.45 GHz industrial, scientific, and medical (ISM) band (802.11, 802.11b, 802.11g) or 5 GHz band (802.11g/a). On the data-link layer, the DCF and the point coordination function (PCF) were specified for media access in IEEE 802.11. IEEE 802.11 supports dynamic switching between the DCF and the PCF depending on the load condition, number of wireless nodes and also packet size.

The DCF is essentially a 1-persistent random access protocol with delay. It is a distributed algorithm using the CSMA/CA technique while the PCF achieves coordination

Network	802.11 DCF	802.11 PCF	802.15.4
Data-link layer	CSMA/CA	Polling	CSMA/CA
Periodic data	No	Yes (with jitter)	Yes (beacon-enabled)
Throughput	54 Mbps	54 Mbps	125 kbps
Range	50-100m	50-100m	10m

Table 3.3: Comparison between wireless IIoT networks.

with a centralized polling mechanism. With the DCF, all wireless nodes compete to gain access to the channel. A node may only start to transmit when the channel is sensed as idle by a carrier-sensing mechanism. Otherwise, it has to wait until the end of the current transmission and restart transmission when the channel is detected to be idle. The DCF functions well when the data load is low, and on the other hand, poorly during high load conditions. However, it does not offer deterministic data communication [57]. On the contrary, the PCF is designed for time-bounded services due to the polling mechanism and consequently can be more attractive to industrial control applications.

IEEE 802.11 is optimized to transmit packets that contain large payloads. In contrast to IEEE 802.11, IEEE 802.15.4 standard focuses on the low-rate wireless personal area network (LR-WPAN) where wireless devices transmit packets with small payloads infrequently over a static network. Effective data rate of up to 250 Kbps is supported at the 2.4 GHz band with direct sequence spread spectrum (DSSS) modulation technique. The standard basically supports two types of channel access schemes, depending on if there is beacon or not. Beacon is periodically sent out by a network coordinator under the beacon-enabled scheme. A contention access period (CAP) takes place as soon as the beacon broadcast is over. The MAC protocol applied during the CAP is slotted CSMA/CA. Optionally, there can be a contention-free period (CFP) after the CAP. In the CFP, guaranteed time slots (GTSS) are reserved for real-time periodic communication. On the other hand, beaconless operation scheme uses unslotted CSMA/CA, and consequently the controlling of the medium access is completely decentralized and asynchronous.

In this thesis, IEEE 802.11 is selected over IEEE 802.15.4 to be deployed to interconnect with the Ethernet-based portion of the hybrid network because of much higher transmission rate and larger range coverage as shown in Table 3.3, and the additional possibility of imposing prioritization on IEEE 802.11 packets by assigning ACs.

Figure 3.5: Network topology of a typical NCS

3.2.4 Topology in Hybrid Networks

The first version of Ethernet specification only supported line topology based on hubs and repeaters. Nowadays, with the development of switches and bridges, Ethernet-enabled technologies in general support multiple topologies, such as star, line, and tree. This gives flexibility for device installations in NCSs. It is worth mentioning here that the proprietary implementation of modified Ethernet MAC, such as ASIC, FPGA or SoC, normally supports internal switch which gives more flexibility in terms of topology.

An example of the system topology is illustrated with Figure 3.5, where controllers, devices, switches, bridge and wireless nodes are denoted as *ctrl*, *dev*, *swt*, *br*, and *wln*, respectively. The number of controlled devices and wireless nodes are represented by N_{dev} and N_{wln} , respectively. The controllers and devices are interconnected via switches on the wired part. On the other hand, a bridge or a gateway can extend Ethernet-based segment to the wireless part. It is also possible for wireless nodes to form additional wired sub-networks as described in [120].

3.3 Task Model

The task model can be established based on the processes defined and presented in Figure 3.1. Each process can be described mathematically by $J_x = (S_x, D_x, P_x) \mid \forall x, D_x \leq P_x$, where S_x , D_x and P_x are the start time, execution time, and deadline of task J_x . For

time-driven tasks, P_x specifies the period of the task. In case of event-driven tasks, P_x can be interpreted as the minimum interval between task executions.

As discussed earlier, sensor tasks are considered as time-triggered in the thesis, and can be described as $J_s = (S_s, D_s, P_s)$ where P_s means the sampling period. On the other hand, actuator and controller are event-driven as they act only upon receiving data frame from the network. Correspondingly, they can be respectively described as $J_a = (S_a, D_a, P_a)$ and $J_c = (S_c, D_c, P_c)$. For J_a , P_a can be considered equal to D_a as the actuator task is ready for the next incoming event once the current one is processed. However, in order for the control cycles and network cycles to be synchronized, controller period P_c can be longer than D_c which represents the execution time of computation task. Network task J_e is executed twice, namely J_{e-sc} and J_{e-ca} in each control cycle. J_e for S-C and C-A links are both scheduled periodically based on P_s , i.e., $J_{e-sc} = (S_{e-sc}, D_{e-sc}, \mathcal{D}_e)$ and $J_{e-ca} = (S_{e-ca}, D_{e-ca}, \mathcal{D}_e)$, where $\mathcal{D}_e$ is the bus cycle of the network.

Next, $J_x(k), k \in \mathbb{N}^+$ is defined as task J_x in the k -th control cycle. The data flow of the controlled plant of index n in the k -th control cycle can be stated as $J_s(n, k) \rightarrow J_{e-sc}(n, k) \rightarrow J_c(k) \rightarrow J_{e-ca}(n, k) \rightarrow J_a(n, k)$.

3.4 Problem Statement

The remaining of this thesis is to address the following challenges that exist in the IIoT-based NCSs in industrial automation.

(i) Design of a generic IIoT-powered hybrid network

With conventional industrial automation, one large-scale control system typically consists of several subsystems, with several controllers and networks [6]. For example, one controller can be installed to control only devices connected to a fieldbus or industrial Ethernet network, and another controller aims only at managing wireless nodes in a wireless local area network (WLAN). It is complicated for devices to exchange data from different subsystems or networks as the processes are not synchronized with each other. Controller-to-controller communication is required in case a device in the wired network needs to talk to a wireless node in the WLAN.

This increases the complexity of the system and, consequently, makes the system inflexible to cater for varying control requirements in Industry 4.0. However, with the increasing computational power and communication bandwidth, it is technically feasible to reduce the complexity of the system by integrating subsystems together whenever plausible.

From the communication point of view, it is challenging to incorporate the following features into one network:

- The network is operational with all system configurations as stated in Table 3.1;
- The network is able to provide guaranteed communication services for time-critical control applications (e.g., networked motion control);
- The network can also offer best-effort communication services for non-time-critical applications (e.g., environmental control);
- The network can integrate real-time Ethernet and wireless technologies together to achieve both hard real-time control and mobile applications (e.g., AGV remote tracking) within the same network cycle;
- The network should be capable of delivering consistent services in both small- and large-scale system, i.e., scalability.

(ii) **Analysis of the QoS of the hybrid networks for deployment in a NCS**

Networked control relies on the real-time capability of the communication link. The determinism is critical for data exchange between the controller and devices because communication link can degrade the control performance of the NCS by introducing information distortion or mishandling. However, the constraints from the communication networks should be identified prior to the installation of controllers, actuators and sensors in the system. Otherwise, it can lead to problematic situations such as the need of system redesign which can be very costly. It is of great significance to verify the networking design by evaluating network performance for

the considered control application before deploying devices and components in the real system.

As there is limited research effort spent on designing a generalized hybrid real-time network, the QoS modelling of the hybrid network with all desired features is not comprehensively available from any existing works. It is also complicated to characterize the communication performance such as cycle time and end-to-end delays due to large dependency on system parameters such as number of controlled devices, forwarding/processing latencies, network topology, payload size and response time of components. The result of network performance analysis must be available in a form that is ready for controller design and/or verification.

Chapter 4

Design of Real-time Hybrid IIoT Networks

With the background information provided in Chapter 3 on the NCS and the existing networking specifications, this chapter designs a hybrid network which addresses the problems listed in Chapter 3. First of all, the frame organization is introduced, and the calculations of frame length are provided. Next, the MAC of different communication channels is presented. Lastly, wireless extension schemes through different layers of the network is conceptually discussed. The network design is presented with all necessary information so that the performance analysis can be carried out in Chapter 5.

4.1 Conceptual Design

There is no point to define a completely new network from scratch and the idea of the network suggested in this thesis is mainly originated from existing technologies — Ethernet POWERLINK, EtherCAT and PROFINET. EtherNet/IP and Modbus-TCP/IP are built on top of standard TCP/IP suite. Both are not suitable for real-time control loops which are closed over the network due to the time-variant delay caused by the TCP/IP stack. Therefore, they are considered in this thesis only as a performance benchmark.

Firstly, the network should specify one guaranteed (GTD) channel and one best-effort (BE) channel to meet different communication requirement. This concept of multi-channel comes from PROFINET and Ethernet POWERLINK. PROFINET supports varieties of real-time classes, i.e, `RT_CLASS_UDP`, `RT_CLASS_1`, `RT_CLASS_3` as defined

in IEC 61158-6-10 [121]. These classes can be mapped to NRT, RT and HRT groups illustrated in Table 3.2. The GTD phase aims at providing guaranteed services with strictly bounded network delays. It can be applied for networked control applications in the wired portion of the hybrid network. On the other hand, the BE phase can be deployed to integrate the wireless segment on either the data-link or the application layer [57].

Second, the data-link of the GTD channel reuses protocols and services defined in EtherCAT and PROFINET IRT. EtherCAT outperforms PROFINET IRT when the payload is small and therefore the summation-frame scheme is recommended during the GTD phase for low data volume applications [33].

Lastly, Ethernet POWERLINK adopts a polling method where the managing node sends a poll request to a controlled device and waits for the response from slave device. The polling scheme is adopted in this thesis for wireless extension over the BE channel as it can easily include wireless communication in the bus cycle. This mechanism is considered in this thesis mainly for the wireless extension via the BE channel.

4.2 Frame Structure

From this section onwards, the proposed networking solution is described and discussed. The frame organizations of wired and wireless sub-networks are obviously different. On the wired segment, there are various types of frames based on communication classes defined in Table 3.2. NRT, RT and HRT frames can be organized based on standard UDP, standard Ethernet, and modified Ethernet frames shown in Figure 4.1.a, 4.1.b and 4.1.c, respectively. The layer-2 frame length (in Byte) for the n -th controlled device or wireless node can be calculated as

$$f(n, \gamma) = f_o(\gamma) + f_p(n, \gamma), \quad \forall n \in \{\mathbb{D} \cup \mathbb{W}\}, \quad (4.1)$$

where γ is the protocol used by the frame, and the numbers of data bytes of overhead and payload are denoted with f_o and f_p , respectively. The payload is specified according to the control application, with the constraint of maximum transfer unit (MTU) from protocol γ as shown in Figure 4.1. Transmission time is defined as the time necessary for the sender

to transmit all data bits via transmission channel, i.e., $T_{frm}(n, \gamma) = 8f(n, \gamma)/C(\gamma)$, where $C(\gamma)$ is the channel capacity (bit rate) in bps for protocol γ .

4.2.1 Non-real-time Frame

It is common that NRT standards, such as EtherNet/IP and PROFINET NRT, make use of UDP to pack process data, instead of TCP. This is simply because of UDP's advantages over TCP, such as its lightweight, smaller overhead, and consequently faster and more efficient transmissions. The UDP-based frame structure can be illustrated with Figure 4.1.a. Padding bytes are added to the frame if the frame is sized 64 Bytes or less. When $\gamma = \text{NRT}$, f_o on the physical and the data-link layers is 70 Bytes in total, including virtual local area network (VLAN) tag. This indicates that the bandwidth is not even half-utilized for the NRT frame with $f_p < 70$ Bytes. The MTU for the NRT frame is 1472 Bytes.

4.2.2 Real-time Frame

A RT frame is built directly on top of IEEE 802.3 to reduce communication overhead compared to UDP. Additionally, the RT frame generally adopts IEEE 802.1Q [122] with high priority in the VLAN tag to ensure that it can be forwarded with priority in switches. The structure of a RT frame is shown in Figure 4.1.b. The value of the VLAN *TPID* field is by default 0x8100 to indicate that the frame is an 802.1Q frame. The *PCP* specifies a priority value ranging from 0 to 7. The RT frame can be sent with priority 6 or 7 so that it is forwarded first by the switch [123].

From Figure 4.1.b, the MTU of the RT frame is 1500 Bytes which is larger than the case of NRT frame due to reduced overhead. For $\gamma = \text{RT}$, padding bytes are inserted when $f_p < 46$ Bytes. The payload can be used for the wired part only or both wired and wireless segments. The payload may contain process data of either one wireless node or the whole wireless portion (as illustrated in Figure 4.1.b), depending on the operating mode on the wireless side. When the wireless part operates in PCF mode, the *data* field can be made up by N_{wln} subframes. Either *store and forward* or *cut-through* switch can be applied to forward the RT frame in the network. The time required to forward frames with *store and forward* technique is normally higher than $10 \mu\text{s}$ and is frame length

8 Bytes	6 Bytes	6 Bytes	4 Bytes	2 Bytes	20 Bytes	8 Bytes	18 - 1472 Bytes	4 Bytes	12 Bytes
Preamble/ SFD	Destination	Source	VLAN Tag	EtherType	IP Header	UDP Header	Data	FCS	IFG

4.1.a:

4.1.b:

4.1.c:

Figure 4.1: Data frame structure for real-time class (a) NRT; (b) RT; (c) HRT;

dependent [124]. On the other hand, *cut-through* is capable of forwarding a frame with $3 \mu\text{s}$, regardless of the size of the frame [124].

4.2.3 Hard Real-time Frame

The HRT frame in principle can be adapted from the standard Ethernet frame to achieve better performance. For example, a switching method named as *fast forwarding* was introduced to shorten forwarding delays by rearranging the frame structure of PROFINET version 2.3. With *fast forwarding*, the destination information in the *FrameID* field is shifted to the start of the header (see Figure 4.1.c). It is possible to cut down forwarding delay significantly from $3 \mu\text{s}$ to $0.6 \mu\text{s}$ by making early decisions during forwarding [16]. This mechanism is adopted in the thesis to improve the real-time performance of the frame. VLAN is no longer used here as it can only improve the performance of the RT frame.

Next, the frame length is to be formulated for scenarios of individual and summation-frame. When $\gamma = \text{HRT_IFP}$, the length of the HRT frame for station n is given by

$$f(n) = f_o(\gamma) + \max(46, f_p(n)), \quad 0 \leq f_p(n) \leq 1500, \quad \forall n \in \mathbb{D}, \quad (4.2)$$

On the other hand, SFP such as EtherCAT and the DFP-enabled PROFINET IRT packs sub-frames in a HRT frame. The controller might only need to issue one Ethernet frame to accommodate all process data needed by networked devices in the system, provided that the total payload is small enough to be fit into a frame. In this case, there can be N_{dev} subframes encapsulated in HRT frame. Consequently, the usage of bandwidth is very efficient as communication overheads on the physical and the data-link layers is shared by all devices. For $\gamma = \text{HRT_SFP}$, the length of the summation-frame $f(\mathbb{D})$ is given by:

$$f(\mathbb{D}) = \begin{cases} f_o(\gamma) + \max(42, \sum_{n \in \mathbb{D}} f_p(n)), & 0 < \sum_{n \in \mathbb{D}} f_p(n) \leq 1496, \quad \forall n \in \mathbb{D} \\ \lceil \frac{N_{dev}}{\sum_{n \in \mathbb{D}} f_p(n)} \rceil f_o(\gamma) + \sum_{n \in \mathbb{D}} f_p(n), & \sum_{n \in \mathbb{D}} f_p(n) > 1496, \quad \forall n \in \mathbb{D} \end{cases} \quad (4.3)$$

where $\sum_{n \in \mathbb{D}} f_p(n)$ denotes the total payload size of the summation-frame for $\forall n \in \mathbb{D}$. The header required for each summation-frame is 4 Bytes if the fieldbus memory management unit (FMMU) concept defined in EtherCAT technology is reused to map data from logical space to the local address space at each device. With the FMMU, it is possible for the controller to address several devices with a single summation-frame using logical addressing [125]. The advantage of the SFP dwindles when the payload is increased. It can be seen that f_o gets multiplied in the event that $\sum_{n \in \mathbb{D}} f_p(n)$ is greater than 1496 bytes.

4.2.4 IEEE 802.11

On the wireless sub-network, the format of the DATA and the ACK frame defined in the IEEE 802.11 standard [126] is reused without any modification in the thesis (see Figure 4.2). It should be noted that packet aggregation introduced in 802.11n is not considered here. The overheads induced in the physical layer of the DATA frame are a 24-Byte preamble/header and inter-frame spacings, i.e., short inter-frame spacing (SIFS), DCF inter-frame spacing (DIFS), etc. which is not included in Figure 4.2. The overhead at layer-2 $f_o(\gamma = \text{WLAN})$ is 34 Bytes (802.11a/b/g). The length of the MAC layer frame can be simply calculated as

$$f(n) = f_o(\gamma) + f_p(n), \quad 0 \leq f_p(n) \leq 2312, \quad \forall n \in \mathbb{W}, \gamma = \text{WLAN} \quad (4.4)$$

Note that the physical layer overhead is calculated separately as it is transmitted at a lower rate (e.g., 6 Mbps in 802.11g).

Similar to the *PCP* field in a VLAN tag in IEEE 802.3, the traffic identifier (*TID*) field in IEEE 802.11 is a 4-bit indicator of the priority of the wireless data frame. The MTU on the wireless segment is significantly larger than that of the Ethernet-like protocols, i.e., $\gamma \in \{\text{NRT}, \text{RT}, \text{HRT_IFP}, \text{HRT_SFP}\}$. However, the payload for sensors and actuators in industrial automation are normally small (less than 100 Bytes). In this case, the payload at every wireless node should not exceed the threshold which is equal to the MTU of the wired segment.

Figure 4.2: IEEE 802.11 DATA frame structure

Figure 4.3: Bus cycle incorporating guaranteed, best effort and reserved phases

4.3 Communication Scheduling

4.3.1 Bus Cycle

In order to satisfy communication requirements from various applications, a network cycle can be broken into several virtual communication channels, i.e., a deterministic channel with guaranteed QoS, an open channel with only IEEE 802.3-compliant best-effort QoS and a reserved channel serving as a transition phase, as illustrated in Figure 4.3. This concept is adopted from existing PROFINET and Ethernet POWERLINK technologies.

4.3.2 Guaranteed Channel

The GTD phase is only reserved for cyclic HRT frames. The GTD channel adopts either static TDMA ($\gamma = \text{HRT_IFP}$) or *processing on-the-fly* ($\gamma = \text{HRT_SFP}$) mechanism as the medium access to transmit HRT frames. Both schemes adopted by the GTD phase is capable of deterministic communication behaviour.

The static TDMA schedules frame transmissions according to the network topology deployed [42]. The schedule is downloaded to every device in the network during ini-

tialization phase. With the help of clock synchronization, data transmissions take place on the devices exactly at time instances predefined in the static schedule. Consequently, the data deliveries are deterministic and the communication QoS such as bus cycle and end-to-end delay can be ensured.

It is important to optimize the scheduling efficiency as the scheduling is designed and commissioned before communication starts. A simple line topology illustrated in Figure 4.3 is discussed here as an example. One possible way to schedule the communication here is to address devices according to their logical positions in the network topology [113]. Slipstreaming scheme shown in Figure 4.4.a means that device that is physically the last can be attended first. This mechanism is optimal for line topology to achieve minimal cycle time [40, 42]. Additionally, communications in S-C and C-A links can take place simultaneously due to full-duplex operation. It can be seen from the figure that both $J_{e-sc}(k)$ and $J_{e-ca}(k)$ are made up of a collection of inbound and outbound frame transmissions. Research paper [112] describes scheduling algorithms in detail.

The bridging mechanism defined in IEEE 802.1Q is not activated during this phase. Switches only forward frames based on the pre-defined schedule and no MAC address checking is performed. As only one frame is sent on one port at any time instance, no queueing delay is possible in switches. As a result, time-triggered communications require proprietary hardware to accept the schedule.

On the other hand, *processing on-the-fly* offers unique advantages. It requires much less effort in terms of planning as compared to the static TDMA approach, regardless of the topological complexity. The transmission is only initiated by the controller. The controlled devices only act passively as they are not permitted to start a data transmission. The process output is extracted from the frame and saved in the DPRAM area of the device. Simultaneously, the input data from the same device are inserted into the frame as the frame passes through each device. This results in simple but fast data exchange between the controller and the devices, as displayed in Figure 4.4.b.

4.3.3 Best-effort Channel

The RT and NRT traffics are allowed to appear during the BE phase. HRT frames which arrive at a switch are not accepted during this phase. Consequently, this phase does not

4.4.a:

4.4.b:

Figure 4.4: Space-time diagram of transmissions using (a) IFP (with slipstreaming effect); (b) SFP.

require any special hardware to be implemented and existing Ethernet infrastructures (Ethernet controller, switches, etc.) can be used. The bus scheduling in the BE phase can be freely defined. In other words, raw Ethernet frames and UDP messages can be sent on either cyclic or event-driven basis. This enables applications the flexibility to perform data exchange between controllers and devices.

RT frames are always marked with a VLAN tag (IEEE 802.1Q). The forwarding of frames during the BE phase shall be performed according to VLAN priorities defined in IEEE 802.1D [127]. Although the RT traffic enjoys higher priority compared to the NRT, it may still suffer from low but unpredictable delays [70]. If there are still RT frames yet to be sent in the queue at the end of this period, the transmissions are suspended until the next upcoming BE phase.

Sensor devices do not start sampling based on the global clock during the BE phase as they are not synchronized with each other, i.e., sensor tasks run freely at the expiry of the local timers.

4.3.4 Reserved Channel

The reserved phase is an optional safety margin between the BE and the GTD phase. It is defined to avoid sending of RT or NRT frames after starting the next scheduled GTD phase. This is to guarantee that no HRT frame transmission is to be blocked in the next GTD phase and consequently no jitter exists among the bus cycles. It can be also used to compensate the computational time needed by the controller in order to ensure that all process data required for the GTD and the BE phase are ready for transmission.

4.4 Wireless Extension

4.4.1 Distributed Coordination Function Mode

The frame structure presented in Figure 4.1.b (without subframes) can be applied for 802.11 DCF-based extension. The medium access scheme of DCF mode is shown in Figure 4.5. In DCF mode, transmissions are separated using inter-frame gaps. Wireless nodes sense the channel to determine if another node is transmitting through the network.

When wireless medium is detected to be not busy, the node waits for a DIFS (DCF inter-frame spacing) time. During this time, if the media is still sensed to be free, the station can transmit DATA frame. On the other hand, if the medium is busy during DIFS, an exponential backoff procedure is introduced. Similarly, the backoff procedure is also invoked when wireless media is busy in the first place.

The backoff procedure defers transmission from the node by firstly waiting until the end of current transmission. After the existing transmission is over, the node waits for an additional DIFS followed by a backoff delay. The backoff delay can be expressed as $T_{bof} = U[0, CW - 1] \times T_{slot}$. A random variable $U[0, CW - 1]$ is uniformly distributed within interval $[0, CW - 1]$ where CW is called backoff window or contention window. T_{slot} is the basic time unit slotted by IEEE 802.11 to sense data transmission taken place by any node. The initial CW is set to $CW = 1$. CW is doubled, before reaching a given maximum value in case of collision as two or more nodes decrease their backoff timers to 0 at the same time. The backoff timer decreases itself if the medium is sensed to be idle consistently for a DIFS. On the other hand, the timer is frozen if transmission is detected in the channel, and the countdown is only resumed when channel is detected as idle again for a DIFS. The DATA packet transmission can be initiated from a node when the timer of the node is expired.

Each frame transmission has to be explicitly acknowledged by the destination node. An ACK packet is sent to the source node after a SIFS (short inter-frame spacing) immediately following the successful reception of the DATA frame. SIFS is defined to be shorter than DIFS so that ACK packet is sent with priority. In case ACK is not received by the source within a certain time, the transmission is deemed failed and the source increases exponentially the CW and invokes another transmission after waiting for a backoff time. This attribute is considered in the network performance analysis in Chapter 5.

4.4.2 Point Coordination Function Mode

Linking wired segment to an 802.11 PCF-based sub-network is possible via the summation-frame structure illustrated in Figure 4.1.b. The summation-frame is used on the wired segment to transport data required on the wireless side from the controller to the bridge,

Figure 4.5: Medium access of DCF mode

Figure 4.6: Medium access of PCF mode

and vice versa. The bridge unpacks the summation-frame and sends the subframes to the wireless nodes according to the message scheduling in PCF mode.

PCF offers contention-free access to the medium using a round-robin algorithm as shown in Figure 4.6. It requires a bridge or an AP to be present to act as a point coordinator. The wireless devices within radio coverage are cyclically polled by the point coordinator and assigned with time slots for data transmission. This enables stable and predictable data communication even the bandwidth is intensively used in the radio network. Unfortunately, PCF is not quite commercially available as it can be only suitable to industrial applications where determinism is highly needed. Industrial PCF or iPCF is introduced by Siemens as a proprietary variant of deterministic PCF [32].

4.4.3 Extension at Physical Layer

The wireless extension at the physical layer requires same MAC scheme on both sides of the wired and wireless sub-networks. The two segments can be interconnected via a repeater which converts different communication media or signalling techniques. This is

Figure 4.7: System structure of bridge-based wireless extension

possible with transitional fieldbus protocols such as PROFIBUS. For example, literature [128] introduced cut-through forwarding devices which relates the bit-rate requirements in the radio segment. This allows the fieldbus data-link (FDL) layer to operate on different physical layers, i.e., RS-485 on the wired part and DSSS on the wireless part.

However, this is typically not possible with Ethernet-based networks as the data-link layer of standard Ethernet or modified Ethernet is varied from IEEE 802.11 (a/b/g/n/ac) or IEEE 802.15.4. Consequently, solutions should be explored on higher layers such as the data-link or the application layer.

4.4.4 Extension at Data-link Layer

The layer-2 protocols and services of both segments may not be necessarily identical but should be at least similar. It is not possible to interconnect sub-networks with very distinct properties (e.g., addressing methodology and MAC scheme) in a hybrid network. A bridge or switch is used to, respectively, interconnect two or more than two networks at layer-2. An AP can be applied as a bridge to form a hybrid wired/wireless real-time networks as illustrated with Figure 4.7.

4.4.4.1 Guaranteed Channel

The MAC mechanisms between $\gamma \in \{\text{HRT_IFP}, \text{HRT_SFP}\}$ and $\gamma = \text{WLAN}$ are completely different. The MAC schemes adopted in wireless protocols (e.g., DCF and PCF) cannot cope with the hard real-time characteristics of the GTD phase. It is, however, theoretically possible to incorporate IEEE 802.15.4-based wireless cycles in the GTD phase. This

requires the usage of TDMA in the wireless protocol and additionally a global clock to synchronize all wired devices and wireless nodes in the entire network. However, this is not practically feasible as the wireless network significantly slows down the bus cycle for the wired devices. The bus cycle of the wired part is typically in the range of 100 μ s to 1 ms [72]. Slot time is fixed at 10 ms with WirelessHART and configurable from 6 ms to 250 ms with ISA100.11a to ensure contention-free transmission [129]. It is therefore nearly not possible to include wireless cycles during the GTD phase, especially when the number of wireless nodes can be large in the AGV-based system. Additionally, the extension scheme requires global clock synchronization between wired and wireless parts which can be quite difficult with the indeterministic internal processing delays in the AP.

4.4.4.2 Best-effort Channel

The BE bandwidth can be utilized to carry out RT communications for wired devices and/or implement wireless extension. There are two solutions to incorporate wireless traffic in the bus cycle during the BE phase.

A polling-based message scheduling scheme can be realized to achieve wireless extension under the DCF mode. A RT frame issued by the controller travels across the bridge, gets encapsulated in a IEEE 802.11 PDU and reaches the addressed wireless node. The wireless node responds with an inbound frame which is to be received by the controller. The uncertainty caused by the nature of DCF can be reduced by the polling mechanism adopted in the BE phase as wireless traffic is kind of scheduled in this way as shown in Figure 4.8. It should be noted that Figure 4.8 only shows the data communication on the wireless sub-network. There are also data delivery for the controlled devices before and/or after polling wireless nodes.

The second solution adopts the frame structure proposed in Figure 4.1.b to achieve PCF-based wireless extension. The message scheduling of this solution is presented by Figure 4.9. The data prepared for the wireless nodes is packed by the controller into the *data* field on layer-2. The bridge receives the IEEE 802.3 frame, unpacks it into subframes, and transmits the subframes to the wireless destination node as indicated in the SF header of each subframe. As PCF is a contention-free scheme, no simultaneous access to the wireless medium is expected on the wireless segment. This can be helpful to achieve deterministic bus cycle and end-to-end delay through the BE channel.

Figure 4.8: Space-time diagram of DCF-based wireless extension

Figure 4.9: Space-time diagram of PCF-based wireless extension

4.4.5 Extension at Application Layer

There are no practical constraints to implement wireless extension at the application layer. Interconnecting wired and wireless segments at layer-7 is possible through a gateway. As the PDU is transferred at the application layer, there is in practice no restriction on the type of protocols running on each side of the gateway. The protocols and services on both sides of the segments can be very distinct from each other.

The gateway presented in Figure 4.10 incorporates a network controller for wired segment, a wireless RF module (802.11 or 802.15.4) and a shared microcontroller. The network controller may need to incorporate special hardware components to handle HRT frames. One side of the gateway act as an EtherCAT slave, a PROFINET device, or an Ethernet POWERLINK node, while the other side functions as a ZigBee coordinator, a WirelessHART/ISA100.11a gateway or a point coordinator (IEEE 802.11 PCF). As protocol conversion is realized at the application layer, the extension scheme is generic and in principle can be applied to all wired and wireless IIoT technologies. As shown in Figure 4.10, the cycle on the wired and wireless sub-networks are in a way decoupled. This can be helpful to maintain the consistency in the real-time capabilities of both sides.

Figure 4.10: System structure of gateway-based wireless extension

The real-time characteristics of the wired segment are less influenced by the wireless part as compared with the case of bridge-based extension.

4.4.5.1 Guaranteed Channel

With the gateway, the communication cycles from the wired and wireless segments are decoupled. The advantage is that the cycle time of the wired segment is not much affected by the wireless segment. It is possible to achieve fast cycle in the GTD phase. Since the real-time capability on the wireless network is significantly poorer compared to the wired network, it makes no much sense to exchange data between wired and wireless segments using the GTD phase. Instead, it can be done in the BE phase using RT frames.

4.4.5.2 Best-effort Channel

An outbound RT frame can be used to exchange all necessary process data with the wireless nodes via the gateway. The response from the wireless nodes can be collected at the gateway and sent back to the controller. The data exchange between the wired and the wireless segments can be done based on either cyclic or event-driven mechanism on the gateway. The cyclic mechanism is already implemented by the PCF protocol in IEEE 802.11 or TDMA in IEEE 802.15.4 where the gateway periodically talks with the wireless nodes via polling or at pre-defined time instances, respectively. The gateway initiates the communication cycles on the wireless sub-network. On the other hand, the controller also

communicates with all wired devices every bus cycle through the BE phase. Both cycles run independently on two sides of the gateway without any kind of synchronization.

Alternatively, data can be exchanged on event basis between the wired and the wireless segments. When the controller wants to send the data to a wireless node, it issues a RT frame via the BE channel. The frame is received by the gateway. The reception of the RT frame is an event to trigger the gateway to start transmission of an 802.11 or 802.15.4 packet. With the DCF mechanism, the packet is transmitted to the addressed wireless node. When a wireless node needs to pass information to the controller, it transfers the data firstly to the gateway. The gateway stores the data in its buffer and waits for the next available BE phase to send it back to the controller.

4.5 Conclusion

This chapter defines a communication scheme combining benefits from several existing standard, including EtherCAT, PROFINET and Ethernet POWERLINK. The network comes with several phases to achieve various real-time classes. The GTD bandwidth enables real-time networked control applications in the wired segment. The BE channel, on the other hand, is suitable for conventional DCS and can be also used for wireless extension on the data-link or the application layer. The extension approach is fully compatible with IEEE 802.11 DCF or PCF mode on the wireless segment. The real-time capability, scalability and interoperability of the solution is to be analyzed in typical industrial control systems in the next chapter.

Chapter 5

Analysis of Communication Quality of Service

As discussed in the literature review, bus cycle time is critical to the traditional DCS, whereas network delay and frame loss are the major challenges in the NCS. Prior to the controller design, the characteristics of communication QoS such as bus cycle, data delay and data loss (in terms of packet dropout rate or reliability) should be evaluated. And the evaluation results should serve as the input to the simulator of the control performance.

Queueing theory plays a central role in modelling and evaluating delays of random access networks. However, in controlled access networks such as real-time Ethernet, communication buffers are eliminated to get rid of the non-deterministic queueing delays in order to deal with time-critical (and cyclic) data delivery. As a result, old packets are discarded for cyclic data communication, and senders and receivers perform non-consuming write and read from the network. Because of such configurations, delay modelling is mathematically simpler for real-time Ethernet in comparison with random access networks. Nevertheless, evaluations on network QoS remain a must in order to verify the system design in industrial automation.

5.1 Cycle Time

Bus cycle time $\mathcal{D}_e$ is one of the key communication performance indicators. It can be treated as the execution time of J_e which is the communication time needed for the controller to exchange process data with all devices/nodes once. It is composed of data

transmission time T_{frm} , hardware latencies such as forwarding delays and cable delays (ignorable), and response time (only for polling-based network). Transmission time for the frame to be sent at device n can be obtained by $T_{frm}(n, \gamma) = 8f(n)/C(\gamma)$, where $C(\gamma)$ is the transmission rate of protocol γ . The minimum achievable cycle time of the network can be computed as

$$\mathcal{D}_e = \alpha(\mathcal{D}_e^{\text{GTD}} + \mathcal{D}_e^{\text{RSVD}}) + \beta\mathcal{D}_e^{\text{BE}}, \quad (5.1)$$

where $\alpha \in [0, 1]$ indicates if the GTD phase is planned in the system since it is needed only for high-performance control. $\alpha = 0$ means that the GTD phase is not used by the application. Similarly, $\beta \in [0, 1]$ represents if the BE phase is scheduled, especially for wireless communication. $\mathcal{D}_e^{\text{GTD}}$, $\mathcal{D}_e^{\text{BE}}$ and $\mathcal{D}_e^{\text{RSVD}}$ represent partial cycle time for the GTD, the BE and the RSVD phase, respectively. For the NCS where control variables are exchanged over cyclic networks, the minimum achievable cycle time determines the maximum effective sampling rate of the system, i.e., update rate of sensor data at the controller can never be faster than the bus cycle.

5.1.1 Guaranteed Phase

The bandwidth required for the GTD channel can be estimated as

$$\mathcal{D}_e^{\text{GTD}}(\gamma) = \begin{cases} T_{frm}(\mathbb{D}, \gamma) + 2 \max(\mathcal{N}_{swt}(n))T_{fwd_swt}(\gamma), & \forall n \in \mathbb{D} \quad \gamma = \text{HRT_SFP}, \\ \sum_{n \in \mathbb{D}} T_{frm}(n, \gamma) + \min(\mathcal{N}_{swt}(n))T_{fwd_swt}(\gamma), & \forall n \in \mathbb{D}, \quad \gamma = \text{HRT_IFP}, \end{cases} \quad (5.2)$$

where $T_{frm}(\mathbb{D}, \gamma = \text{HRT_SFP})$ denotes the transmission time required for the summation-frame which packs payloads from every device, $\mathcal{N}_{swt}(n)$ refers to the number of switches between the controller and the wired device n , and $T_{fwd_swt}(\gamma = \text{HRT})$ refers to the forwarding delays at the (internal) switches and can be approximated to $0.6 \mu\text{s}$ [16]. It is noticed that $2 \max(\mathcal{N}_{swt}(n))T_{fwd_swt}(\gamma)$ represents the ring latency in the case of SFP. In case of fragmentation of data into multiple frames, the ring latency only applies once as the next transmission can be issued as soon as the current transmission is finished, i.e., the controller does not need to wait until it receives the current frame travelled back through the ring. With IFP, data transmission can be scheduled immediately after the previous one because of the time-triggered scheme.

As discussed in Chapter 4, it is not suitable to use the GTD phase for wireless extension due to mismatched characteristics between the wired and the wireless segments. Consequently, the modelling here does not consider time required for wireless processes. That is to say, only cyclic HRT frames are scheduled and considered during the GTD phase.

5.1.2 Best-effort Phase

The BE bandwidth can be utilized by the wired (optional) and the wireless segment and its bandwidth can be expressed as a combination of both:

$$\mathcal{D}_e^{\text{BE}} = \mathcal{D}_e^{\text{BE-WD}}(\mathbb{D}) + \mathcal{D}_e^{\text{BE-WL}}(\mathbb{W}). \quad (5.3)$$

Both $\mathcal{D}_e^{\text{BE-WD}}$ and $\mathcal{D}_e^{\text{BE-WL}}$ can be optional, depending on whether the wired devices and/or wireless nodes are participated in communication through the BE channel. On one hand, devices can already exchange data cyclically with the controller via the GTD channel where the QoS can be better provided. This assumption will generally apply during delay analysis later in this chapter. On the other hand, not all applications ask for wireless communication.

Assuming the inbound and outbound process data on the network are identical, the minimum achievable cycle time required by the wired devices during the BE phase can be estimated as the following

$$\begin{aligned} \mathcal{D}_e^{\text{BE-WD}}(\mathbb{D}) = \sum_{n \in \mathbb{D}} (2T_{frm}(n, \gamma) + 2\mathcal{N}_{swt}(n)T_{fwd_swt}(\gamma) + T_{rsp_dev}(n)) \\ + \text{sizeof}(\mathbb{D})T_{rsp_ctrl}, \quad \gamma = \text{RT}, \end{aligned} \quad (5.4)$$

where $\text{sizeof}(\mathbb{D})$ returns the number of elements in set $\mathbb{D}$. Symbols T_{rsp_ctrl} and T_{rsp_dev} mean the response time required by the controller and the devices and can be approximated, according to Ethernet POWERLINK, to $1 \mu\text{s}$ and $8 \mu\text{s}$, respectively [73]. T_{fwd_swt} can be approximated to $3 \mu\text{s}$ which is the typical delay of a *cut-through* switch when $\gamma = \text{RT}$ [16, 124].

On the other hand, the bandwidth required for the wireless data can be calculated as

Figure 5.1: Transmissions in wireless segment during BE phase

$$\mathcal{D}_e^{\text{BE-WL}}(\mathbb{W}) = \sum_{n \in \mathbb{W}} (2T_{frm}(n, \gamma) + T_{rsp.br}(n)) \quad (5.5)$$

$$+ \text{sizeof}(\mathbb{W})(T_{rsp.ctrl} + 2\mathcal{N}_{swt}(br)T_{fwd.swt}(\gamma) + 2T_{fwd.br}), \quad \gamma = \text{RT},$$

where $\mathcal{N}_{swt}(br)$ refers to the number of switches on the transmission path between the controller and the bridge. $T_{fwd.br}$ is the bridge delay and can be approximated to $10 \mu\text{s}$ as discussed earlier. $T_{rsp.br}(n)$ denotes the response time of the bridge for wireless node n . As illustrated in Figure 5.1, it consists of two DATA frames, two ACK frames, two DIFS and two SIFS, and can be computed accordingly as

$$T_{rsp.br}(n) = 2(T_{frm}(n, \gamma = \text{WLAN}) + T_{ack}) + T_{ifs} + T_{bof}(n), \quad \forall n \in \mathbb{W}, \quad (5.6)$$

where $T_{frm}(n)$ is the transmission time for wireless node n , either at the bridge or the wireless node itself. Again, it is assumed that the inbound and the outbound data length is the same on the wireless network. It can be obtained as $T_{frm}(n) = 27.14 \times 10^{-6} + 8f(n)/C(\gamma = \text{WLAN})$. T_{ack} is the transmission time required for the ACK frame and can be estimated to $24.1 \mu\text{s}$. T_{ifs} refers to the total inter-frame space needed, i.e., two DIFS ($2 \times 28 \mu\text{s}$) and two SIFS ($2 \times 10 \mu\text{s}$). According to [20], the backoff time T_{bof} can be written as a uniform random value ranging from 0 to $CW_{\min}T_{slot}$:

$$T_{bof}(k) = U[0, CW_{\min}T_{slot}](k). \quad (5.7)$$

The fixed part of $T_{rsp.br}$ includes the PHY preamble, PHY header and MAC header of the DATA frame, the ACK frame, and the inter-frame spaces. It can be estimated to around $178.62 \mu\text{s}$ according to Table 5.1 collected from [130–132]. As a result, the response time at the bridge can be rewritten as $T_{rsp.br}(n) = 178.62 \times 10^{-6} + 16f(n)/C(\gamma = \text{WLAN}) + T_{bof}(n)$.

Table 5.1: Communication parameters in IEEE 802.11g

Parameter	Value
Data rate C	54 Mbps
Basic rate	6 Mbps
T_{slot}	9 μs
PHY preamble and header	24 Bytes / 22 μs
DIFS	28 μs
SIFS	10 μs
MAC header	28 Bytes / 4 μs
ACK data	14 Bytes / 2 μs
CW_{min}	15
CW_{max}	1023

5.1.3 Reserved Phase

In the case of $\gamma = \text{HRT_SFP}$, the computation from the controller can only start after receiving the summation-frame. It is only meaningful to send the next summation-frame after the completion of the computation. Therefore, the RSVD phase is used to compensate the controller delay, i.e., $\mathcal{D}^{\text{RSVD}} \geq D_c$. Otherwise, this phase mainly serves as a safety margin to avoid jitter on the bus cycle, and the minimum duration of the RSVD phase should be at least greater than the transmission time of 1526 bytes which is the maximum length of an Ethernet frame. Consequently, $\mathcal{D}^{\text{RSVD}} \geq 1526 \times 8 / C(\gamma = \text{NRT}) = 122 \mu s$. In the case where this phase is needed as a compensation for the computational time from the controller, the duration of the phase should be extended accordingly.

5.2 End-to-end Delay on OSI Model Layers

An end-to-end delay between sender and receiver normally consists of hardware latencies, stack delays and optionally application execution time [69, 133].

The delivery time is defined in IEC 61784-2 as the time required for a packet to be transmitted from the source to the destination node [69]. It is equivalent to end-to-end delay which includes S-C delay τ_{sc} , computational delay from controller τ_{ctrl} and C-A delay τ_{ca} in the context of NCS. It is stated in [69] that an end-to-end delay is composed of the traversal time from each layer of the communication (presented in Figure 2.1), i.e., τ_{net} , τ_{sta} and τ_{app} from the network, the stack and the application layer, respectively. As a result, the end-to-end delay τ can be simply formulated with

$$\begin{aligned}\tau(n, k) &= \tau_{sc}(n, k) + \tau_{ctrl}(k) + \tau_{ca}(n, k) \\ &= \tau_{net}(n) + \tau_{sta}(k) + \tau_{app}(k).\end{aligned}\tag{5.8}$$

The network delay $\tau_{net}(n)$, which eventually contributes to τ_{sc} and τ_{ca} , in most cases varies from device to device. In other words, different wired devices or wireless nodes can experience various S-C and C-A delays, depending on the position of the device or node located in the network topology. Additionally, τ_{sc} and τ_{ca} are time-varying parameters because of stack execution time $\tau_{sta}(k)$.

5.2.1 Network Delay

Network delay τ_{net} refers to the communication time induced by physical and hardware layer such as cable/radio, transceiver, network controller, and so on. It is made up of transmission time, node latencies, propagation delay, and optionally the access delay. Among these delay components, the transmission delay and the propagation delay are normally static (constant and predictable) once the network is designed and deployed. However, access delay can be variable. It refers to the waiting time of a frame in the sending or the receiving buffer of the DPRAM to access the network.

The calculation of network delay mainly depends on communication parameters, e.g., number of devices, medium access method, cable length, traffic load, etc. It can be analyzed using time-space diagrams shown in Figure 5.2 (without wireless segment) and Figure 5.3 (with wireless segment) which is to be discussed in detail later in this chapter.

5.2.2 Stack Delay

Communication stack implements the protocols and services from the data-link layer to the application layer, in the form of either hardware or software. Correspondingly, stack delay τ_{sta} refers to the traversal time of the stack. For IIoT technologies, communication stack on layer-2 are mostly realized by hardware, e.g., FPGA, ASIC or SoC [44, 134, 135]. The resulting delay is highly related to the hardware platform on which the stack is executed. Communication stack for layer-3 and above, such as TCP/IP, is realized by software [114, 115]. As the network and the transport layer are also involved in the IP-based communication, stack delay is expected to be relatively high in EtherNet/IP and

5.2.a:

5.2.b:

Figure 5.2: Gantt chart of a wired network using (a) IFP; (b) SFP

Figure 5.3: Gantt chart of a hybrid network

Modbus TCP/IP. The delay is dependent on the computational power of the CPU. Due to the difficulty to formulate the stack delays into mathematical form, τ_{sta} is normally obtained by experiments and can be offered by the stack developer or owner. Different from τ_{net} , τ_{sta} is irrelevant to communication parameters. Its performance analysis therefore can take place on device-level. In [136], stack delays of EtherNet/IP, EtherCAT, and PROFINET IRT were experimentally measured on the same hardware platform.

5.2.3 Application Delay

Application delay τ_{app} refers to the any computational time required on layer-7, for example, application profiles, sensor signal processing, controller algorithms (e.g., PID), etc. Application profile such as CANopen, PROFIdrive, CIP, etc., specifies process data, configuration parameters, and diagnostic information needed by the control system. It is not mandatory in industrial communication and hence the corresponding delays induced are not considered in this thesis. Application delay is not applicable to hubs, bridges, routines, switches as these devices do not run on the application layer. However, controller delays caused by computations from the application layer should be considered in the calculation of end-to-end delays from sensors to actuators.

5.3 Sensor-to-Controller Network Delay

5.3.1 Network Delay of Wired Network

As discussed in Chapter 3, sensors perform periodic sampling every P_s , driven by either global clock or local timer. The measurements from the sensor are processed by algorithms which take a duration of D_s . After that, the processed measurements are stored in the shared DPRAM, ready to be collected by the controller via the network task J_{e-sc} . The start time of sensor with index n at the k -th control cycle can be obtained as

$$S_s(n, k) = S_s(n, k = 0) + kP_s, \quad (5.9)$$

where $S_s(n, k = 0)$ is the initial start time of $J_s(n)$. If clock synchronization is applied, the jitter of $S_s(n, k)$ should be less than $1 \mu s$. Otherwise, larger jitter can be expected if local timer is used to trigger $S_s(n, k)$. Whether to use global clock or local timer also affects

the start time of J_{e_sc} . With clock synchronization, the processes involved in the inbound communication is more predictable and deterministic [14]. The network controller is able to raise an interrupt or signal to the CPU at certain time instances. Input shift delays can be defined in a way that the measurements from sensors are available right before the arrival of summation-frame in case of $\gamma = \text{HRT_SFP}$ or the departure of the S-C frame in case of $\gamma = \text{HRT_IFP}$, so that the access delay can be minimized or removed. Otherwise, in case of free-running sensors driven by local timers, additional access delay of $[0, P_s)$ can be expected. Consequently, it is preferred to use the globally synchronized clock to drive $S_s(n, k)$ to minimize the variability of the latency between J_s and J_{e_sc} . In this case, the start time of S-C communication can be written as

$$S_{e_sc}(n, k) = S_s(n, k) + D_s(n). \quad (5.10)$$

For $\gamma = \text{HRT_SFP}$, the controller task is triggered only when the summation-frame is travelled throughout the ring. In case of $\gamma = \text{HRT_IFP}$, the controller's computation for actuator n can be executed as soon as the inbound frame from sensor n is received. Once the inbound frame is received and the payload from sensor is stored in the DPRAM of the network card of the controller, the network card can raise an interrupt to the controller CPU in order for J_c to compute set-point for actuator based on the sensor measurement read from the DPRAM. The start time of J_c can be obtained as

$$S_c(n, k) = S_{e_sc}(k) + D_{e_sc}^{\text{GTD}}(n), \quad (5.11)$$

where $D_{e_sc}^{\text{GTD}}(n)$ represents the network delay from $J_s(n)$ to J_c using communication services provided by the GTD bandwidth. The access time to the DPRAM for data copy is ignored. It can be obviously known from Figure 5.2 or 5.3 that $D_{e_sc}^{\text{GTD}}(n) < \mathcal{D}_e, \forall n \in \mathbb{D}$. $D_{e_sc}^{\text{GTD}}(n)$ can be computed as

$$D_{e_sc}^{\text{GTD}}(n) = \mathcal{N}_{swt}(n)T_{fwd_swt}(\gamma) + \begin{cases} 0, & \forall n \in \mathbb{D}, & \gamma = \text{HRT_SFP}, \\ \sum_{m \in \check{\mathbb{D}}} T_{frm}(m, \gamma), & \forall n \in \mathbb{D}, & \gamma = \text{HRT_IFP}, \end{cases} \quad (5.12)$$

where $\check{\mathbb{D}} = \{n, n+1, \dots, N_{dev}\}$ is a subset of $\mathbb{D}$, and $\mathcal{N}_{swt}(n)$ denotes the number of switches on the forwarding path for sensor n . Finally, the network delay on the S-C path

$\tau_{sc,net}(n, k)$ can be computed with virtual time-stamps by measuring the time interval from the end of $J_s^n(k)$ to the start of $J_c(k)$:

$$\tau_{sc,net}(n, k) = S_c(k) - (S_s(n, k) + D_s(n)) = D_{e-sc}^{GTD}(n). \quad (5.13)$$

5.3.2 Network Delay of Wireless Network

To simplify mathematical modelling, sensors connected on the wired or the wireless segments are assumed to communicate with the controller using only the GTD or the BE phase, respectively. The network delays between the controller and the wireless sensors are to be discussed here. The task timing diagram of the wireless extension is shown in Figure 5.3 where both the GTD and the BE phases are presented.

During the BE phase, the sensor devices are still time-triggered, however, with local timer instead of global clock. The sampling can be done as fast as every $D_s(n)$. The measurement from the sensor n needs to wait for an access delay of $U[0, D_s(n)]$ in the buffer for the transmission opportunity. After that, the sensor data are embedded in the RT frame and transmitted to the controller through the BE channel.

The S-C delays can be calculated by the sum of latencies of the wired segment, bridge and the wireless segment, i.e.,

$$\tau_{sc,net}(n, k) = \mathcal{D}_e^{BE-WL}(\mathbb{W} - \tilde{\mathbb{W}}) + D_{e-sc}^{BE-WD}(n) + T_{fwd.br}(k) + D_{e-sc}^{BE-WL}(n, k), \forall n \in \mathbb{W}. \quad (5.14)$$

where $\tilde{\mathbb{W}} = \{1, 2, \dots, n\}$ represents a subset of $\mathbb{W}$. $D_{e-sc}^{BE-WD}(n)$ denotes the network delays on the wired sub-network for wireless node n , and it can be computed as

$$D_{e-sc}^{BE-WD}(n) = T_{frm}(n, \gamma) + \mathcal{N}_{swt}(br)T_{fwd.swt}(\gamma), \quad \gamma = \mathbf{RT}, \forall n \in \mathbb{W}. \quad (5.15)$$

The network delay of the wireless segment $D_{e-sc}^{BE-WL}(n, k)$ includes the time for a SIFS, an ACK frame, a DATA frame $T_{frm}(n, \gamma = \mathbf{WLAN})$, $\forall n \in \mathbb{W}$, a DIFS and a backoff procedure:

$$D_{e-sc}^{BE-WL}(n, k) = U[0, D_s](k) + T_{frm}(n, \gamma = \mathbf{WLAN}) + T_{ack} + T_{ifs} + T_{bof}(k). \quad (5.16)$$

The SIFS, the ACK and the DIFS can be approximate to total $62.17 \mu s$ as per Table 5.1.

5.4 Controller-to-Actuator Network Delay

5.4.1 Network Delay of Wired Network

Similar to the analysis of the S-C link, it is assumed that all actuators are triggered by the HRT frames received from the GTD channel. Without integration of wireless sub-network, the task $J_{e-ca}(n)$ is triggered at the expiry of the deadline for controller task J_c . Of course, before the expiry time of P_c , the controller should have completed the computation for actuator n and the result (i.e., set-point of actuator n) should have been copied to the DPRAM of the network card (see Figure 5.2). The start time of J_{e-ca} can be described as

$$S_{e-ca}(k) = S_c(k) + P_c. \quad (5.17)$$

where P_c is the deadline for controller task J_c . Again, data copy time from the controller's RAM to the DPRAM of the network controller is neglected. On the other hand, it is possible in the scenario described by Figure 5.3 that the controller has finished its computation before the end of the RSVD phase. In this case, J_{e-ca} can start as soon as the current network cycle is completed.

$$S_{e-ca}(k) = S_{e-ca}(k-1) + \mathcal{D}_e. \quad (5.18)$$

The actuator task is event-driven, and therefore is activated when the outbound frame is received via the communication service task J_{e-ca} . This behaviour can be described as

$$S_a(n, k) = S_{e-ca}(k) + D_{e-ca}^{\text{GTD}}(n), \quad (5.19)$$

where $D_{e-ca}^{\text{GTD}}(n)$ is communication service time for actuator n or the network delay from controller to actuator n . It does not necessarily need a full bus cycle for the actuators to receive data from the network after a bus cycle is kicked off (see Figure 5.2), i.e., $D_{e-ca}(n) \leq \mathcal{D}_e, \forall n \in \mathbb{D}$. Each actuator may experience delay differently, depending on the node index n .

In case that the SFP is adopted, the frame packing process data of all devices travels along C-A and thereafter S-C links in serial. As a result, the summation-frame transmission time, denoted as $T_{frm}(\mathbb{D})$, applies to all actuators. Alternatively, the network card

of the controller issues N_{dev} frames during $\mathcal{D}_e$. Therefore, the calculation of $D_{e-ca}(n)$ can be expressed as

$$D_{e-ca}^{GTD}(n) = \mathcal{N}_{swt}(n)T_{fwd_swt}(\gamma) + \begin{cases} T_{frm}(\mathbb{D}), & \forall n \in \mathbb{D} & \gamma = \text{HRT_SFP}, \\ \sum_{m \in \tilde{\mathbb{D}}} T_{frm}(m), & \forall n \in \mathbb{D} & \gamma = \text{HRT_IFP}, \end{cases} \quad (5.20)$$

where $\tilde{\mathbb{D}} = \{1, 2, \dots, n\}$ is a subset of $\mathbb{D}$. The actuation takes place in D_a after its network controller receives data from the bus. The network delay experienced by the communication service on the C-A link can be calculated as

$$\tau_{ca,net}(n, k) = S_a(n, k) - (S_c(k) + P_c) = D_{e-ca}^{GTD}(n). \quad (5.21)$$

5.4.2 Network Delay of Wireless Network

The communication between the controller and actuator n is scheduled right after actuator $n - 1$, according to the polling scheme designed in Chapter 4. The time required to poll actuator set $\{1, 2, \dots, n - 1\}$ can be described as $\mathcal{D}_e^{\text{BE-WL}}(\mathbb{W} - \check{\mathbb{W}})$ where $\check{\mathbb{W}}$ refers to the subset $\{n, n + 1, \dots, N_{wln}\}$ of $\mathbb{W}$. The total C-A delay can be calculated by the sum of delays of the wired segment, bridge device and the wireless segment:

$$\tau_{ca,net}(n, k) = \mathcal{D}_e^{\text{BE-WL}}(\mathbb{W} - \check{\mathbb{W}}) + D_{e-ca}^{\text{BE-WD}}(n) + T_{fwd_br}(k) + D_{e-ca}^{\text{BE-WL}}(n, k), \forall n \in \mathbb{W}. \quad (5.22)$$

The actuation activity $D_a(n)$ is triggered as soon as actuator n receives the data frame through the BE channel. The C-A delays of the wired segment can be obtained as

$$D_{e-ca}^{\text{BE-WD}}(n) = T_{frm}(n, \gamma) + \mathcal{N}_{swt}(br)T_{fwd_swt}(\gamma), \quad \gamma = \text{RT}, \forall n \in \mathbb{W}. \quad (5.23)$$

The wireless segment delay includes the time for transmitting a DATA frame and a DIFS. The actuator does not need to wait for the completion of ACK frame as the reception of the C-A frame is known to each actuator. Hence, the C-A delays of the wireless part can be written as

$$D_{e-ca}^{\text{BE-WL}}(n, k) = T_{frm}(n, \gamma = \text{WLAN}) + T_{ifs}, \forall n \in \mathbb{W}. \quad (5.24)$$

5.5 Component Delay

5.5.1 Controller Delay

From Figures 5.2 and 5.3, it can be noticed that the controller cycle is in synchronization with the network cycle. The controller computation starts when the summation-frame or each individual frame is received via the GTD or the BE channel. The execution time D_c is the sum of computational time of all controlled plants (wired devices and wireless nodes) during the scan time of the controller, i.e.,

$$D_c = \sum_{n \in \mathbb{D}} (D_c^{\text{GTD}}(n) + D_c^{\text{BE-WD}}(n)) + \sum_{m \in \mathbb{W}} D_c^{\text{BE-WL}}(m), \quad (5.25)$$

where D_c^{GTD} , $D_c^{\text{BE-WD}}$, and $D_c^{\text{BE-WL}}$ denote the computational time for the controller to calculate the process data needed for wired devices through the GTD channel and the BE phase and wireless nodes through the BE channel, respectively. It should be highlighted that most of the OS running on the controller is preemptive, i.e., the calculation needed for the GTD phase should be assigned with higher priority than that for the BE phase.

Without assuming a powerful CPU on the controller where the execution time D_c can be neglected, the computational capability of the controller brings constraints in the system design and operation. The inequality $D_c \leq \mathcal{D}_e$ should hold so that the computation must be finished before the output gets the next opportunity for transmission. If there is difficulty satisfying this inequality due to some reasons, for example slow CPU on the controller or $\gamma = \text{HRT_SFP}$ without wireless extension, the RSVD phase can be expanded so that it can compensate the computational time required for the controller.

Controller delays τ_{ctrl} can be obtained as the time required for the controller to wait for the next communication cycle, i.e.,

$$\tau_{ctrl} = \begin{cases} \mathcal{D}_e^{\text{RSVD}} & (\gamma = \text{HRT_IFP or HRT_SFP}), \text{ without wireless extension} \\ \mathcal{D}_e^{\text{BE}} + \mathcal{D}_e^{\text{RSVD}} & (\gamma = \text{HRT_IFP or HRT_SFP}), \text{ with wireless extension} \\ \mathcal{D}_e^{\text{GTD}} + \mathcal{D}_e^{\text{RSVD}} & (\gamma = \text{RT or WLAN}), \text{ with wireless extension} \end{cases} \quad (5.26)$$

The controller delay can be treated as part of τ_{app} as it is caused on the application layer.

5.5.2 Bridge Delay

A bridge is typically implemented by an embedded system running a lightweight OS, forwarding data frames from the wired portion to the wireless portion of the hybrid network and vice versa [137, 138]. As presented from Figure 4.7, the forwarding delays introduced by bridges $T_{fwd.br}$ are incurred on the physical and the data-link layers. The bridge delay should be taken into account calculating the end-to-end delays of the hybrid network.

Performance evaluation on commercial AP has been performed in literatures [139, 140]. As suggested in [21], the bridge delay can be caused by latency and queueing. The bridge latency can be estimated to 5-10 μs [21]. On the other hand, the queueing delay is not considered in the thesis due to ordered message scheduling from the polling technique applied in the wireless extension during the BE phase.

5.5.3 Gateway Delay

Different from bridge device, delays of a gateway can be introduced on layer-1, layer-2 and layer-7 (see Figure 4.10). The gateway is more complex and expensive in the communication infrastructure as compared with the bridge [67]. The protocol conversion is mainly implemented by a piece of software which requires more resources such as CPU and RAM as compared to hardware-oriented implementation on the data-link layer. Consequently, the delay incurred at the gateway $T_{fwd.gw}$ is generally higher and more unpredictable compared to that at a bridge or switch device, i.e., $T_{fwd.gw} > T_{fwd.br}$.

5.6 Delay Model

The total end-to-end delay $\tau(n, k)$ can be computed using Formula (5.8) after all delay elements are obtained. τ is a time-variant parameter because of variable latencies during the communication, such as the stack delay and the access delay. Due to the controlled medium access adopted in the GTD phase as well as the BE phase, it is expected that the delay with respect to device n is bounded within its minimum and maximum values denoted as $\min(\tau(n, k))$ and $\max(\tau(n, k))$, respectively.

Next, a delay probability model is introduced to consolidate the characteristics of τ . Basically, the statistics of end-to-end delay τ can be concluded with a histogram as suggested in [141]. In other words, τ can be summarized by a discrete-time probability density function as follows:

$$\Pr_{\tau}(n, q) = [\Pr_{\tau}(n, 1) \quad \Pr_{\tau}(n, 2) \quad \cdots \quad \Pr_{\tau}(n, Q)], \quad (5.27)$$

where $\Pr_{\tau}(n, q)$ refers to the probability of $\tau(n) \in [\min(\tau(n, k)) + (q-1)\Delta\tau(n), \min(\tau(n, k)) + q)\Delta\tau(n)]$. Q means the number of bins defined for the histogram consolidating the delay probability. It can be obviously known that the larger the Q , the more accurate the delay model can present.

The delay model is to be provided to the control system design and verification as an input parameter. In practice, it is possible to obtain delay results from mathematical analysis, simulation or experiment from a standalone network performance analysis and store the histogram as a QoS database. The QoS database may contain other useful information such as cycle time and reliability model. This evaluation process will be elaborated in Chapter 7.

5.7 Communication Reliability

Communication error can be any type of information distortion or mishandling such as data corruption, repetition, interchanging, and loss [142]. The common sources of communication problems in industrial applications can be poor cabling, weak radio signals, faulty networking hardware, and especially electromagnetic interference which can introduced by electric and electronic devices. The case of packet dropout is discussed here. Packet dropout exists in lossy communication links, and is another significant factor when designing networked controllers as it may lead to stability issues [80, 83]. It can be found in both Ethernet-based and wireless networks.

5.7.1 Reliability on Wired Network

From [143], a communication error can happen on the links and the nodes in a network. In this analysis, only packet dropout caused by bit errors on the transmission link is considered. A packet is dropped when at least one bit is erroneous. It should be mentioned

that traditional *store and forward* switch performs data checking before forwarding the packet to the next node. *Cut-through* and *fast forward* switches, however, are not as reliable as *store and forward*. In order to minimize forwarding delays, packets are forwarded right after the destination sections have been decoded, without any cyclic redundancy check (CRC) checking. As a result, the corrupted data (with a wrongly coded CRC) are propagated to the destination node [144].

Several safety mechanisms (watchdog, CRC, and sequence number for example) can be defined to detect and locate communication errors on the wired segment. The frame check sequence *FCS* field is a 32-bit error-detecting CRC added to the IFP frame. On the other hand, each subframe contains a 2-Byte CRC in the case of SFP so that errors can be located on the subframe level. The packet dropout rate can be obtained with $\text{Pr}_e(n) = 1 - (1 - \text{Pr}_{ber}(\gamma))^{8f(n)}$ where $\text{Pr}_{ber}(\gamma)$ is the constant bit error rate (BER) for protocol γ . The reliability of the GTD channel R^{GTD} is defined as the probability that no communication error is caused by bit errors for all data transmissions during the GTD phase of a bus cycle. Consequently, the reliability $\mathcal{R}^{\text{GTD}}$ can be computed as

$$\mathcal{R}^{\text{GTD}}(\gamma) = \begin{cases} (1 - \text{Pr}_{ber}(\gamma))^{8f(\mathbb{D})}, & \gamma = \text{HRT_SFP} \\ \prod_{n=1}^{2N_{dev}} (1 - \text{Pr}_{ber}(\gamma))^{8f(n)}, & \forall n \in \mathbb{D}, \quad \gamma = \text{HRT_IFP} \end{cases} \quad (5.28)$$

Note that the reliability of the IFP is calculated as the product of reliabilities of individual frames on both S-C and C-A links.

5.7.2 Reliability on Wireless Network

Generally speaking, the wireless sub-network can suffer more setbacks than the wired sub-network in terms of reliability, as it is quite susceptible to packet loss caused by RF interference, weak signals, etc. Here the reliability is obtained only from the BER on the wireless link.

As the scheduling on the wireless segment is already moderated using polling mechanism, the modelling of the packet dropout rate provided in [145, 146] is no longer applicable here. Again, only the packet loss caused by the BER is considered. The reliability of the wireless segment in a bus cycle can be obtained as

$$\mathcal{R}^{\text{BE.WL}}(\gamma = \text{WLAN}) = \mathcal{R}^{\text{BE.WD}}(\gamma = \text{RT}) \prod_{n=1}^{2N_{wln}} (1 - \Pr_{ber}(\gamma = \text{RT})^{8f(n)}), \quad \forall n \in \mathbb{W}, \quad (5.29)$$

where $\mathcal{R}^{\text{BE.WD}}(\gamma = \text{RT})$ denotes the reliability of RT frames during a bus cycle, and it can be calculated by reusing Formula 5.28. The reliability $\mathcal{R}^{\text{BE.WL}}(\gamma = \text{WLAN})$ depends on the successful transmission of both RT and WLAN frames .

5.8 Conclusion

The communication scheme suggested in Chapter 4 has been analyzed with mathematical modelling on QoS indicators including cycle time $\mathcal{D}_e$, end-to-end delay τ and reliability $\mathcal{R}$. For cycle time, the bandwidth required for the GTD, the BE and the RSVD phases have been studied. The network delays on the C-A and S-C links for both wired and wireless sub-networks have been formulated separately using the task model proposed in Chapter 2 together with a virtual time-stamping approach. Delays caused by special components in the system, e.g., controllers, bridges, and gateways are also discussed, as they are part of the end-to-end delays as well. A delay model $\Pr_\tau$ has been proposed to summarize the characteristics of the end-to-end delays. At last, packet dropout rate has been studied and the calculation of network reliability for data communications in a bus cycle is given.

Chapter 6

Evaluation on Communication Quality of Service

With the mathematical analysis performed in Chapter 5, this chapter presents the analysis results on the communication QoS, including cycle time, end-to-end delay and communication reliability. The analysis is performed by implementing the network modelling described in Chapter 4 in MATLAB. The comparison between SFP and IFP is also addressed and discussed.

6.1 Cycle Time

6.1.1 Performance Comparison between SFP and IFP

The first evaluation objective is to study how layer-2 protocol can affect the cycle time between $\gamma = \text{HRT_SFP}$ and $\gamma = \text{HRT_IFP}$. The assessment result is illustrated in Figure 6.1. The x-, y-, and z-axis shows payload per device f_p , the number of devices N_{dev} and the ratio of bus cycle time between the SFP and the IFP, i.e., $\mathcal{D}_e^{\text{GTD}}(\gamma = \text{HRT_SFP})/\mathcal{D}_e^{\text{GTD}}(\gamma = \text{HRT_IFP})$, respectively.

The results demonstrate that the SFP outperforms the IFP in general, especially when f_p is small. This can be naturally explained by the fact that the SFP offers better coding efficiency and bandwidth utilization as compared to the IFP. The advantage of the SFP reduces when the total data volume is increased as f_p and/or N_{dev} grows. This advantage is most obvious when $f_p < 46$ Bytes. This is because padding bytes are required for the IFP if $f_p < 46$ Bytes. On the other hand, the SFP packs small payload together and

Figure 6.1: Efficiency factor of SFP relative to IFP as a function of f_p and N_{dev}

therefore, padding can be avoided. The only scenario where the IFP outperforms the SFP is when only one device is installed in the system. In this extreme case, the SFP introduces more overhead than the IFP due to summation-frame header and the CRC.

6.1.2 Cycle Time of Wireless Segment

The second objective of the evaluation is to study how the cycle time can be influenced by payload per device f_p and the number of wireless nodes N_{wln} on a wireless network. It is assumed in this assessment that the backoff time is a fixed constant, i.e., $T_{bof} = CW_{\min}T_{slot}/2$. This is to improve the readability of results by removing the randomness from T_{bof} which does not directly impact on the cycle time. It is also assumed that there are no wired devices deployed in the system, i.e., $N_{dev} = 0$. The performance of cycle time of the wireless network $\mathcal{D}_e^{\text{BE-WL}}$ is shown in Figure 6.2.

From the result, it can be observed that the cycle time is affected by the number of devices much more than the payload per wireless node. Recalling earlier Equation (5.6), this can be explained as the bandwidth utilization to transmit f_p is quite low. The communication overhead such as the switching delays on the wired segment, and the backoff time, the IFS and the ACK transmission time on the wireless segment is high. On the other hand, the cycle time is linearly increased as N_{wln} grows. This observation is completely in line with Equation (5.5) where $\mathcal{D}_e^{\text{BE-WL}}$ is in linear relationship with the size of wireless network $\mathbb{W}$. As a result, the wireless cycle time is mainly influenced by the scale of the system.

6.2 Network Delay

6.2.1 Performance Comparison between SFP and IFP

In this section, the performance of network delay τ_{net} on the S-C and C-A links is investigated with a system of $N_{dev} = 10$ and the results are presented in Figure 6.3 and Figure 6.4, respectively.

It can be noted from the Figure 6.3 that the SFP operates much more efficiently than the IFP on the S-C link. This behaviour can be interpreted as the *processing on-the-fly* mechanism adopted in the SFP does not require transmission on the S-C path. The data

Figure 6.2: Cycle time of wireless segment as a function of f_p and N_{wln}

Figure 6.3: Efficiency factor of SFP relative to IFP on S-C path as a function of f_p and n

Figure 6.4: Efficiency factor of SFP relative to IFP on C-A path as a function of f_p and n

can be simply stored in the DPRAM of the network controller to be appended to the incoming frame. The advantage of the SFP is most obvious on devices with small index. This is mainly caused by the slipstreaming effect as the controller sends the first frame to the last device ($n = N_{dev}$) and the last frame to the first device ($n = 1$) in the IFP. The ratio $\tau_{sc,net}(\gamma = \text{HRT_SFP})/\tau_{sc,net}(\gamma = \text{HRT_IFP})$ is constant when $f_p \leq 46$ Bytes because the padding bytes are required in case of IFP. When $f_p > 46$, increasing f_p has negative impact on the IFP due to larger frame size. On the other hand, it has no impact on the SFP because no transmission is required for S-C communication. Consequently, the advantage of the SFP enhances when f_p is increased with more than 46 Bytes.

The evaluation result is quite different on the C-A transmission path as shown in Figure 6.4 due to the event-driven nature of actuator devices. From the figure, it can be seen that the performance of the SFP is most advantageous when device index n is big. In order for the controller to communicate with the first actuators $n = 1$, there is only one frame transmission required in both IFP and SFP. The delay of the SFP is much higher than that of the IFP. This is caused by the fact that the summation-frame is longer than the individual frame in general, since the summation-frame packs the payload of all the devices. On the other hand, as the device number increases, the comparison result changes dramatically. The SFP outperforms the IFP the most when the device index is big while the payload is small.

6.2.2 Network Delay of Wireless Segment

This delay evaluation assumes a hybrid network-based system with $N_{dev} = 10$ and $N_{wln} = 10$ interconnected with a bridge installed at the end of the wired segment. The results of $\tau_{sc,net}(\gamma = \text{WLAN})$ and $\tau_{ca,net}(\gamma = \text{WLAN})$ are presented in Figures 6.5 and 6.6, respectively.

In general, both $\tau_{sc,net}(\gamma = \text{WLAN})$ and $\tau_{ca,net}(\gamma = \text{WLAN})$ increase linearly with the payload size. However, what impact more on the network delays is the node index. For the S-C link, the smaller the node number, the higher the network delays (see Figure 6.5). This is mainly caused by the IFP and the slipstreaming scheme adopted in the BE phase for wireless extension as the sensors with bigger index get to transmit first. Therefore, less access delay is expected for sensors installed further from the controller

Figure 6.5: Network delay of S-C link as a function of f_p and N_{wln}

Figure 6.6: Network delay of C-A link as a function of f_p and N_{wln}

in the network topology. This behaviour is completely turned around in the case of C-A link as displayed in Figure 6.6.

With Figures 6.2, 6.5 and 6.6, it can be concluded that the size of the wireless network is a much more influential factor on the cycle time and network delays of the wireless sub-network than the payload size.

6.3 Reliability

The comparison between the SFP and the IFP in terms of reliability factor $\mathcal{R}(\gamma = \text{HRT_SFP})/\mathcal{R}(\gamma = \text{HRT_IFP})$ is presented in Figure 6.7, assuming that unsolved BER $\text{Pr}_{ber} = 10^{-4}$.

The reliability of the summation-frame is poorer than each individual frame. However, it can be noticed from Figure 6.7 that the SFP provides higher overall system-level reliability than the IFP, i.e., lower packet dropout rate, due to smaller total frame size and consequently lower probability of transmission errors during each bus cycle. The relationship between efficiency factor and f_p is V-shaped, and the SFP is least advantageous when $f_p = 46$ Bytes. On the other hand, the efficiency factor increases along with parameter N_{dev} .

6.4 Conclusion

With regard to the GTD channel, the SFP is generally superior to the IFP in terms of the QoS for typical industrial control applications. It offers shorter communication cycle, smaller delays and higher transmission reliability. The SFP is especially efficient when the data volume is small. The communication requirement in industrial NCSs is quite different from other commercial domains such as office and enterprise applications. It is important to adapt standard Ethernet in terms of frame packing and MAC so that the communication network is most efficient to the industrial applications.

However, the SFP has its own limitation in the industrial automation applications. For example, the performance of the NCS over a SFP-based network can be highly dependent on the controller delay D_c . Additionally, the SFP is not as flexible as the IFP to be adopted for wireless extension on the data-link layer.

Figure 6.7: Efficiency factor of SFP relative to IFP as a function of f_p and N_{dev}

Chapter 7

Integrated Network-based Control — A Case Study

This chapter presents a case study of a network-based control system with motion control and AGV tracking applications. The case study is presented to verify the performance of the network design such as end-to-end delay and scalability.

Firstly, the background of the case study including system parameters and network topology is described. Wired devices and wireless nodes are integrated with one communication backbone designed by the scheme suggested in Chapter 4. Secondly, the limitation from existing simulators is discussed, and an evaluation strategy is proposed accordingly. Next, the applications considered in the case study are discussed individually. The cost functions of the control objective are defined to quantify the QoC. At last, the effectiveness of the network is evaluated by comparing the results of cost functions between local and networked control. The scalability of the network is demonstrated as well.

7.1 System Integration

The case study, as illustrated in Figure 1.1, considers an industrial control system with both motion control and AGV applications. The system schematic and parameters are presented in Figure 7.1 and Table 7.1, respectively. There are N_{dev} controlled devices and N_{wln} wireless nodes controlled by an industrial PC or a PLC. The line topology is assumed here as this should have little impact on the evaluation results [109]. A small

Table 7.1: Simulation parameters

Parameter	Value
N_{dev}	10 (small system) / 50 (large system)
N_{wln}	0 (no wireless) / 10 (small system) / 50 (large system)
f_p	10 Bytes (small system) / 100 Bytes (large system)
D_s	0.2 ms
D_c	1 ms
D_a	0.3 ms
Q	20
T_{sim}	60 s

Figure 7.1: System topology

system of $N_{dev} = N_{wln} = 10$ with $f_p = 10$ Bytes and a large system of $N_{dev} = N_{wln} = 50$ with $f_p = 100$ Bytes are considered to demonstrate the scalability (number of devices and data volumes) of the network. Additionally, assumptions are made as follows to simplify the performance evaluation without losing any generality:

- Each device/node has identical payload size, $\forall n \in \{\mathbb{D} \cup \mathbb{W}\}$
- Node latencies on all controlled devices are constant and identical
- No acyclic traffic appears in the network
- Controller has a powerful CPU and no expansion on the RSVD phase needed
- All wired devices request network services only during the GTD phase
- Cable propagation delay of 5 ns per meter is ignored

- Packet dropout is not considered, i.e., the reliability model derived in Chapter 5 is not applied

It should be also highlighted here that the coordination control of multiple AGVs normally requires direct data communication among the AGVs. Since the coordination algorithm is not the focus of the thesis, the communication activities for the needs of coordination is not considered. The communication activities necessary for the remote tracking are carried out between the bridge and each AGV through the BE phase of the bus cycle.

7.2 Evaluation Strategy

There are multiple tools available to evaluate the performance of the NCS. MATLAB toolbox named TrueTime is available for simulation of networked and embedded control systems [141]. It currently supports various network types, i.e., CAN, Ethernet, PROFINET IO (RT_CLASS_1 and RT_CLASS_3), etc. and wireless standard including IEEE 802.11b WLAN and IEEE 802.15.4 ZigBee. However, the toolbox cannot be used directly to simulate the behaviour of the hybrid network. Therefore, this toolbox is used in this chapter only to evaluate the outcome of the networked controller.

OMNeT++ is a C++-based discrete-event network simulator for evaluating communication networks [147]. Unlike TrueTime, OMNeT++ does not directly provide the network blocks with pre-defined network types, it offers a framework to model the network using modules and their interconnections. However, it is not possible to perform networking and control co-design by incorporating the control system simulation with OMNeT++.

Given the limitations of the existing available tools, the simulation methodology defined in this thesis is to perform network and control simulation separately as illustrated in Figure 7.2. The network simulation is carried out with a piece of MATLAB code based on the network model analyzed in Chapter 5. The output from the network simulation is a database which contains the QoS of the network, including cycle time, end-to-end delay and communication reliability. And this output can be used off-line as the input to the control simulation study performed by a Simulink model based on TrueTime toolbox.

Figure 7.2: Evaluation process on system performance

Firstly, network delay τ_{net} is highly dependent on system parameters for example N_{dev} and N_{wln} and also network parameters such as f_p . Consequently, the mathematical formulations described in Chapter 5 can be written in MATLAB code to obtain QoS results. Secondly, stack delay τ_{sta} is irrelevant to the system. It is strongly related to the dedicated hardware implementation of the stack, and hence can be very difficult to be modelled mathematically. The easiest way to obtain τ_{sta} is to perform experimental study. Such a study was made for EtherNet/IP, EtherCAT, and PROFINET IO by company Softing AG on an Altera Nios II soft processor [136]. The minimum, average and maximum values of τ_{sta} were provided in [136], however, without probability distributions. In this thesis, it is assumed that τ_{sta} is a random variable following PERT distribution according to the minimum, average and maximum values. Lastly, end-to-end delay can be obtained with τ_{net} and τ_{sta} and the results can be summarized using the delay model $\Pr_\tau$ described in Chapter 5 and stored as a database (e.g., MATLAB .mat file). The database is used as input parameters of a TrueTime model for networked control performance study. The controlled plant and the control loops are completely simulated by the model using TrueTime Version 2.0 Beta 8.

7.3 Communication Quality of Service

First, the results of network cycle time are presented in Table 7.2 for system configured with $N_{dev} = 10$ and $N_{wln} = 10$. It can be seen from the table that majority of the cycle time is contributed from the BE phase which is needed for wireless extension. Moreover, both SFP and IFP can achieve very fast bus cycle for the GTD phase.

Next, the end-to-end delay is studied with the scenario that $N_{dev} = 10$ and no wireless extension is necessary. This study is to demonstrate the real-time capability of the GTD phase. The performance of the end-to-end delays for the controlled plant installed in the centre ($n = N_{dev}/2$) is illustrated in Figures 7.3 to 7.5 and Table 7.3. The standard deviation (SD) of the delays is available in Table 7.3. It should be highlighted that the controller delay is excluded from the figures and the table, because it is very device-specific and is not the focus of this thesis. The first characteristic observable from the figure is that the delays are bounded. The next noticeable point is the distinct gap of the real-time performance between the HRT protocol and the NRT protocol.

Table 7.2: Cycle time performance ($N_{dev} = 10$, $N_{wln} = 10$)

Item	HRT_SFP	HRT_IFP
$\mathcal{D}_e^{GTD} (\mu s)$	94.88	106.20
$\mathcal{D}_e^{BE} (\mu s)$	3889.51	3889.51
$\mathcal{D}_e^{RSVD} (\mu s)$	122	122
$\mathcal{D}_e (\mu s)$	4106.39	4117.71

 Table 7.3: Comparison on end-to-end delays ($N_{dev} = 10$, $N_{wln} = 0$)

Results	HRT_SFP		HRT_IFP		NRT	
	S-C	C-A	S-C	C-A	S-C	C-A
Min. (μs)	97.85	180.75	161.23	150.81	2658.25	2551.38
Average (μs)	232.18	315.11	295.4	284.96	4168.36	4055.64
Max. (μs)	367.01	450.04	430.56	419.92	6090.85	5993.34
SD (μs)	67.33	67.31	67.42	67.23	852.81	864.26

The SFP offers the best QoS among all selected protocols $\gamma \in \{\text{HRT_SFP}, \text{HRT_IFP}, \text{NRT}\}$. Control data transfer on the SFP-based network is of delay of around 550 μs in the S-C and the C-A links together. This is caused by the fact that the SFP is most effective to applications with small data volume per device [15, 16, 74]. This leads to faster bus cycle and shorter network delay as compared to the IFP. It can be also noticed from Table 7.2 that the stack delay τ_{sta} and its jitter of the HRT candidates are much smaller than those of the NRT. This is because the payload travels in the HRT stack directly between the data-link layer and the application layer without passing through the TCP/IP stack. The worst real-time performance is delivered by the NRT candidate with more than 4 ms delays on the communication path.

7.4 Networked Control via Guaranteed Channel

7.4.1 Servo Control Application

The next process in the evaluation chain is to assess the control performance of the NCS with the delay model obtained earlier. In this study, four possible configurations are considered in order to facilitate performance comparison: direct connection, networked control over $\gamma = \text{HRT_IFP}$, $\gamma = \text{HRT_SFP}$, and $\gamma = \text{NRT}$. A DC servo process given in [141] is used as the reference system here and its second-order plant model is given by

$$G(s) = \frac{1000}{s(s+1)}. \quad (7.1)$$

7.3.a:

7.3.b:

Figure 7.3: Probability distribution of end-to-end delays for HRT_IFP: (a) τ_{sc} ; (b) τ_{ca} .

7.4.a:

7.4.b:

Figure 7.4: Probability distribution of end-to-end delays for HRT_HFP: (a) τ_{sc} ; (b) τ_{ca}

7.5.a:

7.5.b:

Figure 7.5: Probability distribution of end-to-end delays for NRT: (a) τ_{sc} ; (b) τ_{ca}

The servo motor is controlled by a networked PD controller according to

$$\begin{cases} P(k) = K(r(k) - y(k)), \\ D(k) = \delta D(k-1) + \rho(y(k-1) - y(k)), \\ u(k) = P(k) + D(k), \end{cases} \quad (7.2)$$

where $r(k)$, $u(k)$, $y(k)$ are the set-point, input, and output of the plant, respectively, during the k -th control cycle. Parameters of the PD controller, i.e., δ , ρ and K are set to the same values as in [141].

7.4.2 Cost Functions

Maximum overshoot is selected as the first cost function J_1 of the servo system. Additionally, integral absolute error (IAE) is considered as the second control performance criterion J_2 , which is given by

$$J_2 = \sum_{k=0}^{k_f} |e(k)|, \quad (7.3)$$

where $e(k)$ is the error between set-point and actual response at time instance k and $k_f = \frac{T_{sim}}{P_s}$.

7.4.3 Quality of Control

The step response of the servo process is shown in Figure 7.6, and the QoC results are presented in Table 7.4. The local control, as expected, delivers the best control performance against networked control, with maximum overshoot of 2.19% and settling time of 122 ms. On the other hand, it can be observed that both $\gamma = \text{HRT_SFP}$ and $\gamma = \text{HRT_IFP}$ are able to offer similar control results as compared to the local control. The IAE and overshoot achieved by networked control using $\gamma = \text{HRT_SFP}$ or $\gamma = \text{HRT_IFP}$ is almost equal to that of local control. However, the control performance is not satisfactory when $\gamma = \text{NRT}$ due to end-to-end delay. The servo process is unable to settle its response within 500 ms. The overshoot is about 6 times higher than that in local control. Scalability wise, the performance of $\gamma = \text{HRT_SFP}$ and $\gamma = \text{HRT_IFP}$ is excellent as the control performance does not vary much between the small and the big system set-up.

Table 7.4: QoC results of the motion control system

Connectivity	J_1	J_2
$N_{dev} = 10$		
Local	2.19%	17.85
$\gamma = \text{HRT_SFP}$	2.56%	17.93
$\gamma = \text{HRT_IFP}$	2.65%	17.94
$N_{dev} = 50$		
Local	2.19%	17.85
$\gamma = \text{HRT_SFP}$	2.84%	17.95
$\gamma = \text{HRT_IFP}$	3.02%	18.07

 Figure 7.6: System response of the servo system when $N_{dev} = 50$

Therefore, it can be concluded that the end-to-end delay has great impact on the control loop performance. Correspondingly, the HRT protocol can achieve more satisfactory control performance than the NRT standard due to faster and more deterministic data communication.

7.5 Remote Tracking over Hybrid Network

7.5.1 AGV Kinematic Model

The differential drive AGV considered in the thesis consists of two driving wheels and one caster wheel (see Figure 7.7). Translational velocity v and rotational velocity w can be respectively presented as

$$v = \frac{w_{R,c}\rho}{2} + \frac{w_{L,c}\rho}{2}, \quad (7.4)$$

$$w = \frac{w_{R,c}\rho}{W} - \frac{w_{L,c}\rho}{W}, \quad (7.5)$$

where ρ means the radius of the rear wheels on the AGV, and W refers to the distance between the rear wheels. $w_{R,c}$ and $w_{L,c}$ represent the current angular velocity of right and left wheels, respectively. Equation (7.4) and (7.5) can be re-written as the following kinematic model:

$$\begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} \frac{\rho}{2} & \frac{\rho}{2} \\ \frac{\rho}{W} & -\frac{\rho}{W} \end{bmatrix} \begin{bmatrix} w_{R,c} \\ w_{L,c} \end{bmatrix}. \quad (7.6)$$

Using translational and rotational velocities, the posture of the AGV after a control period P_c can be obtained as

$$x_c(k+1) = x_c(k) + vP_c \cos(\theta_c(k) + w\frac{P_c}{2}), \quad (7.7)$$

$$y_c(k+1) = y_c(k) + vP_c \sin(\theta_c(k) + w\frac{P_c}{2}), \quad (7.8)$$

$$\theta_c(k+1) = \theta_c(k) + wP_c. \quad (7.9)$$

Figure 7.7: AGV kinematics

Figure 7.8: Local path tracking control flow

7.5.2 Local Path Tracking Control

The local path tracking scheme is mainly reused from [96] and can be presented by Figure 7.8. The scheme aims to control the AGV so that it moves along a desired path. The tracking algorithm from reference [148] is reused due to its simplicity and feasibility for embedded implementation. The tracking algorithm derives a quadratic curve which connects the current position and the reference position. The translational and rotational velocities required for the AGV to move to the reference position are then calculated.

- (i) **Find the optimal point $s = s_0$ on the path from the current position**

Based on the initial ($k = 1$) or current ($k > 1$, $k \in \mathbb{N}$) position of the AGV

$[x_c(k) \ y_c(k)]^T$, the algorithm finds an optimal point $[x_p(s) \ y_p(s)]^T$ on the desired path so that distance between the current position and the path is minimized, i.e.,

$$\min_{s=s_0} \sqrt{(x_p(s) - x_c(k))^2 + (y_p(s) - y_c(k))^2}. \quad (7.10)$$

(ii) **Compute the projected travelling distance $s(k)$**

The projected travelling distance can be obtained via the algorithm defined in [148]. Without loss of generality, it is assumed that the desired path is composed of j segments and each segment is either a line or a curve with a constant curvature $\kappa(j)$. The initial projected distance s_t can be calculated as

$$s_t = s_0 + \Delta s(j), \quad (7.11)$$

where $\Delta s(j) = \frac{\Delta s_{\max}}{1 + \psi \kappa(j)}$. $\Delta s_{\max}$ is the maximum possible travelling distance between the AGV and the reference point, and ψ is a positive constant. The algorithm of $s(k)$ can be illustrated with Figure 7.9. If s_t is shorter than the distance to be travelled at the end of the j -th segment $s_{seg}(j)$, the projected travelling distance of the k -th control cycle $s(k) = s_t$. Otherwise, the projected distance requires the AGV to move to the next segment, i.e., $s_t = s_0 + \Delta s(j + 1)$. If the resulted s_t is longer than the distance at the end of segment j , i.e., $s_t > s_{seg}(j)$, $s(k)$ is given with value of s_t . Otherwise, $s(k) = s_{seg}(j)$. This is to ensure that the AGV does not move backtrack as s_t might be shorter than $s(k - 1)$, depending on the segment configuration. Therefore, it is better to set s_t to the end of segment j to guarantee that moving backtrack is not possible.

(iii) **Obtain the reference position $[x_r(k) \ y_r(k) \ \theta_r(k)]^T$**

With the projected travelling distance on the path computed, the reference position can be obtained from the path constructed. The path can be always presented as a function of the travelling distance s , i.e.,

Figure 7.9: Algorithm of computing $s(k)$

$$x_p(s) = f_x(s), y_p(s) = f_y(s), \text{ and } \theta_p(s) = f_\theta(s). \quad (7.12)$$

Then, by substituting s with $s(k)$, the reference point can be obtained from the desired path, i.e., $[x_r(k) \ y_r(k) \ \theta_r(k)] = [x_p(s(k)) \ y_p(s(k)) \ \theta_p(s(k))]$.

(iv) **Compute error vector**

The error vector $[e_x \ e_y \ e_\theta]^T$ between the reference position and the current position is obtained as

$$\begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} \cos \theta_c & \sin \theta_c & 0 \\ -\sin \theta_c & \cos \theta_c & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x_c \\ y_r - y_c \\ \theta_r - \theta_c \end{bmatrix}. \quad (7.13)$$

The orthogonal rotation matrix is to transform the AGV coordinate from the global to the local coordinate.

(v) **Obtain the quadratic curve and reference velocities**

The error vector is used to compute the quadratic curve in the local coordinate which links between current point to the reference point

$$y_M = Ax_M^2, \quad (7.14)$$

where $A = \text{sign}(e_x) \frac{e_y}{e_x^2}$. Note that the AGV runs forward if e_x is positive and moves backward if e_x is negative. The AGV moves along the calculated quadratic curve with the following desired translational velocity v_r and rotational velocity w_r

$$v_r = \text{sign}(e_x) \sqrt{\dot{x}_M^2 (1 + 4A^2(k) \dot{x}_M^2)}, \quad (7.15)$$

$$w_r = \frac{2A(k) \dot{x}_M^3}{v_r^2(k)}. \quad (7.16)$$

Accordingly, if x_M at the time of $kP_c \leq t < (k+1)P_c$ is given by

$$x_M = K(k)(t - kP_c), \text{ where gain factor } K(k) = \text{sign}(e_x) \frac{\phi}{1 + |A(k)|}, \quad (7.17)$$

where ϕ is a factor of the speed of the translational and rotational movements. The AGV runs faster when ϕ is large, and vice versa.

v_r and w_r can be approximated to the following when P_c is sufficiently small.

$$\hat{v}_r \approx K(k), \quad (7.18)$$

$$\hat{w}_r \approx 2A(k)K(k). \quad (7.19)$$

The desired angular speeds of right and left wheels required by the AGV in order to reach the reference position can be calculated from $\hat{v}_r$ and $\hat{w}_r$:

$$w_{R,r}(k) = \frac{\hat{v}_r(k)}{\rho} + \frac{W\hat{w}_r(k)}{2\rho} \quad (7.20)$$

$$w_{L,r}(k) = \frac{\hat{v}_r(k)}{\rho} - \frac{W\hat{w}_r(k)}{2\rho} \quad (7.21)$$

The reference speeds of both wheels are then known to the PI controllers of the wheel motors.

(vi) **Set $k = k + 1$ and repeat all steps until destination is reached**

The current position of the AGV for the $(k + 1)$ -th control loop can be calculated with Equation (7.7), (7.8) and (7.9). The destination is reached when $|s(k) - s_{seg}(\max(j))| \leq s_{tol}$ is satisfied.

Figure 7.10: Remote path tracking control flow

7.5.3 Remote Path Tracking Control

The remote path tracking control can be illustrated with Figure 7.10. The difference from the local tracking scenario is the end-to-end delays $\tau(k)$ imposed on the components and the network. Firstly, due to S-C link delay τ_{sc} , the controller does not know the exact current location of the AGV, but $[x_c(t - \tau_{sc}(k)) \ y_c(t - \tau_{sc}(k)) \ \theta_c(t - \tau_{sc}(k))]$. On the other hand, the desired speeds of both wheels $w_{R,r}$ and $w_{L,r}$ computed at time t will only be applied at $t + \tau_{ca}$. This provokes inconsistent operation of the local tracking algorithms [95]. Consequently, it is encouraged to adjust the controller output with respect to the network condition characterized by Pr_τ as discussed in Chapter 5. In this thesis, the following scheme is defined to use the estimated AGV posture instead of the actual posture in steps of (7.10) and (7.13) of the local tracking algorithm, i.e.,

$$\hat{x}_c(k) = x_c(k) + v\tau(k) \cos(\theta_c(k) + w\frac{\tau(k)}{2}), \quad (7.22)$$

$$\hat{y}_c(k) = y_c(k) + v\tau(k) \sin(\theta_c(k) + w\frac{\tau(k)}{2}), \quad (7.23)$$

$$\hat{\theta}_c(k) = \theta_c(k) + w\tau(k). \quad (7.24)$$

7.5.4 Application Parameters

Without loss of generality, the path is constructed by a straight line and two arcs. The desired moving path $[x_p(s) \ y_p(s)]^T$ of the AGV can be described as the followings.

- (i) **Segment 1: Straight line of 10 meters**

Table 7.5: Simulation parameters for the AGV application

Parameter	Value
r	0.08 meter
W	1.2 meter
$s_{\max}$	0.5 meter
s_{tol}	0.1 meter
ψ	5
ϕ	1

$$x_p(s) = 0, y_p(s) = s \quad (7.25)$$

where s denotes the distance travelled by the AGV on the path. For Segment 1, $0 \leq s \leq s_{seg}(1)$ and $s_{seg}(1) = 10$.

(ii) **Segment 2: Curve with a radius of 10 meters**

$$x_p(s) = 10 - 10 \cos \frac{1}{10}(s - s_{seg}(1)), y_p(s) = 10 + \frac{1}{10} \sin 10(s - s_{seg}(1)) \quad (7.26)$$

where $s_{seg}(1) \leq s \leq s_{seg}(2)$ and $s_{seg}(2) = 10 + 10\pi$.

(iii) **Segment 3: Curve with a radius of 2 meters**

$$x_p(s) = 22 - 2 \cos \frac{1}{2}(s - s_{seg}(2)), y_p(s) = 10 - 10 \sin \frac{1}{2}(s - s_{seg}(2)) \quad (7.27)$$

where $s_{seg}(2) \leq s \leq s_{seg}(3)$ and $s_{seg}(3) = 10 + 12\pi$.

Other system parameters are specified according to Table 7.5. The explanations of the mathematical symbols can be found in Appendix.

7.5.5 Cost Functions

The first cost function J_3 is defined to measure the deviation of the AGV movement from the desired path. J_3 can be expressed as

$$J_3 = \sum_{k=1}^{\max(k)} \sqrt{(x_p(s(k)) - x_c(k))^2 + (y_p(s(k)) - y_c(k))^2}, \quad (7.28)$$

where $(x_c(k)$ and $y_c(k))$ denote the current position of the AGV in x- and y-axis, respectively.

The second cost function J_4 simply represents the total travelling time spent by the AGV to move from the source to the destination in the system. The calculation of J_4 can be given by

$$J_4 = \max(k)P_c \quad (7.29)$$

7.5.6 Quality of Control

The actual path $[x_c(k) \ y_c(k)]^T$ followed by the AGV and the distance between the desired and the actual path are the evaluation objects. The control result is presented in Figures 7.11 and 7.12.

In Figure 7.11, it can be observed that the AGV has no problem in following the pre-defined path $[x_p(s) \ y_p(s)]^T$ with both local and remote tracking schemes. However, the accuracy with the remote tracking is already impacted by the end-to-end delay induced on the hybrid network. This can be noticed in Figure 7.12 and Table 7.6. In the case of local tracking, the AGV moves within the tolerance of 0.1 m for majority of the time. On the other hand, the end-to-end delay begins to reduce tracking stability in the case of a small network where $N_{dev} = N_{wln} = 10$. The performance impact is most noticeable when a large network ($N_{dev} = N_{wln} = 50$) is considered. With S-C and C-A network delays of average 174 ms and 165 ms, it is no longer possible to maintain the distance between the AGV and the desired path of less than 0.1 m for most of the time. The performance indicator J_3 under remote tracking scheme in a large system is almost 4 times higher as compared with the case of local tracking. This is in line with the observation in Figure 7.12. Moreover, the AGV takes 0.7 s more travelling time to reach the destination point within acceptable distance tolerance s_{tol} when it is controlled over the network than the dedicated connection locally. Therefore, both tracking accuracy and travelling time outcomes are degraded by the communication delays incurred by the network.

Figure 7.11: Actual path of the AGV

Table 7.6: Comparison on QoC between local and remote tracking

Control scheme	J_3 (m)	J_4 (s)
Local tracking	4.31	11.0
Remote tracking ($N_{dev} = 10$, $N_{wln} = 10$)	4.93	11.0
Remote tracking ($N_{dev} = 50$, $N_{wln} = 50$)	15.80	11.7

Figure 7.12: Closest distance between desired and actual path

7.6 Conclusion

This case study presented in this chapter verified the design and analysis carried out in Chapters 4 and 5. An industrial NCS with motion control application and AGV tracking application has been described. Cost functions of both applications have been defined to illustrate the performance of the control. The simulation results showed that the network delays have degraded the system performance to some extent. However, the proposed networking solution is capable of satisfying communication and control requirements in both small and large systems studied. This offers flexibility and scalability to the system to be prepared for Industry 4.0.

Comparing to the existing standard, the network suggested in the thesis is able to offer guaranteed services which are not available in EtherNet/IP, Modbus TCP/IP and Ethernet POWERLINK. The network can be interconnected to a wireless sub-network which cannot be achieved by EtherCAT and PROFINET IRT. The polling method proposed in [20, 21] is reused in the BE phase for wireless extension. This helps the network access to be more controlled and deterministic. Thus, it should meet all common communication requirements defined for networked control in next generation of industrial automation.

Chapter 8

Conclusion and Recommendations

8.1 Conclusion

This thesis designed and analyzed a real-time communication network which offers both guaranteed and best-effort QoS and can be extended with a wireless segment. The thesis also explored its networked control applications in the industrial NCS. The thesis can be summarized with the following points:

- (i) Chapter 3 presented the structure of the NCS in industrial automation. A task model was introduced to study the behaviours of system processes. With a GTD channel, a BE channel and a RSVD channel, a generic networking solution was proposed in Chapter 4 to meet different control requirements such as environmental control, transport, networked motion control and mobile robotics.
- (ii) Wireless extension possibilities using the GTD and the BE channels were investigated. Interconnections of the sub-networks at the data-link or the application layer during the BE phase were described.
- (iii) The communication QoS indicators (bus cycle, end-to-end delay and reliability) of the proposed IIoT network were discussed in Chapter 5. The bus cycle times in various phases were mathematically formulated. The delays on S-C and C-A links were analyzed using Gantt charts and formulated for both wired and wireless segments. A delay model was introduced to contribute to a communication QoS database which can be used as the input to controller design and verification.

- (iv) The effectiveness of the proposed networking solution was verified via the simulation study of a control system consisting of both hard real-time servo control and mobile AGV remote tracking. Cost functions were defined as QoC indicators for both local and remote applications. An evaluation methodology was proposed and implemented. The control performance was evaluated and the relationship between the performance and the communication QoS was discussed to demonstrate the effectiveness of the network.

8.2 Recommendations for Further Research

Future potential studies are listed as follows:

- (i) **Performance optimization for the BE channel**

In the analysis in Chapter 4, a polling mechanism is used to include wireless communication into the bus cycle. On one hand, the integration of the wireless sub-network is straightforward with the polling scheme. On the other hand, the communication is performed on a half-duplex basis under the polling scheme. It is of great significance to optimize the MAC scheme during the BE phase to improve the overall bus performance (cycle time and delays) while retaining the possibility of wireless extension.

- (ii) **Investigation of IEEE time-sensitive networking (TSN) standard**

The thesis suggested a multi-phase communication model for NCSs in industrial automation. The motivation behind is to generalize all proprietary technologies into a standard one. There is also such a trend in the industry, and as a result, the TSN is being developed and standardized by IEEE working group. It can be worth spending effort to compare the suggested model with the TSN as defined in IEEE [149], especially on the topics of MAC scheme and wireless extension.

- (iii) **Improvement on the design flow of industrial NCS**

It would be of significance to come up with a general NCS design flow from the engineering point of view. For example, control objectives can be some QoC targets, such as settling time, overshoot, tracking error, etc. However, to fulfil these control targets, certain services on the network have to be guaranteed (for example short $\tau_{sc} + \tau_{ca}$). Based on these requirements, candidates of networking protocols can be properly selected. Next, tasks like detailed design (e.g., scheduling, number of switches, size of payloads), implementation and testing on the network have to be carried out to deliver QoS as per required. With that, it comes back to the controller design, based on the QoS delivered from the network. Eventually, the control system has to be implemented, tested and verified. The whole NCS design process is similar to the *V-model* which can be commonly found in automotive industry. A concept of network and control co-design was recently proposed by researchers [150–152]. The right design flow will offer effectiveness and correctness in the NCS design.

(iv) Network simulation using commercial tools

The thesis provided the mathematical modelling on the designed communication network. Due to the limitation of resource, the QoS results are not cross checked against simulation studies or experiments. It would be necessary to verify the solution via a simulation using commercial tool such as OMNeT++, and compare the simulative results with the analysis results. The framework established using the commercial tool would also allow industrial practitioners to design their project in an efficient way.

List of Publication

Journal Papers

- X. Wu and L. Xie, “On load balancing strategies for baggage screening at airports”, *Journal of Air Transport Management*, vol. 62, pp. 82-89, 2017.
Note: This article is not incorporated in the thesis.
- X. Wu and L. Xie, “Towards an IIoT-based architecture for baggage handling systems”, *Journal of Communications*, vol. 12, no. 8, pp. 475-481, 2017.
- X. Wu and L. Xie, “Performance evaluation of industrial Ethernet protocols for networked control application”, *Control Engineering Practice*, vol. 84, pp. 208-217, 2019.

Conference Papers

- X. Wu, L. Xie and F. Lim, “EtherCAT-enabled next generation baggage handling systems”, in *Proceedings of the 18th IEEE Conference on Emerging Technologies & Factory Automation (ETFA)*, Cagliari, Italy, pp. 1-6, 2013.
- X. Wu, L. Xie and F. Lim, “Network delay analysis of EtherCAT and PROFINET IRT protocols”, in *Proceedings of the 40th Annual Conference of the IEEE Industrial Electronics Society (IECON)*, Dallas, TX, USA, pp. 2597-2603, 2014.
- X. Wu and L. Xie, “End-to-end delay evaluation of industrial automation systems based on EtherCAT”, in *Proceedings of the 42nd IEEE Conference on Local Computer Networks (LCN)*, Singapore, pp. 70-77, 2017.

- X. Wu and L. Xie, “On the wireless extension of EtherCAT networks ”, in *Proceedings of the 42nd IEEE Conference on Local Computer Networks (LCN)*, Singapore, pp. 235-238, 2017.

References

- [1] R. Drath and A. Horch, “Industrie 4.0: Hit or hype?,” *IEEE Industrial Electronics Magazine*, vol. 8, pp. 56–58, June 2014.
- [2] L. Zhang, Y. Luo, F. Tao, B. H. Li, L. Ren, X. Zhang, H. Guo, Y. Cheng, A. Hu, and Y. Liu, “Cloud manufacturing: A new manufacturing paradigm,” *Enterprise Information Systems*, vol. 8, no. 2, pp. 167–187, 2014.
- [3] Texas Instruments, “Industry 4.0 solutions,” 2018 (accessed on 2018-10-14). <http://www.ti.com/applications/industrial/industry-4-0.html>.
- [4] R. A. Gupta and M.-Y. Chow, “Overview of networked control systems,” in *Networked Control Systems: Theory and Applications*, pp. 1–23, Springer London, 2008.
- [5] J. Garcia, F. R. Palomo, A. Luque, C. Aracil, J. M. Quero, D. Carrion, F. Gamiz, P. Revilla, J. Perez-Tinao, M. Moreno, P. Robles, and L. G. Franquelo, “Reconfigurable distributed network control system for industrial plant automation,” *IEEE Transactions on Industrial Electronics*, vol. 51, pp. 1168–1180, Dec 2004.
- [6] X. Wu and L. Xie, “Towards an IIoT-based architecture for baggage handling systems,” *Journal of Communications*, vol. 12, no. 8, pp. 475–481, 2017.
- [7] R. M. Murray, K. J. Astrom, S. P. Boyd, R. W. Brockett, and G. Stein, “Future directions in control in an information-rich world,” *IEEE Control Systems Magazine*, vol. 23, pp. 20–33, April 2003.
- [8] Y. Zhang, Z. Guo, J. Lv, and Y. Liu, “A framework for smart production-logistics systems based on CPS and industrial IoT,” *IEEE Transactions on Industrial Informatics*, vol. 14, pp. 4019–4032, Sept 2018.

REFERENCES

- [9] F.-L. Lian, W. Moyne, and D. Tilbury, “Network design consideration for distributed control systems,” *Control Systems Technology, IEEE Transactions on*, vol. 10, no. 2, pp. 297–307, 2002.
- [10] J.-D. Decotignie, “Ethernet-based real-time and industrial communications,” *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1102–1117, 2005.
- [11] M. Felser, “Real-time Ethernet — Industry prospective,” *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1118–1129, 2005.
- [12] P. Danielis, J. Skodzik, V. Altmann, E. B. Schweissguth, F. Golasowski, D. Timmermann, and J. Schacht, “Survey on real-time communication via Ethernet in industrial automation environments,” in *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, pp. 1–8, Sept 2014.
- [13] F.-L. Lian, J. R. Moyne, and D. M. Tilbury, “Performance evaluation of control networks: Ethernet, ControlNet, and DeviceNet,” *IEEE Control Systems Magazine*, vol. 21, pp. 66–83, Feb 2001.
- [14] M. Sung, I. Kim, and T. Kim, “Toward a holistic delay analysis of EtherCAT synchronized control processes,” *International Journal of Computers, Communications & Control*, vol. 8, no. 4, pp. 608–621, 2013.
- [15] J. Jasperneite, M. Schumacher, and K. Weber, “Limits of increasing the performance of industrial Ethernet protocols,” in *Emerging Technologies and Factory Automation (ETFA), 2007 IEEE International Conference on*, pp. 17–24, 2007.
- [16] G. Prytz, “A performance analysis of EtherCAT and PROFINET IRT,” in *Emerging Technologies and Factory Automation (ETFA), 2008 IEEE International Conference on*, pp. 408–415, 2008.
- [17] S. Höme, S. Palis, and C. Diedrich, “Design of communication systems for networked control system running on PROFINET,” in *Factory Communication Systems (WFCS), 2014 10th IEEE Workshop on*, pp. 1–8, May 2014.

REFERENCES

- [18] R. Schlesinger, A. Springer, and T. Sauter, “New approach for improvements and comparison of high performance real-time Ethernet networks,” in *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, pp. 1–4, Sep. 2014.
- [19] A. Willig, “Polling-based MAC protocols for improving real-time performance in a wireless PROFIBUS,” *IEEE Transactions on Industrial Electronics*, vol. 50, pp. 806–817, Aug 2003.
- [20] L. Seno and S. Vitturi, “Wireless extension of Ethernet POWERLINK networks based on the IEEE 802.11 wireless LAN,” in *2008 IEEE International Workshop on Factory Communication Systems*, pp. 55–63, May 2008.
- [21] L. Seno, S. Vitturi, and C. Zunino, “Analysis of Ethernet POWERLINK wireless extensions based on the IEEE 802.11 WLAN,” *IEEE Transactions on Industrial Informatics*, vol. 5, pp. 86–98, May 2009.
- [22] H. Wang, W. Hou, and M. Fei, “Polling-based protocol converter for the integration of hybrid wired/wireless industrial control networks,” in *Proceedings of the 29th Chinese Control Conference*, pp. 5772–5777, July 2010.
- [23] R. A. Gupta and M. Chow, “Networked control system: Overview and research trends,” *IEEE Transactions on Industrial Electronics*, vol. 57, pp. 2527–2535, July 2010.
- [24] Industrial Ethernet Book, “The world market for industrial Ethernet components,” 2012 (accessed on 2018-12-08). <https://iebmedia.com/index.php?id=8595&parentid=74&themeid=255&showdetail=true&bb=true>.
- [25] HMS Industrial Networks, “Network trends + Technologies vision,” 2017 (accessed on 2018-08-02). http://ethercat.org/forms/italy2017/files/14_HMS_Trends_Technologies_Vision.pdf.
- [26] HMS Industrial Networks, “Industrial Ethernet is now bigger than fieldbuses,” 2018 (accessed on 2018-11-09). <https://www.hms-networks.com/press/2018/02/27/industrial-ethernet-is-now-bigger-than-fieldbuses>.

REFERENCES

- [27] HMS Industrial Networks, “Industrial network market shares 2019 according to HMS,” 2019 (accessed on 2019-05-10). <https://www.hms-networks.com/press/2019/05/07/industrial-network-market-shares-2019-according-to-hms>.
- [28] B. Galloway and G. P. Hancke, “Introduction to industrial control networks,” *IEEE Communications Surveys Tutorials*, vol. 15, pp. 860–880, Second 2013.
- [29] Q. Liu and Y. Li, “Modbus/TCP based network control system for water process in the firepower plant,” in *2006 6th World Congress on Intelligent Control and Automation*, vol. 1, pp. 432–435, June 2006.
- [30] X. Qian, S. He, D. Guo, and Y. Jing, “On time-critical data transmission of EtherNet/IP,” in *2006 6th World Congress on Intelligent Control and Automation*, vol. 1, pp. 4623–4625, June 2006.
- [31] L. Seno and S. Vitturi, “A simulation study of Ethernet POWERLINK networks,” in *2007 IEEE Conference on Emerging Technologies and Factory Automation (EFTA 2007)*, pp. 740–743, Sept 2007.
- [32] R. Pigan and M. Metter, *Automating with PROFINET: Industrial communication based on industrial Ethernet*. Wiley-VCH, 2008.
- [33] X. Wu, L. Xie, and F. Lim, “Network delay analysis of EtherCAT and PROFINET IRT protocols,” in *Industrial Electronics Society, IECON 2014 - 40th Annual Conference of the IEEE*, pp. 2597–2603, Oct 2014.
- [34] N. Kalappa, K. Acton, M. Antolovic, S. Mantri, J. Parrott, J. Luntz, J. Moyne, and D. Tilbury, “Experimental determination of real time peer to peer communication characteristics of EtherNet/IP,” in *2006 IEEE Conference on Emerging Technologies and Factory Automation*, pp. 1061–1064, Sept 2006.
- [35] J. Barrios-Avila, A. Rosado-Muoz, T. Iakymchuk, and M. Garca-Chulbi, “POWERLINK and Ethernet/IP comparison as robust industrial Ethernet protocols,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 363–368, 2017. 20th IFAC World Congress.

- [36] L. Drkop, J. Jasperneite, and A. Fay, “An analysis of real-time Ethernets with regard to their automatic configuration,” in *2015 IEEE World Conference on Factory Communication Systems (WFCS)*, pp. 1–8, May 2015.
- [37] P. Ferrari, A. Flammini, F. Venturini, and A. Augelli, “Large PROFINET IO RT networks for factory automation: A case study,” in *ETFA2011*, pp. 1–4, Sep. 2011.
- [38] S. Vitturi, L. Peretti, L. Seno, M. Zigliotto, and C. Zunino, “Real-time Ethernet networks for motion control,” *Computer Standards & Interfaces*, vol. 33, no. 5, pp. 465–476, 2011.
- [39] A. Przyby, “Hard real-time communication solution for mechatronic systems,” *Robotics and Computer-Integrated Manufacturing*, vol. 49, pp. 309–316, 2018.
- [40] M. Schumacher, J. Jasperneite, and K. Weber, “A new approach for increasing the performance of the industrial Ethernet system PROFINET,” in *Factory Communication Systems, 2008 IEEE International Workshop on*, pp. 159–167, 2008.
- [41] D. Gunzinger, C. Kuenzle, A. Schwarz, H. Doran, and K. Weber, “Optimising PROFINET IRT for fast cycle times: A proof of concept,” in *Factory Communication Systems (WFCS), 2010 8th IEEE International Workshop on*, pp. 35–42, May 2010.
- [42] L. Wisniewski, J. Jasperneite, and C. Diedrich, “Effective and fast approach to schedule communication in PROFINET IRT networks,” in *Industrial Electronics (ISIE), 2013 IEEE International Symposium on*, pp. 1–6, 2013.
- [43] P. Ferrari, A. Flammini, D. Marioli, A. Taroni, and F. Venturini, “Evaluation of timing characteristics of a prototype system based on PROFINET IO RT Class 3,” in *Emerging Technologies and Factory Automation (ETFA), 2007 IEEE International Conference on*, pp. 1254–1261, 2007.
- [44] C. Felser, M. Felser, and H. Kaghazchi, “Improved architecture for PROFINET IRT devices,” in *Emerging Technologies Factory Automation (ETFA), 2012 IEEE International Conference on*, pp. 1–8, 2012.

REFERENCES

- [45] R. Schlesinger, A. Springer, and T. Sauter, “Improving PROFINET IRT frame packing using Ethernet control characters,” in *2015 IEEE World Conference on Factory Communication Systems (WFCS)*, pp. 1–4, May 2015.
- [46] R. Schlesinger, A. Springer, and T. Sauter, “Automatic packing mechanism for simplification of the scheduling in PROFINET IRT,” *IEEE Transactions on Industrial Informatics*, vol. 12, pp. 1822–1831, Oct 2016.
- [47] G. Cena, A. Valenzano, and C. Zunino, “An arbitration-based access scheme for EtherCAT networks,” in *2008 IEEE International Conference on Emerging Technologies and Factory Automation*, pp. 416–423, Sept 2008.
- [48] G. Cena, I. Bertolotti, A. Valenzano, and C. Zunino, “A high-performance CAN-like arbitration scheme for EtherCAT,” in *Emerging Technologies Factory Automation (ETFA), 2009 IEEE International Conference on*, pp. 1–8, 2009.
- [49] G. Patti, L. L. Bello, G. Alderisi, and O. Mirabella, “An EDF-based swapping approach to enhance support for asynchronous real-time traffic over EtherCAT networks,” in *2013 IEEE 18th Conference on Emerging Technologies Factory Automation (ETFA)*, pp. 1–8, Sept 2013.
- [50] L. Bello, G. Patti, G. Alderisi, V. Patti, and O. Mirabella, “A flexible mechanism for efficient transmission of aperiodic real-time messages over EtherCAT networks,” in *Factory Communication Systems (WFCS), 2014 10th IEEE Workshop on*, pp. 1–8, May 2014.
- [51] L. Lo Bello, E. Bini, and G. Patti, “Priority-driven swapping-based scheduling of aperiodic real-time messages over EtherCAT networks,” *Industrial Informatics, IEEE Transactions on*, vol. 11, pp. 741–751, June 2015.
- [52] G. Santandrea, “A PROFINET IO application implemented on wireless LAN,” in *2006 IEEE International Workshop on Factory Communication Systems*, June 2006.

- [53] R. Bayindir and Y. Cetinceviz, “A water pumping control system with a programmable logic controller (PLC) and industrial wireless modules for industrial plants — An experimental setup,” *ISA Transactions*, vol. 50, no. 2, pp. 321–328, 2011.
- [54] Y. Cetinceviz and R. Bayindir, “Design and implementation of an Internet based effective controlling and monitoring system with wireless fieldbus communications technologies for process automation — An experimental study,” *ISA Transactions*, vol. 51, no. 3, pp. 461–470, 2012.
- [55] S. Rinaldi, P. Ferrari, N. M. Ali, and F. Gringoli, “IEC 61850 for micro grid automation over heterogeneous network: Requirements and real case deployment,” in *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)*, pp. 923–930, July 2015.
- [56] M. R. Caldieri, J. A. Bigheti, and E. P. Godoy, “Implementation and evaluation of wireless networked control systems using Modbus,” *IEEE Latin America Transactions*, vol. 15, pp. 206–212, Feb 2017.
- [57] G. Cena, A. Valenzano, and S. Vitturi, “Hybrid wired/wireless networks for real-time communications,” *IEEE Industrial Electronics Magazine*, vol. 2, pp. 8–20, March 2008.
- [58] T. Sauter, S. Soucek, W. Kastner, and D. Dietrich, “The evolution of factory and building automation,” *IEEE Industrial Electronics Magazine*, vol. 5, pp. 35–48, Sept 2011.
- [59] J. von Hoyningen-Huene, A. Mueller, S. Dietrich, and G. May, “Comparison of wireless gateway concepts for industrial real-time communication,” in *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1–4, Sept 2016.
- [60] J. Kjellsson, A. E. Vallestad, R. Steigmann, and D. Dzung, “Integration of a wireless I/O interface for PROFIBUS and PROFINET for factory automation,” *IEEE Transactions on Industrial Electronics*, vol. 56, pp. 4279–4287, Oct 2009.

- [61] J. kerberg, M. Gidlund, T. Lennvall, J. Neander, and M. Bjrkman, “Integration of WirelessHART networks in distributed control systems using PROFINET IO,” in *2010 8th IEEE International Conference on Industrial Informatics*, pp. 154–159, July 2010.
- [62] W. Ikram, N. Jansson, T. Harvei, N. Aakvaag, I. Halvorsen, S. Petersen, S. Carlsen, and N. F. Thornhill, “Wireless communication in process control loop: Requirements analysis, industry practices and experimental evaluation,” in *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, pp. 1–8, Sept 2014.
- [63] Y. Wei and W. Han, “The wireless gateway design of the industrial Ethernet EtherCAT network,” *Modular Machine Tool & Automatic Manufacturing Technique*, no. 8, pp. 26–29, 2013.
- [64] M. Yoo, S. Lee, and W. Lee, “Method and gateway for extending EtherCAT network,” April 2016. US Patent 9,313,050.
- [65] X. Wu and L. Xie, “On the wireless extension of EtherCAT networks,” in *2017 IEEE 42nd Conference on Local Computer Networks (LCN)*, pp. 235–238, Oct 2017.
- [66] K. Zhang, L. Hu, P. Zuo, X. Wu, and K. Miao, “Wireless extension mechanism and logic design for FPGA-based Ethernet POWERLINK node,” in *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*, pp. 1–7, June 2016.
- [67] Y.-C. Li, S. H. Hong, X. Huang, G. Chen, and X. Liang, “Implementation of a POWERLINK-WirelessHART gateway for industrial automation,” in *2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pp. 1–6, June 2016.
- [68] L. Seno, S. Vitturi, and C. Zunino, “Real time Ethernet networks evaluation using performance indicators,” in *Emerging Technologies Factory Automation (ETFA), 2009 IEEE International Conference on*, pp. 1–8, 2009.

- [69] IEC, “IEC 61784-2-10: Industrial communication networks – Profiles – Part 2: Additional Fieldbus profiles for real-time networks based on ISO/IEC 8802-3, year = 2014,,”
- [70] P. Ferrari, A. Flammini, D. Marioli, A. Taroni, and F. Venturini, “Experimental analysis to estimate jitter in PROFINET IO Class 1 networks,” in *2006 IEEE Conference on Emerging Technologies and Factory Automation*, pp. 429–432, Sept 2006.
- [71] C. L. Toh and L. Norum, “A performance analysis of three potential control network for monitoring and control in power electronics converter,” in *Industrial Technology (ICIT), 2012 IEEE International Conference on*, pp. 224–229, 2012.
- [72] E. R. Jrmy Robert, Jean-Philippe Georges and T. Divoux, “Minimum cycle time analysis of Ethernet-based real-time protocols,” *International Journal of Computers, Communications and Control*, vol. 7, no. 4, pp. 743–757, 2012.
- [73] M. Knezic, B. Dokic, and Z. Ivanovic, “Performance analysis of the Ethernet POWERLINK poll-response chaining mechanism,” in *2015 IEEE World Conference on Factory Communication Systems (WFCS)*, pp. 1–4, May 2015.
- [74] K. Wang, X. Li, and X. Wang, “Analysis of real-time Ethernet communication technologies of summation frame and individual frames,” in *2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC)*, pp. 23–26, Oct 2017.
- [75] T. Skeie, S. Johannessen, and O. Holmeide, “Timeliness of real-time IP communication in switched industrial Ethernet networks,” *IEEE Transactions on Industrial Informatics*, vol. 2, pp. 25–39, Feb 2006.
- [76] J. Parrott, J. Moyne, and D. Tilbury, “Experimental determination of network quality of service in Ethernet: UDP, OPC, and VPN,” in *2006 American Control Conference*, pp. 4864–4869, 2006.

REFERENCES

- [77] K.-C. Lee, S. Lee, and M. H. Lee, “Worst case communication delay of real-time industrial switched Ethernet with multiple levels,” *Industrial Electronics, IEEE Transactions on*, vol. 53, no. 5, pp. 1669–1676, 2006.
- [78] P. Lindgren, J. Eriksson, M. Lindner, A. Lindner, D. Pereira, and L. M. Pinho, “Response time for IEC 61499 over Ethernet,” in *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)*, pp. 1206–1212, July 2015.
- [79] S. Canovas and C. Cugnasca, “Implementation of a control loop experiment in a network-based control system with LonWorks technology and IP networks,” *Industrial Electronics, IEEE Transactions on*, vol. 57, pp. 3857–3867, Nov 2010.
- [80] N. Bibinagar and K. Won-jong, “Switched Ethernet-based real-time networked control system with multiple client-server architecture,” *Mechatronics, IEEE/ASME Transactions on*, vol. 18, no. 1, pp. 104–112, 2013.
- [81] N. Subha and G. Liu, “Design and practical implementation of external consensus protocol for networked multi-agent systems with communication delays,” *Control Systems Technology, IEEE Transactions on*, vol. 23, pp. 619–631, March 2015.
- [82] A. Jestratjew and A. Kwiecien, “Performance of HTTP protocol in networked control systems,” *Industrial Informatics, IEEE Transactions on*, vol. 9, no. 1, pp. 271–276, 2013.
- [83] W.-J. Kim, K. Ji, and A. Ambike, “Networked real-time control strategies dealing with stochastic time delays and packet losses,” in *2005 American Control Conference*, vol. 1, pp. 621–626, 2005.
- [84] P. L. Tang and C. W. de Silva, “Compensation for transmission delays in an Ethernet-based control network using variable-horizon predictive control,” *IEEE Transactions on Control Systems Technology*, vol. 14, pp. 707–718, July 2006.
- [85] . Cuenca, J. Salt, A. Sala, and R. Piza, “A delay-dependent dual-rate PID controller over an Ethernet network,” *IEEE Transactions on Industrial Informatics*, vol. 7, pp. 18–29, Feb 2011.

- [86] C. Lai and P. Hsu, “Realization of networked control systems on Ethernet with varied time delay,” in *2010 IEEE International Conference on Systems, Man and Cybernetics*, pp. 66–73, Oct 2010.
- [87] C. Lai and P. Hsu, “The butterfly-shaped feedback loop in networked control systems for the unknown delay compensation,” *IEEE Transactions on Industrial Informatics*, vol. 10, pp. 1746–1754, Aug 2014.
- [88] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. Sastry, “Foundations of control and estimation over lossy networks,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 163–187, 2007.
- [89] M. Rostan, J. Stubbs, and D. Dzilno, “EtherCAT enabled advanced control architecture,” in *Advanced Semiconductor Manufacturing Conference (ASMC), 2010 IEEE/SEMI*, pp. 39–44, 2010.
- [90] K. Erwinski, M. Paprocki, L. M. Grzesiak, K. Karwowski, and A. Wawrzak, “Application of Ethernet POWERLINK for communication in a Linux RTAI open CNC system,” *IEEE Transactions on Industrial Electronics*, vol. 60, pp. 628–636, Feb 2013.
- [91] V. Q. Nguyen, N. V. P. Tran, H. N. Tran, K. M. Le, and J. W. Jeon, “A closed-loop stepper motor drive based on EtherCAT,” in *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, pp. 3361–3365, Oct 2017.
- [92] W. Zheng, H. Ma, and X. He, “Modeling, analysis, and implementation of real time network controlled parallel multi-inverter systems,” in *Power Electronics and Motion Control Conference (IPEMC), 2012 7th International*, vol. 2, pp. 1125–1130, 2012.
- [93] A. Krej, T. Popule, and M. Goubey, “Closing the motion control loops via industrial Ethernet network,” in *Proceedings of the 2014 15th International Carpathian Control Conference (ICCC)*, pp. 273–278, May 2014.

REFERENCES

- [94] J. Velagic, A. Kaknjo, N. Osmic, and T. Dzananovic, “Networked based control and supervision of induction motor using OPC server and PLC,” in *Proceedings ELMAR-2011*, pp. 251–255, Sept 2011.
- [95] Y. Tipsuwan and M.-Y. Chow, “Gain scheduler middleware: A methodology to enable existing controllers for networked control and teleoperation — Part I: Networked control,” *IEEE Transactions on Industrial Electronics*, vol. 51, pp. 1218–1227, Dec 2004.
- [96] Y. Tipsuwan and M.-Y. Chow, “Gain scheduler middleware: A methodology to enable existing controllers for networked control and teleoperation — Part II: Teleoperation,” *IEEE Transactions on Industrial Electronics*, vol. 51, pp. 1228–1237, Dec 2004.
- [97] C. Lozoya, P. Marti, M. Velasco, and J. M. Fuertes, “Effective real-time wireless control of an autonomous guided vehicle,” in *2007 IEEE International Symposium on Industrial Electronics*, pp. 2876–2881, June 2007.
- [98] T. Kumagai, H. Yoshida, and K. Satoda, “Adaptive remote control of a mobile robot system with delay fluctuation,” in *2018 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 1–5, Jan 2018.
- [99] J. Santos, A. Conceio, T. Santos, and H. Arajo, “Remote control of an omnidirectional mobile robot with time-varying delay and noise attenuation,” *Mechatronics*, vol. 52, pp. 7 – 21, 2018.
- [100] Y. Zeinaly, B. D. Schutter, and H. Hellendoorn, “An MPC scheme for routing problem in baggage handling systems: A linear programming approach,” in *2013 10th IEEE International Conference on Networking, Sensing and Control (ICNSC)*, pp. 786–791, April 2013.
- [101] M. Rostan, “XTS revolutionary drive technology enabled by EtherCAT,” 2012 (accessed on 2018-08-02). https://www.pc-control.net/pdf/032012/products/pcc_0312_xts_ethercat_e.pdf.

REFERENCES

- [102] Siemens AG, “SIMOTION The high-end motion control system,” 2018 (accessed on 2018-12-20). <https://www.siemens.com/global/en/home/products/automation/systems/motion-control.html>.
- [103] ODVA, “CIP motion,” 2018 (accessed on 2018-08-02). <https://www.odva.org/Technology-Standards/Common-Industrial-Protocol-CIP/CIP-Motio>.
- [104] X. Fang, K. Lim, L. Kyaw, X. Wu, and H. Yang, “Motor drive controller,” Jan 2015. U.S. Patent Application 14/281,491.
- [105] P. Gaj, J. Jasperneite, and M. Felser, “Computer communication within industrial distributed environment — A survey,” *IEEE Transactions on Industrial Informatics*, vol. 9, pp. 182–189, Feb 2013.
- [106] G. Cena, S. Scanzio, A. Valenzano, and C. Zunino, “Performance evaluation of the EtherCAT distributed clock algorithm,” in *2010 IEEE International Symposium on Industrial Electronics*, pp. 3398–3403, July 2010.
- [107] G. Cena, I. Bertolotti, S. Scanzio, A. Valenzano, and C. Zunino, “Evaluation of EtherCAT distributed clock performance,” *Industrial Informatics, IEEE Transactions on*, vol. 8, no. 1, pp. 20–29, 2012.
- [108] A. Ulusoy, O. Gurbuz, and A. Onat, “Wireless model-based predictive networked control system over cooperative wireless network,” *Industrial Informatics, IEEE Transactions on*, vol. 7, no. 1, pp. 41–51, 2011.
- [109] M. Knezic, B. Dokic, and Z. Ivanovic, “Topology aspects in EtherCAT networks,” in *Power Electronics and Motion Control Conference (EPE/PEMC), 2010 14th International*, pp. T1–1–T1–6, 2010.
- [110] X. Wu, L. Xie, and F. Lim, “EtherCAT-enabled next generation baggage handling systems,” in *Emerging Technologies Factory Automation (ETFA), 2013 IEEE International Conference on*, pp. 1–6, 2013.
- [111] EtherCAT Technology Group, “Industrial Ethernet technologies,” 2014 (accessed on 2018-10-25). https://www.ethercat.org/download/documents/Industrial_Ethernet_Technologies.pdf.

REFERENCES

- [112] Z. Hanzalek, P. Burget, and P. Sucha, “PROFINET IO IRT message scheduling with temporal constraints,” *Industrial Informatics, IEEE Transactions on*, vol. 6, no. 3, pp. 369–380, 2010.
- [113] L. Wisniewski, M. Schumacher, J. Jasperneite, and S. Schriegel, “Fast and simple scheduling algorithm for PROFINET IRT networks,” in *2012 9th IEEE International Workshop on Factory Communication Systems*, pp. 141–144, May 2012.
- [114] Microchip Technology Inc., “Microchip’s TCP/IP stacks,” 2018 (accessed on 2018-10-27). <https://www.microchip.com/SWLibraryWeb/product.aspx?product=TCPIPSTACK>.
- [115] Green Hills Software, “GHNet TCP/IP stack,” 2018 (accessed on 2018-10-27). https://www.ghs.com/products/comm_tcp-ip.html.
- [116] G. Cena, I. C. Bertolotti, S. Scanzio, A. Valenzano, and C. Zunino, “Synchronize your watches: Part I: General-purpose solutions for distributed real-time control,” *IEEE Industrial Electronics Magazine*, vol. 7, pp. 18–29, March 2013.
- [117] G. Cena, I. C. Bertolotti, S. Scanzio, A. Valenzano, and C. Zunino, “Synchronize your watches: Part II: Special-purpose solutions for distributed real-time control,” *IEEE Industrial Electronics Magazine*, vol. 7, pp. 27–39, June 2013.
- [118] Beckhoff Automation GmbH, “EtherCAT G: Ultimate I/O performance,” 2018 (accessed on 2018-12-30). https://download.beckhoff.com/download/Document/catalog/Beckhoff_EtherCAT_G_e.pdf.
- [119] Ethernet POWERLINK Standardization Group (EPSG), “POWERLINK basics,” 2014 (accessed on 2018-12-30). https://www.ethernet-powerlink.org/fileadmin/user_upload/Dokumente/Dokumente/PL_FLY_Basics_en_WEB.pdf.
- [120] W. Jian-Zhong, L. Fei-Ping, and C. Ji, “Research on robust tracking control of networked control systems,” in *2010 International Conference on Computing, Control and Industrial Engineering*, vol. 1, pp. 267–270, June 2010.

- [121] IEC, “IEC 61158-6-10: Industrial communication networks – Fieldbus specifications – Part 6-10: Application layer protocol specification – Type 10 elements,” 2014.
- [122] IEEE, “IEEE 802.1Q-2018: IEEE standard for local and metropolitan area networks — Bridges and bridged networks,” 2018.
- [123] P. Neumann and A. Pöschmann, “Ethernet-based real-time communications with PROFINET IO,” in *Proceedings of the 7th WSEAS International Conference on Automatic Control, Modeling and Simulation*, pp. 54–61, 2005.
- [124] H. Flatt, J. Jasperneite, and F. Schewe, “An FPGA based cut-through switch optimized for one-step PTP and real-time Ethernet,” in *2013 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication (ISPCS) Proceedings*, pp. 7–12, Sept 2013.
- [125] IEC, “IEC 61158-3-12: Industrial communication networks – Fieldbus specifications – Part 3-12: Data-link layer service definition – Type 12 elements,” 2014.
- [126] IEEE, “IEEE 802.11-2016: IEEE standard for information technology — Telecommunications and information exchange between systems — Local and metropolitan area networks — Specific requirements — Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications,” 2016.
- [127] IEEE, “IEEE 802.1D-2004: IEEE standard for local and metropolitan area networks — Media access control (MAC) bridges,” 2004.
- [128] C. Koulamas, S. Koubias, and G. Papadopoulos, “Using cut-through forwarding to retain the real-time properties of PROFIBUS over hybrid wired/wireless architectures,” *IEEE Transactions on Industrial Electronics*, vol. 51, pp. 1208–1217, Dec 2004.
- [129] J. Song, S. Han, A. Mok, D. Chen, M. Lucas, M. Nixon, and W. Pratt, “WirelessHART: Applying wireless technology in real-time industrial process control,” in *2008 IEEE Real-Time and Embedded Technology and Applications Symposium*, pp. 377–386, April 2008.

REFERENCES

- [130] Y. Xiao and J. Rosdahl, “Throughput and delay limits of IEEE 802.11,” *IEEE Communications Letters*, vol. 6, pp. 355–357, Aug 2002.
- [131] S. . Wang, Y. . Chen, T.-H. Lee, and A. Helmy, “Performance evaluations for hybrid IEEE 802.11b and 802.11g wireless networks,” in *PCCC 2005. 24th IEEE International Performance, Computing, and Communications Conference, 2005.*, pp. 111–118, April 2005.
- [132] A. Duda, “Understanding the performance of 802.11 networks,” in *2008 IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 1–6, Sept 2008.
- [133] M. Kaliwoda and C. Rehtanz, “Measuring delays of Ethernet communication for distributed real-time applications using carrier sense,” in *Industrial Informatics (INDIN), 2015 IEEE 13th International Conference on*, pp. 174–178, July 2015.
- [134] M. Zargari, L. Y. Nathawad, H. Samavati, S. S. Mehta, A. Kheirkhahi, P. Chen, K. Gong, B. Vakili-Amini, J. A. Hwang, S. M. Chen, M. Terrovitis, B. J. Kaczynski, S. Limotyrakis, M. P. Mack, H. Gan, M. Lee, R. T. Chang, H. Dogan, S. Abdollahi-Alibeik, B. Baytekin, K. Onodera, S. Mendis, A. Chang, Y. Rajavi, S. H. Jen, D. K. Su, and B. A. Wooley, “A dual-band CMOS MIMO radio SoC for IEEE 802.11n wireless LAN,” *IEEE Journal of Solid-State Circuits*, vol. 43, pp. 2882–2895, Dec 2008.
- [135] D. Ganz and H. Dermot Doran, “Extending summation-frame communication systems for high performance and complex automation applications,” in *Factory Communication Systems (WFCS), 2014 10th IEEE Workshop on*, pp. 1–8, May 2014.
- [136] F. Iwanitz, “Flexible real-time-Ethernet anschaltung mit FPGA,” *messtec drives Automation — Special real-time Ethernet*, 2011.
- [137] G. Cena, I. C. Bertolotti, A. Valenzano, and C. Zunino, “Reasoning about communication latencies in real WLANs,” in *2007 IEEE Conference on Emerging Technologies and Factory Automation (EFTA 2007)*, pp. 187–194, Sept 2007.

REFERENCES

- [138] M. Bertocco, C. Narduzzi, and F. Tramarin, “Estimation of the delay of network devices in hybrid wired/wireless real-time industrial communication systems,” in *2012 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, pp. 2016–2021, May 2012.
- [139] L. Seno, S. Vitturi, and F. Tramarin, “Influence of real components behavior on the performance of wireless industrial communication systems,” in *2011 IEEE International Symposium on Industrial Electronics*, pp. 1224–1229, June 2011.
- [140] F. Tramarin, C. Narduzzi, S. Vitturi, and M. Bertocco, “A calibrated test-set for measurement of access-point time specifications in hybrid wired/wireless industrial communication ,” *Information*, vol. 9, no. 5, 2018.
- [141] A. Cervin, D. Henriksson, B. Lincoln, J. Eker, and K. Arzen, “How does control timing affect performance? Analysis and simulation of timing using Jitterbug and TrueTime,” *Control Systems, IEEE*, vol. 23, pp. 16–30, June 2003.
- [142] IEC, “IEC 61784-3: Industrial communication networks – Profiles – Part 3: Functional safety fieldbuses – General rules and profile definitions,” 2016.
- [143] L. Li, P. Cong, K. Cao, J. Zhou, T. Wei, M. Chen, S. Hu, and X. S. Hu, “Game theoretic feedback control for reliability enhancement of EtherCAT-based networked systems,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1–1, 2018.
- [144] P. Ferrari, A. Flammini, D. Marioli, and A. Taroni, “Experimental evaluation of PROFINET performance,” in *Factory Communication Systems, 2004 IEEE International Workshop on*, pp. 331–334, 2004.
- [145] M. A. R. Siddique and J. Kamruzzaman, “Performance analysis of PCF based WLANs with imperfect channel and failure retries,” in *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*, pp. 1–6, Dec 2010.
- [146] G. Z. Khan, E. Park, and R. Gonzalez, “Performance analysis of early packet loss detection in WiFi direct 802.11 networks,” in *2017 IEEE 13th International*

REFERENCES

- Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 1–7, Oct 2017.
- [147] A. Varga and R. Hornig, “An overview of the OMNeT++ simulation environment,” in *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems*, pp. 1–10, 2008.
- [148] K. Yoshizawa, H. Hashimoto, M. Wada, and S. Mori, “Path tracking control of mobile robots using a quadratic curve,” in *Proceedings of Conference on Intelligent Vehicles*, pp. 58–63, Sept 1996.
- [149] J. Farkas, L. L. Bello, and C. Gunther, “Time-sensitive networking standards,” *IEEE Communications Standards Magazine*, vol. 2, pp. 20–21, June 2018.
- [150] Y.-B. Zhao, G. Liu, and D. Rees, “Integrated predictive control and scheduling co-design for networked control systems,” *Control Theory Applications, IET*, vol. 2, no. 1, pp. 7–15, 2008.
- [151] M. Branicky, S. Phillips, and W. Zhang, “Scheduling and feedback co-design for networked control systems,” in *Decision and Control, 2002 41st IEEE Conference on*, vol. 2, pp. 1211–1217, 2002.
- [152] M. Branicky, V. Liberatore, and S. Phillips, “Networked control system co-simulation for co-design,” in *2003 American Control Conference*, vol. 4, pp. 3341–3346, 2003.