

Kernel-based ARchitecture for safetY-critical cONtrol

KARYON
FP7-288195

D3.1 – First Report on Supporting Technologies

Work Package	WP3		
Due Date	M12	Submission Date	2012-10-26
Main Author(s)	Jörg Kaiser (OVGU), José Rufino (FFCUL), Elad Michael Schiller (CUT)		
Contributors	Tino Brade (OVGU), Sasanka Potluri (OVGU), Jeferson Souza (FFCUL) Luís Marques (FFCUL)		
Version	1.0	Status	Final
Dissemination Level	Public	Nature	Report
Keywords	Wireless protocols, inaccessibility analysis, adaptive middleware, reliable collaborative sensing, self-stabilizing protocols, assessment of global state.		



Part of the Seventh
Framework Programme
Funded by the EC - DG INFSO

Version history

Rev	Date	Author	Comments
V0.1	2012-04-17	Jörg Kaiser (OVGU)	Preliminary Draft release.
V0.2	2012-09-27	Jörg Kaiser (OVGU)	Partially complete version, with contributions from FFCUL, CUT and OVGU.
V0.3	2012-10-12	Jörg Kaiser (OVGU)	Updated version with additional inputs.
V0.4	2012-10-15	Jörg Kaiser (OVGU)	Revised version with corrections in formatting and updates according to internal review comments.
V0.5	-	Jörg Kaiser (OVGU)	Internal version of the editor.
V0.6	2012-10-22	Jörg Kaiser (OVGU)	Prefinal release.
V1.0	2012-10-26	António Casimiro (FFCUL)	Final review and delivery.

Glossary of Acronyms

ADA	Advanced Driver Assistant
API	Application Programming Interface
AUTOSAR	AUTomotive Open System ARchitecture
BO	Beacon Order
CAP	Contention Access Period
CAN	Controller Area Network
CFP	Contention Free Period
COTS	Commercial Off-The-Shelf
CTS	Clear To Send
CSMA/CA	Carrier Sense Multiple Access / Collision Avoidance
DoW	Description of Work
Dx.y	Deliverable belonging to work package x, with serial number y
EDS	Electronic Data Sheet
FIFO	First In First Out
GPS	Global Positioning System
GTS	Guaranteed Time Slot
EMS	Environment Modelling System
FAMOUSO	Family of Adaptive Middleware for autonomOUS Sentient Objects
FIFO	First In First Out
FMEA	Failure Modes and Effects Analysis
GEMS	Geometric Environment Modelling System
IEEE	Institute of Electrical and Electronics Engineers
IP	Inactive Period
IST	Information Society Technologies
KARYON	Kernel-based ARchitecture for safetY-critical cONtrol
LAN	Local Area Network
LIN	Local Interconnect Network
MAC	Medium Access Control
MLA	Mediator Layer
MLCCA	Multi-Level Composability Check Architecture
MOSAIC	MOdelling Sensors in Adaptive Interactive Control
OEM	Original Equipment Manufacturer
OS	Operating System

OSI	Open Systems Interconnection
PC	Personal Computer
P/S	Publisher/Subscriber
QoS	Quality of Service
RF	Radio Frequency
RPN	Risk Priority Number
RTE	RunTime Environment
RTS	Request To Send
SO	Superframe Order
TCP/IP	Transmission Control Protocol/Internet Protocol
TDMA	Time Division Multiple Access
TEDS	Transducer Electronic Datasheet
TTEthernet	Time-Triggered Ethernet
Tx.y	Task belonging to work package x, with serial number y
UDP	User Datagram Protocol
UID	Unique Identifier
VANET	Vehicular Ad Hoc Network
WLAN	Wireless Local Area Network
WP	Work Package
WPx	Work Package with serial number x
WSN	Wireless Sensor Networks
XML	Extensible Markup Language

Executive Summary

The objectives of WP3 are related to supporting technologies to enable dependable control, interaction, monitoring and fault management in the cooperative distributed system. WP3 constitutes the bridging work package between the more technology independent architecture and conceptual work packages WP2 and WP4 and the demonstration related work in WP5. WP3-tasks include the necessary technology analysis and provide system components needed to set up the demonstrators. The primary focus is on maintaining a high functional level in spite of environment uncertainties and faults of the communication and the computational systems. This report will provide an outline of the services and structure of the supporting technologies. As indicated in the DoW (Figure 9) this report deals mainly with the analysis of technology. Thus, basic characteristics and uncertainties of networks are analysed, first assessment of appropriate middleware is provided and concepts for deriving a global state based on local information are presented.

In particular, the report will encompass:

- Elaborating dependable quantification of temporal uncertainties affecting communication;
- Proposing protocols that deal with uncertainties of networks;
- Defining middleware specifically dealing with dynamic integration of remote sensor systems and cooperation between simulated and real components to enable hardware- and software-in-the-loop scenarios. Particular emphasis is on providing abstractions to handle faults and support environment perception.
- Developing solutions for reliable cooperation between mobile nodes. The KARYON project will study the design of reliable coordination algorithms for the proposed scenarios. This includes the consistent view about the operational state of cooperating entities and their intentions.

This deliverable describes the activities and achievements that have been made in the tasks T3.1, T3.2 and T3.3. Task T3.4 will start after the reporting phase. It should be noted that for the first phase of WP3 this report describes suitable techniques rather than an integration of the techniques. This is in line with the DoW that defines a major decision point after the first year. The selection and integration of technology will be during the first period of the next phase.

Table of Contents

1. Introduction	8
2. Predictability and Resilience in Embedded Networks	10
2.1 Network Inaccessibility	12
2.1.1 Network Inaccessibility in IEEE 802.15.4 wireless networks	13
2.1.2 Network inaccessibility scenarios in IEEE 802.15.4 wireless networks	13
2.1.3 Network inaccessibility results for IEEE 802.15.4 wireless networks	15
2.1.4 IEEE 802.15.4 network inaccessibility analysis tool	16
2.1.5 Summary	19
2.1.6 References	19
2.2 Self-* communication and synchronization primitives	20
2.2.1 Self-Stabilizing TDMA algorithms for Dynamic Wireless Ad-hoc Networks	22
2.2.2 Self-Stabilizing TDMA Alignment Algorithms for Dynamic Wireless Ad-hoc Networks	24
2.2.3 Self-Stabilizing End-to-End Communication in (Bounded Capacity, Omitting, Duplicating and non-FIFO) Dynamic Networks	27
2.2.4 References	31
3. Adaptive Middleware for Advanced Control Systems	33
3.1 Lightweight Dependable Adaptation for Wireless Sensor Networks	35
3.1.1 Overview	35
3.1.2 Technique	36
3.1.3 Evaluation	36
3.1.4 Conclusion	36
3.2 Communication middleware	37
3.3 Sensor middleware	40
3.4 Environment model	44
3.5 References	46
4. Reliable Assessment of Global State	50
4.1 Self-Stabilizing Byzantine Topology Discovery and Message Delivery	51
4.2 Capture Effect Based Communication Primitives	53
4.3 References	54
Annex A Papers and Reports	55
A.1 Predictability and Resilience in Embedded Networks	55
A.1.1 An Approach to Enhance the Timeliness of Wireless Communications	55
A.1.2 Characterization of Network Inaccessibility in IEEE 802.15.4 Wireless Networks ..	55
A.1.3 Characterizing Inaccessibility in IEEE 802.15.4 Through Theoretical Models and Simulation Tools	55
A.1.4 Reducing Inaccessibility in IEEE 802.15.4 Wireless Communications	55
A.1.5 Self-Stabilizing TDMA algorithms for Dynamic Wireless Ad-hoc Networks	55
A.1.6 Autonomous TDMA alignment for VANETs	55
A.1.7 Self-Stabilizing End-to-End Communication in Bounded Capacity, Omitting, Duplicating and Non-FIFO Dynamic Networks	56

A.2 Adaptive Middleware for Advanced Control Systems	56
A.2.1 Lightweight Dependable Adaptation for Wireless Sensor Networks	56
A.2.2 Programming abstractions and middleware for building control systems as networks of smart sensors and actuators.....	56
A.2.3 A fault-aware sensor architecture for cooperative mobile applications.....	56
A.3 Reliable Assessment of Global State.....	56
A.3.1 Self-Stabilizing Byzantine Resilient Topology Discovery and Message Delivery	56
A.3.2 Capture effect based communication primitives	56

List of Figures

Figure 1: The KARYON R2T-MAC architecture.	11
Figure 2: Specific sources of inaccessibility incidents in wireless networks.	12
Figure 3: Superframe structure of the IEEE 802.15.4 in beacon-enabled mode.	13
Figure 4: The IEEE 802.15.4 network inaccessibility durations normalized by, and compared with, TBI (TBI=123 ms).	16
Figure 5: The spreadsheet-based tool interface for MAC parameter configuration.	16
Figure 6: The MLA parameter configuration sheet.	17
Figure 7: The duration of network inaccessibility scenarios for IEEE 802.15.4 wireless networks.	18
Figure 8: The system structure of the supporting technology perspective.	21
Figure 9: Self-stabilizing TDMA-based MAC algorithm.	22
Figure 10: Contour plot of equation (1) for $s = d/T = 1$	24
Figure 11: Unaligned TDMA timeslots.	25
Figure 12: The cricket strategy.	25
Figure 13: Packets formation from the messages in S2E2C algorithm.	30
Figure 14: Layers of middleware.	34
Figure 15: Channel concept implemented in FAMOUSO.	37
Figure 16: Cooperative vehicle development platform.	39
Figure 17: Fault tolerant sensor node providing an abstract output interface that includes a system and event validity value.	41
Figure 18: Abstract representation of the 5-step processing and selection chain inside a MOSAIC node.	42
Figure 19: Modular structure of a MOSAIC sensor node that implements the concepts of Figure 18.....	42
Figure 20: Example of a MOSAIC application implementing the distance control of a mobile robot.....	43
Figure 21: Parking assistance scenario.....	45
Figure 22: Environment model.....	46

List of Tables

Table 1 - Easy-to-use formulas defining the durations of periods of network inaccessibility	14
--	----

1. Introduction

The objectives of WP3 are related to supporting technologies to enable dependable control, interaction, monitoring and fault management in the cooperative distributed system. WP3 constitutes the bridging work package between the more technology independent architecture and conceptual work packages WP2 and WP4 and the demonstration related work in WP5. WP3-tasks include the necessary technology analysis and provide system components needed to support cooperation of autonomous vehicles and set up the demonstrators described in WP5.

The technical contributions described in this report are structured along the tasks defined in WP3. The first part is devoted to predictability and resilience in embedded wireless networks. Providing predictability like temporal guarantees in a wireless communication system may follow two different but complementary approaches depending on the dynamic properties of the network. One approach strives for monitoring the network, predicting inaccessibility times and providing means to put bounds on these inaccessibility times. A major source of unpredictability is depending on the used network arbitration that is sensitive with respect to load and faults. Predicting the network status requires carefully analysis of inaccessibility times of the (Medium Access Control) MAC protocol. In this report such an analysis is carried out for the widely used 802.15.4 standard for wireless communication. It is also described how the protocol can be wrapped into additional software layers to enable lightweight adaptation to changing network conditions.

The second approach to predictable communication is based on avoiding any arbitration conflicts. Some of the most frequently used schemes for medium access control are Time Division Multiple Access (TDMA), Collision Avoidance (CSMA/CA) and back offs. We study different combinations of these frequently used schemes from the perspectives of robustness, autonomous implementation, self-organizing and self-recovery algorithms, i.e., self-stabilization. We discover that, unlike existing implementations, there are self-stabilizing MAC protocols for dynamic wireless networks. We study one of them and show, via rigorous analysis, simulation and test-bed experiments, that it can facilitate, after a brief short convergence/recovery period, the satisfaction of requirements, such as high bandwidth utilization, high predictability degree of bandwidth allocation, and low communication delay in the presence of frequent topological changes to the communication network. In addition, we present self-organizing and self-recovery solutions to timeslot alignment problem for TDMA timeslot and self-stabilizing solutions for data-link layer in (message omitting, duplicating and non-FIFO) dynamic wireless networks. Besides the contribution in the algorithmic front of research, we expect that our proposal can serve as a preferable alternative to the existing implementations of the IEEE 802.11p and by that enable quicker adoption of vehicular ad-hoc network (VANETs).

It is anticipated that cooperative systems, that are in fact a system of systems, will exhibit a large variety of physical network media, protocols, addressing schemes, QoS properties and failure conditions. Implementing distributed cooperative applications therefore would largely benefit from hiding these details of heterogeneity from the communicating objects. The second major part of this report is devoted to adaptable middleware concepts. The proposed adaptive middleware FAMOUSO is specifically designed to deal with heterogeneity and different QoS attributes of networks in distributed control. It provides a uniform event-based communication scheme. Events are disseminated in a publish/subscribe style. In contrast to other publish/subscribe systems we introduce the notion of an event channel. Event channels abstract the (multiple) underlying physical networks and allow specifying QoS attributes like latencies, jitter, or delivery guarantees on a high level of abstraction. Obviously, the required QoS attributes may not always meet what the networks are able to provide. When announcing a channel, knowledge from the network layers is acquired e.g. by continuous monitoring for checking whether the QoS requirements can be met. Benefitting from the uniform event

communication, a middleware support for smart sensors (MOSAIC) is provided on top of the event system. It realizes the scheme of reliable sensors that has been elaborated in WP2. Although the notion of an abstract sensor already provides application relevant information rather than raw sensor data, this information often can only be interpreted in the context of a specific environment. This is particularly true for interpreting data from a camera, a laser scanner or some other distance measuring equipment where the origin and the direction of the sensor must be known. To ease this task, we provide a layer that allows access to an appropriate environment model. The general investigations on suitable environment models will be performed in WP2 but are provided as an easy to use set of services by the middleware.

The high level of abstraction and the well-defined objects and interfaces in the middleware decouple an application to a large extent from the underlying physical system. As outlined in the report, this is also exploited for mixed reality systems in which simulated and real components coexist.

Cooperation needs communication and coordination. The last section therefore deals with issues of coordination between vehicles. We aim at developing solutions for reliable cooperation between mobile nodes. This includes the need to achieve a consistent view about the operational state of cooperating entities. We study what is possible and impossible to achieve within a predictable time bound when considering both Byzantine and benign system settings. It is not clear how can one implement (Byzantine) fault tolerant agreement protocols without a (cryptographic-free) way for learning about the distributed system state and network. We also look into efficient implementation of agreement protocols in benign wireless ad hoc networks.

2. Predictability and Resilience in Embedded Networks

Industrial, defence and vehicular (automotive, aerial, aerospace) are examples of functional application domains where the use of communications with strict requirements is essential. Messages between the different entities (e.g. several vehicles) should not only be exchanged reliably but also on time. Activities such as monitoring and controlling the temperature and pressure of machinery, coordination of a set of cooperative unmanned terrestrial or aerial vehicles, and real-world interfacing through networked sensors and actuators are just a few examples where predictability, dependability and timeliness of communications are a must.

Wired networks have been studied during the last decades establishing an extremely reliable network infrastructure, providing real-time communication support to networked systems and applications. Many of these systems have been built on top of industry standards such as Controller Area Network (CAN) [1] or using special-purpose solutions such as Time-Triggered Ethernet (TTEthernet) [2]

Despite their proved usefulness, the use of wired networks exhibits, in some environments, a set of disadvantages making the use of cabling complex and difficult. For example: high costs in the deployment and maintenance of physical medium (cables); lack of space for cable installation; cable volume and weight; a need for easy and/or heavy mobility. Thus, there may be a real need and certainly there is the interest in introducing the usage of wireless networks in real-time communication domains. However, standard wireless networks alone cannot provide the levels of dependability, and the safety and timeliness properties, required for the use of wireless communications in environments with strict real-time communication requirements.

A fundamental issue concerns a potential lack of determinism of traditional MAC protocols used in wireless communications. Classical approaches to this problem have been based on completely new MAC protocol designs [3,4,5,6,7], modifications to existing standards [8,9,10], and abstract models [11]. Though these solutions aim to enhance the real-time guarantees and reliability of wireless communications, compatibility with wireless network standards is not always a concern, which may rule out the use of standard off-the-shelf technology.

Another key point is that wireless communications are especially susceptible to errors, namely to disturbances in the wireless communication medium. These disturbances may be generated by different sources, such as external electromagnetic interferences, temporary obstacles in the communication path, impairments of electronic radio frequency (RF) circuits [12], interferences of adjacent channels utilization [13], or even malicious attacks [14,15] originated by external/internal sources such as electromagnetic jamming devices or compromised nodes. That means, a set of both accidental and intentional incidents may disturb the operation of MAC protocols [16].

Standard MAC protocols have means to detect the effects of accidental faults and execute additional procedures to re-establish normal MAC protocol operation. During the execution of such recovery procedures, the MAC layer although cannot be considered failed does not provide service, creating periods of network inaccessibility. Intentional faults need to be handled by additional measures and will be addressed in future work. When network inaccessibility incidents occur communications cannot be performed.

Since the periods of network inaccessibility may have durations much higher than the normal worst case network access delay, inaccessibility incidents do represent a source of unpredictability. The effects of network inaccessibility may propagate to the higher layers of a communication stack – usually a collapsed version of the Open Systems Interconnection (OSI) reference model – potentially disrupting the operation of services and applications. As a consequence, the overall dependability, predictability and timeliness properties of the system may be at risk, being compromised at the communication service.

To enhance the predictability and timeliness of wireless networks it is proposed to address the shortcomings of standard network technologies with this respect at the lowest possible level. Furthermore, some specific constructs fundamental to build distributed applications may be missing, or at least are not properly provided off-the-shelf by existent standard wireless networks. For example, a global notion can be helpful in the evaluation of data validity, in the support of coordinated actions and in the implementation of cooperative applications.

The KARYON project aims to use the design principles and techniques successively applied in wired networks to the realm of wireless network technologies, while maintaining compatibility with the standard specifications.

This approach is supported on an innovative extensible component architecture which surrounds the standard MAC level with additional components designed to extend and enhance its native characteristics. The extra components allows to improve MAC level predictability and resilience against accidental disturbances in medium and medium access levels, providing a service layer interface with enhanced predictability and timeliness properties intended to simplify the development of networked real-time protocols and applications.

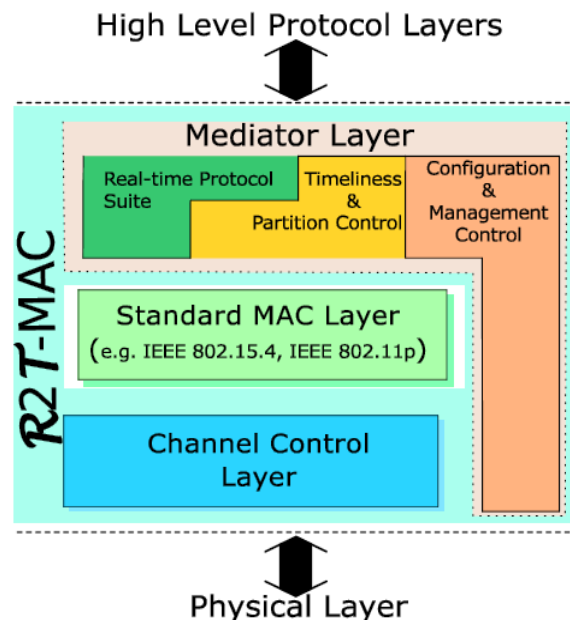


Figure 1: The KARYON R2T-MAC architecture.

The architecture illustrated by Figure 1 is composed by two different layers: MLA, the Mediator Layer and the Channel Control Layer, surrounding a standard MAC layer. This means, such a solution can be incorporated in Commercial Off-The-Shelf (COTS) components without fundamental modifications in the standard MAC level protocol.

The Mediator Layer intermediates the communication and provides error isolation between the MAC and higher layers, minimizing the negative effects caused by disturbances in the medium and medium access control protocols. This is a standard-compliant solution which extends MAC layer services with additional features and guarantees, enhancing the predictability and timeliness of wireless communications. The Mediator Layer may include components to provide services such as reliable and real-time frame transmissions, node failure detection and membership, control of temporary network partitions (inaccessibility), control of resilience of communications, and management of MAC layer and its configurations.

The Channel Control Layer is designed to allow the control of channel state, and also to improve the network resilience by profiting from the control of the diversity of radio channels used on the communication medium.

2.1 Network Inaccessibility

Disturbances induced in the operation of MAC protocols may create temporary partitions in the network, derived of the time required to detect and recover from these situations. These disturbances can be produced by external interferences or by some glitches in the operation of the MAC layer. A solution for controlling these partitions for LAN-based networks was presented in [17]. These temporary network partitions are called periods of network inaccessibility [18, 19] and the definition of this concept is summarized here:

*Certain kinds of components may temporarily refrain from providing service, without that having to be necessarily considered a failure. That state is called **inaccessibility**. It can be made known to the users of network components; limits are specified (duration, rate); violation of those limits implies permanent failure of the component.*

Though many sources of inaccessibility incidents are common to wire and wireless networks, some causes are specific of the wireless realm. For example, some LAN-based networks may offer the capability to detect transmission errors, e.g. due to collisions. In wireless networks these mechanisms do not exist. In wireless networks, the transceiver of each node cannot receive and transmit data simultaneously. Consequently, the algorithms used in the MAC layer do not have means to detect a collision without the support of frame-driven timeout-based mechanisms, or additional control methods.

Frame transmission disturbances may also be derived of the proximity and position of a node in relation to the operating space of other nodes. The circles in Figure 2 show the transmission and interference range of three different nodes. In the example presented in Figure 2, the node A may overlap, total or partial, the frame transmission of node B and vice-versa. This may result in periods of inaccessibility for the two nodes. The RTS/CTS handshake used in IEEE 802.11 tries to solve the hidden node problem. However, this technique does not solve completely the problem and increase the overhead of a transmission, an unacceptable condition, for example, for wireless sensor networks [20].

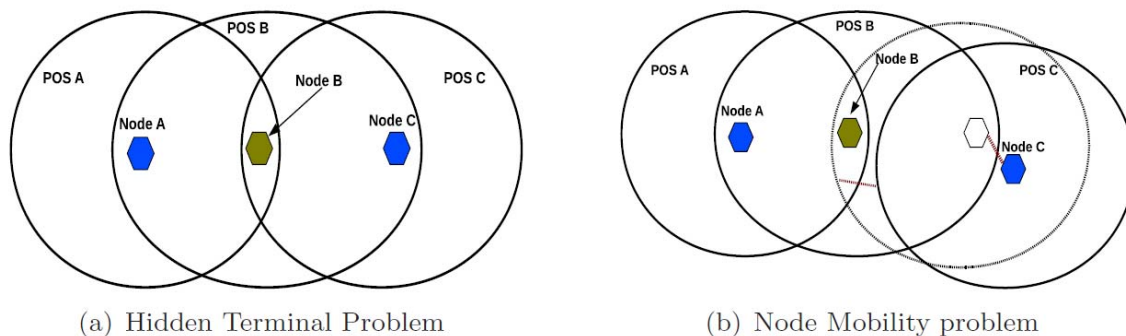


Figure 2: Specific sources of inaccessibility incidents in wireless networks.

The node mobility, provided by wireless technology, allows the change of a node location. This mobility may cause loss of communication between nodes. Figure 2-b shows that, after moving, node C is outside the range of node B and that as such situation induces a period of inaccessibility seen by both nodes. An environment with heavy levels of node mobility may cause the occurrence of various periods of inaccessibility if the nodes will move to positions out of range of each other. The period of network inaccessibility includes, in this case, the time a node is out of range and the time needed to re-establish normal operation of the MAC protocol.

The knowledge of inaccessibility time bounds is fundamental to achieve the support of real-time communication over wireless networks. The inaccessibility characteristics depend on the network alone and can be predicted by the analysis of the MAC protocol. Given the durations of inaccessibility incidents are known, a worst case bound can be integrated in message

schedulability analysis and in communication protocol execution, at the different levels of the system. Optimal network inaccessibility control algorithms allow the real durations of network inaccessibility incidents to be included in protocol execution, instead of the corresponding worst case bound [21], as detailed in the article reprint provided in Annex A.1.1.

In annex: reprint of the paper “An Approach to Enhance the Timeliness of Wireless Communications”. J. L. R. Souza and J. Rufino, Fifth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM 2011), November 2011, Lisbon, Portugal

2.1.1 Network Inaccessibility in IEEE 802.15.4 wireless networks

The IEEE 802.15.4 has two operation modes dubbed nonbeacon-enabled and beacon-enabled. Let us to focus our attention on the beacon-enabled mode, designed to support traffic with temporal restrictions. In a beacon-enabled mode there is a coordinator node that manages and controls the network access. The coordinator uses the superframe structure represented in the diagram of Figure 3 to control the access to the network. The duration of a superframe is calculated utilizing a constant that defines the minimum (also known as base) superframe duration, T_{BSD} , and a beacon order exponent, BO , which is utilized to determine the actual time interval between consecutive beacon frames, T_{BI} , as given by:

$$T_{BI} = T_{BSD} \cdot 2^{BO}$$

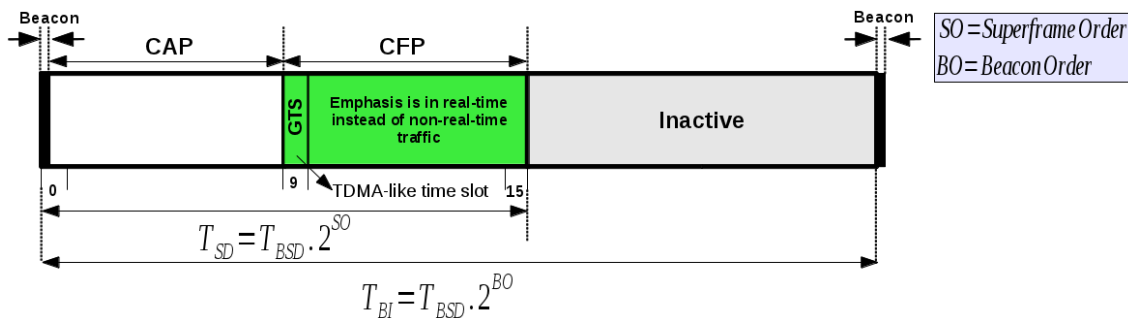


Figure 3: Superframe structure of the IEEE 802.15.4 in beacon-enabled mode.

As illustrated by Figure 3, a superframe has a contention access period (CAP), where nodes compete in equal condition to access the network in a non-real-time manner; a contention free period (CFP), where nodes access the network within exclusive time slots (GTS, the Guaranteed Time Slots), supporting real-time traffic in a similar manner of time division multiple access (TDMA) approaches; and optionally an inactive period (IP), where nodes may enter in a power-save mode. A node may also allocate more than one contiguous time slot for exclusive and contention-free access.

The CAP and CFP together represent the active portion of the superframe structure, which has a duration given by:

$$T_{SD} = T_{BSD} \cdot 2^{SO}$$

where, SO is the superframe order exponent that defines the duration of this active portion. If SO and BO are identical there is no IP within the superframe.

2.1.2 Network inaccessibility scenarios in IEEE 802.15.4 wireless networks

This section provides an overview of network inaccessibility in IEEE 802.15.4 wireless communications, presenting a comprehensive set of relevant network inaccessibility scenarios, being their worst case durations represented by the superscript (^{wc}). A more detailed

characterization of network inaccessibility in IEEE 802.15.4 wireless communications can be found in [16] and in the detailed Technical Report of Annex A.1.2.

In annex: “Characterization of Network Inaccessibility in IEEE 802.15.4 Wireless Networks”. J. L. R. Souza and J. Rufino, Technical Report DI/FCUL, September 2012, Lisbon, Portugal

A summary of the set of easy-to-use formulas defining the (worst case) durations of the different inaccessibility scenarios is presented in Table 1.

Scenario	Equation
Single Beacon Frame Loss	$T_{ina\leftarrow sbfl}^{wc} = T_{BSD} \cdot (2^{BO} + 1)$
Multiple Beacon Frame Loss	$T_{ina\leftarrow mbfl}^{wc} = T_{BSD} \cdot (2^{BO} + 1) \cdot nrLost$
Synchronization Loss	$T_{ina\leftarrow nosync} = T_{BSD} \cdot (2^{BO} + 1) \cdot nrLost$
Orphan Node	$T_{ina\leftarrow orphan}^{wc} = T_{ina\leftarrow nosync} + T_{MLA}(Orphan) + \sum_{j=1}^{nrchannels} (T_{MAC}^{wc}(Orphan) + nrWait \cdot T_{BSD}) + T_{MAC_ack}^{wc}(Realign)$
Coordinating Orphan Realignment	$T_{ina\leftarrow realign}^{wc-sn} = T_{MAC_ack}^{wc}(Realign) + T_{MLA}(Realign)$
Coordinator Conflict Detection	$T_{ina\leftarrow C_Detection}^{wc} = T_{MAC_ack}^{wc}(C_Conflict)$
Coordinator Conflict Resolution	$T_{ina\leftarrow C_Resolution}^{wc} = T_{MLA}(C_Conflict) + \sum_{j=1}^{nrchannels} [T_{MAC}^{wc}(Beacon_R) + nrWait \cdot T_{BSD}] + T_{MLA}(Realign) + T_{MAC}^{wc}(Realign)$
Extract Request	$T_{ina\leftarrow extReq}^{wc} = T_{MAC_ack}^{wc}(ExtReq)$
Association	$T_{ina\leftarrow assoc}^{wc} = \sum_{j=1}^{nrchannels} [T_{MAC}^{wc}(Beacon_R) + nrWait \cdot T_{BSD}] + T_{MLA}(Beacon) + T_{ina\leftarrow extReq}^{wc} + T_{MLA}(AssocReq) + T_{MAC_ack}^{wc}(AssocReq)$
Re-Association	$T_{ina\leftarrow reAssoc}^{wc} = T_{ina\leftarrow nosync} + T_{ina\leftarrow assoc}^{wc}$
GTS request	$T_{ina\leftarrow GTS}^{wc} = T_{MAC_ack}^{wc}(GTS)$

Table 1: Easy-to-use formulas defining the durations of periods of network inaccessibility.

The beacon frame controls the access to the network, and its reception is essential to maintain all the nodes synchronized within the different periods of the superframe structure. If a beacon frame is not correctly received an inaccessibility incident occurs. Thus, a single beacon frame loss occurs when only one beacon is lost. The value of this period of inaccessibility is T_{BI} plus one T_{BSD} period, which is utilized as a margin to overcome some clock deviations that may occur between nodes.

The multiple beacon frame loss occurs when multiple and consecutive beacons are lost and a correct beacon frame is successfully received after the loss of $nrLost$ beacons.

The synchronization loss is a special case of the multiple beacon frame loss scenario where after the loss of $nrLost$ beacons the next beacon is also lost.

To recover from such loss of synchronization two different strategies were identified in the standard specification [22]. Each individual node chooses the recovery strategy to be used. If some data/control frame was received during the last beacon interval, the node assumes an orphan status; otherwise, a re-association procedure should be carried out. In both recovery strategies, the node looks for a coordinator in the given set of logical channels¹. After the channel scan, an coordinator realignment or an (re-)association procedure is performed within the orphan and re-association scenarios, respectively.

In the execution of the association procedure, the channel scan is followed by a beacon processing action, the extract of control information, an association processing action and the actual association with the coordinator. The re-association and association procedures are quite

¹ A logical channel is a numerical representation of a radio frequency (RF) channel utilized by the MAC layer to perform its network communications.

equivalent. The association procedure is executed when a non-coordinator node has no information about its coordinator.

A coordinator conflict occurs when more than one coordinator are active within the same network. By default, each network has a unique identifier, *networkID*, which identifies the network uniquely and is used by the coordinator in beacon transmissions. If some other (possibly old) coordinator enters the network operation space, e.g., after moving away during a long period of time, the network may have two different coordinators transmitting beacons with the same *networkID*. To solve such conflict, the actual coordinator performs a search within a set of specified logical channels. If the coordinator does not found other coordinator sending beacons with its own identifier after the scan in all logical channels, no further action is taken and the network becomes accessible again. Otherwise, a new identifier is selected and, if necessary, a MAC coordinator realignment command is broadcast. In Table 1, this scenario is separated in two individual contributions: coordinator conflict detection, to be performed upon the detection of a coordinator conflict and its notification; a longer coordinator conflict resolution procedure, which includes the logical channel search procedure.

The final scenarios do include the procedure required for requesting the allocation of a GTS slot and the procedure to extract control information from the coordinator.

The different parameters used in the formulas of Table 1 are as follows: *nrchannels*, represents the number of channels to be scanned; *nrWait*, defines the waiting period for a beacon frame in each channel scan, assuming the default value of $nrWait=32$ in the IEEE 802.15.4 standard; $T_{MAC_ack}(frame)$ and $T_{MAC}(frame)$ represent the delay from request to confirmation of a MAC frame transmission time with and without acknowledgement, respectively; $T_{MLA}(action)$ represents the time needed to perform the specified action at the MAC management layer.

Without loss of generality, a uniform value of $T_{MLA}(action) = T_{BI} / 10$ is assumed for the duration of each MAC management layer action.

A preliminary validation of the IEEE 802.15.4 network inaccessibility theoretical model using the NS-2 simulator has been published in [23] and is presented in Annex A.1.3.

In annex: reprint of the paper “**Characterizing Inaccessibility in IEEE 802.15.4 Through Theoretical Models and Simulation Tools**”. J. L. R. Souza, A. Guerreiro and J. Rufino. INForum 1012 Simpósio de Informática – Embedded and Real-Time Systems Track. September 2012, Caparica, Portugal.

2.1.3 Network inaccessibility results for IEEE 802.15.4 wireless networks

Figure 4 summarizes the numeric results obtained for an IEEE 802.15.4 wireless network. The results are normalized by the duration of T_{BI} , the beacon interval. Since these periods of inaccessibility may have durations much higher than the normal network access delay, T_{BI} , the reduction of these periods is of fundamental importance to enhance the predictability and timeliness properties of the network. The FFCUL research effort performed towards this direction is presented in Annex A.1.4.

In annex: pre-print of the paper “**Reducing Inaccessibility in IEEE 802.15.4 Wireless Communications**”. J. L. R. Souza and J. Rufino, (submitted for publication).

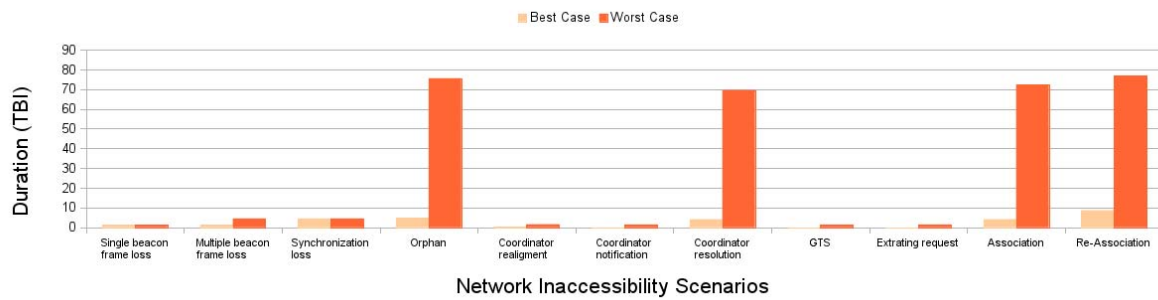


Figure 4: The IEEE 802.15.4 network inaccessibility durations normalized by, and compared with, T_{BI} ($T_{BI}=123$ ms).

2.1.4 IEEE 802.15.4 network inaccessibility analysis tool

A spreadsheet-based tool able to calculate and analyze the periods of network inaccessibility in IEEE 802.15.4 wireless networks, as given by our theoretical model, was designed with usability in mind. The tool is composed by different sheets, but only the sheets/cells specifying the network/model configuration parameters can be modified. The tool was designed to allow the configuration of IEEE 802.15.4 platforms in a simple and intuitive way, as illustrated in Figure 5, with respect to *MAC parameter configuration*.

These range and semantics defined in the IEEE 802.15.4 standard for each MAC configuration parameter specified in Figure 5. Additional explanations are introduced for each of the parameters to help the user to understand the purpose of each specific parameter and how such parameter influences the temporal behaviour of the network. Such explanations show up when a parameter value is selected.

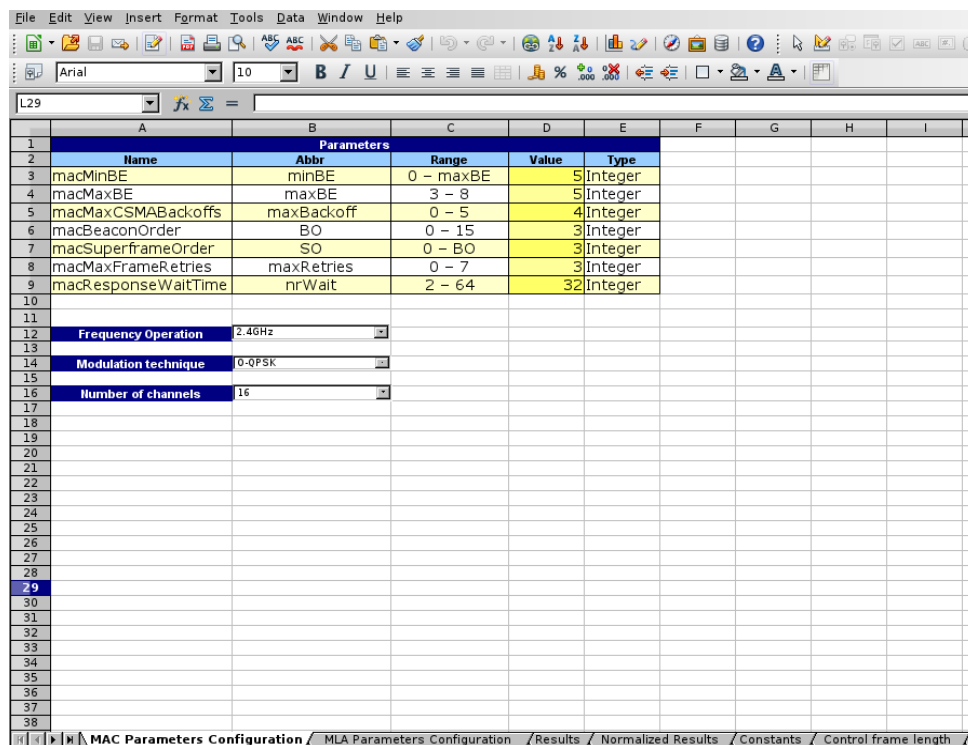
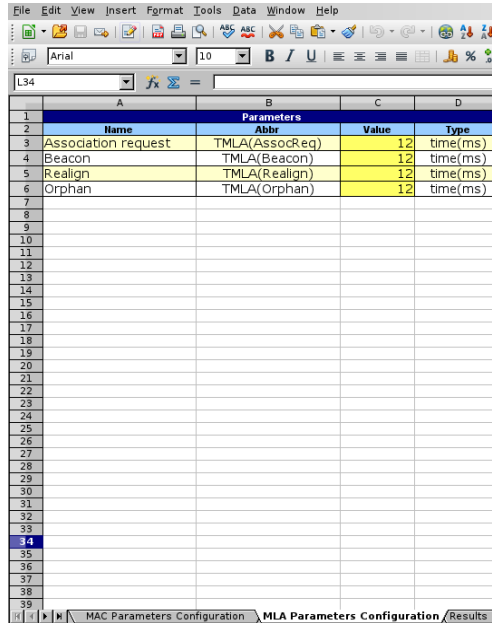


Figure 5: The spreadsheet-based tool interface for MAC parameter configuration.

The *MLA Parameter Configuration* sheet provides a way to define the duration of MAC management operations, which are not explicitly defined in the IEEE 802.15.4 standard. By default, each MLA management action is set to an uniform value of $T_{MLA}(action) = T_{BI}/10$, as illustrated in . However, these values can be changed and replaced by the real processing times of each IEEE 802.15.4 specific platform. The *MLA Parameter Configuration* sheet is illustrated in Figure 6.

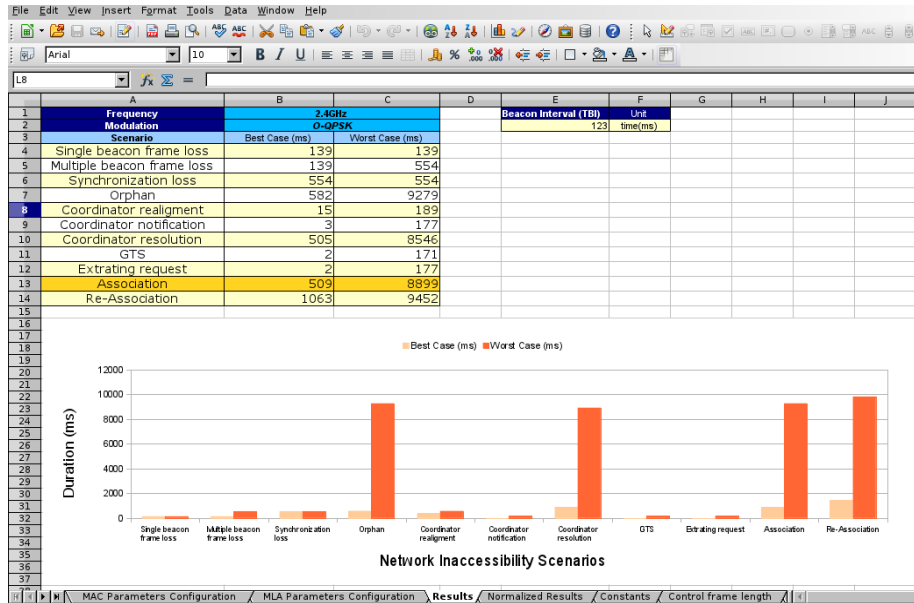


	A	B	C	D
1	Parameters			
2	Name	Abbr	Value	Type
3	Association request	TMLA(AssocReq)	12	time(ms)
4	Beacon	TMLA(Beacon)	12	time(ms)
5	Realign	TMLA(Realign)	12	time(ms)
6	Orphan	TMLA(Orphan)	12	time(ms)
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				
31				
32				
33				
34				
35				
36				
37				
38				
39				

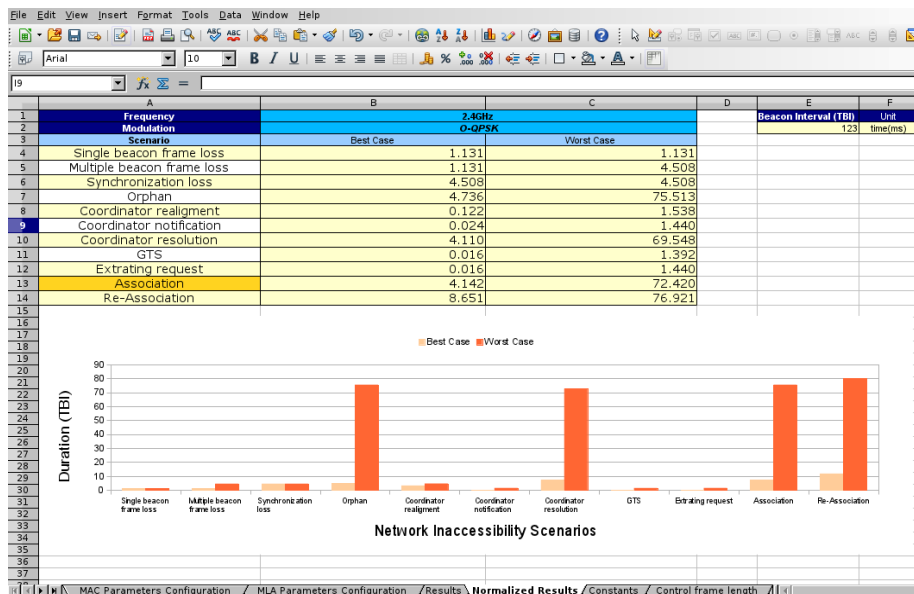
Figure 6: The MLA parameter configuration sheet.

Additional *Control frame length* and *Constants* sheets specify the duration of MAC control frames and the value of the constants utilized by the MAC layer, respectively. Since these values are fixed by the IEEE 802.15.4 standard specification, these sheets cannot be modified by the user.

Our tool uses the values specified and defined in the aforementioned sheets to calculate the network inaccessibility duration, presenting the results of such calculation within the *Results* and *Normalized Results* sheets. The *Result* sheet presents the network inaccessibility duration in millisecond. To complement these true-time results, the *Normalized Results* sheet provides the network inaccessibility durations, where the temporal unit is the beacon interval, T_{BI} . Using the results presented in the *Normalized Results* sheet we can compare the network inaccessibility with the duration of network cycle, which can be utilized to analyze the impact of network inaccessibility in the network operation itself.



a) Results



b) Normalized Results

Figure 7: The duration of network inaccessibility scenarios for IEEE 802.15.4 wireless networks.

Figure 7 presents the *Results* and *Normalized Results* sheets. Additionally, the network inaccessibility results illustrated by Figure 7 can be utilized by other tools, such as network simulators or schedulability analyzers, to evaluate the impact of network inaccessibility in a variety of scenarios and situations.

The spreadsheet-based tool is freely available on the KARYON web site, under the following link: <http://www.karyon-project.eu/documents/software-tools/>. An office suite software with full support and compliance with the Open Document Format for Office Applications (ODF) is required to use the tool. LibreOffice (<http://www.libreoffice.org/download>) is one example of such office suite software. The execution of Macros must be enabled, being a mandatory requirement to use our spreadsheet-based tool.

2.1.5 Summary

This section presented the work developed in the context of the KARYON Project with respect to the enhancement of the predictability and resilience properties of wireless based embedded networks. Taking the IEEE 802.15.4 standard as a use-case, we investigated the inaccessibility characteristics of such wireless networks, defined policies to reduce the duration of inaccessibility incidents and provide a general method to control inaccessibility, to be integrated in top of the MAC exposed interface, as part of the Mediator Layer. The Mediator Layer intermediates the communication and provides error isolation between the MAC and higher layers, minimizing the negative effects caused by disturbances in the medium and medium access control protocols, allowing the use of COTS components. Furthermore, an easy-to-use tool, based on the LibreOffice suite, allowing the evaluation of the different inaccessibility incident durations on different network conditions is publically available.

2.1.6 References

- [1] J. Rufino; C. Almeida; P. Veríssimo; G. Arroz. “Enforcing Dependability and Timeliness in Controller Area Networks”. “32nd Annual Conference of the IEEE Industrial Electronics Society (IECON)”. Paris, France, November 2006.
- [2] H. Kopetz; A. Ademaj; P. Grillinger; K. Steinhammer. “The time-triggered Ethernet (TTE) Design”. “8th IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC)”. Washington, DC, USA, May 2005.
- [3] A. Sahoo; P. Baronia. “An Energy Efficient MAC in Wireless Sensor Networks to Provide Delay Guarantee”. “15th IEEE Workshop on Local Metropolitan Area Networks (LANMAN)”. Princeton NJ, USA, June 2007.
- [4] I. Aad; P. Hofmann; L. Loyola; F. Riaz; J. Widmer. “E-MAC: Self-Organizing 802.11-Compatible MAC with Elastic Real-time Scheduling”. “4th IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS)”. Pisa, Italy, October 2007.
- [5] E. Egea-López; J. Vales-Alonso; A. S. Martínez-Sala; J. García-Haro; P. Pavón-Mariño; M. V. Bueno Delgado. “A Wireless Sensor Networks MAC Protocol for Real-time Applications”. “Journal of Personal Ubiquitous Computing”. vol. 12. pages 111–122. Springer, February 2008.
- [6] P. Bartolomeu; J. Ferreira; J. Fonseca. “Enforcing Flexibility in Real-time Wireless Communications: A Bandjacking Enabled Protocol”. “14th IEEE International Conference on Emerging Technologies Factory Automation (ETFA)”. Mallorca, Spain, September 2009.
- [7] X. Y. Shuai; Z. C. Zhang. “Research of Real-time Wireless Networks Control System MAC Protocol”. “Journal of Networks”. vol. 5. pages 419–426. Academic Publisher, April 2010.
- [8] Y.K. Huang; A. C. Pang; H. N. Hung. “An Adaptive GTS Allocation Scheme for IEEE 802.15.4”. “IEEE Transactions on Parallel and Distributed Systems”, vol. 19. pages 641–651, IEEE Computer Society, May 2008.
- [9] M. Hameed; H. Trsek; O. Graeser; J. Jasperneite. “Performance Investigation and Optimization of IEEE 802.15.4 for Industrial Wireless Sensor Networks”. “13th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)”. Hamburg, Germany, September 2008.
- [10] A. Koubâa; A. Cunha; M. Alves; E. Tovar. “An Implicit GTS Allocation Mechanism in IEEE 802.15.4 for Time-Sensitive Wireless Sensor Networks: Theory and Practice”. “Real-Time Systems Journal”. vol. 39. pages 169–204. Springer, August 2008.

- [11] F. Kuhn; N. Lynch; C. Newport. “The Abstract MAC Layer”. “23rd International Symposium On Distributed Computing (DISC)”. Elche/Elx, Spain, September 2009.
- [12] T. Schenk; E. Fledderus. “RF Impairments in High-rate Wireless Systems - Understanding the Impact of TX/RX-Asymmetry”. “3rd International Symposium on Communications, Control and Signal Processing (ISCCSP)”, St. Julian’s, Malta, March 2008.
- [13] W. L. Tan; K. Bialkowski; M. Portmann. “Evaluating Adjacent Channel Interference in IEEE 802.11 Networks”. “71st IEEE Vehicular Technology Conference (VTC 2010-Spring)”. Taipei, Taiwan, May 2010.
- [14] A. Wood; J. Stankovic. “Denial of Service in Sensor Networks”. “IEEE Computer Magazine”. vol 35. pages 54 – 62. IEEE Computer Society, October 2002.
- [15] P. Radmand; A. Talevski; S. Petersen; S. Carlsen. “Taxonomy of Wireless Sensor Network Cyber Security Attacks in the Oil and Gas Industries”. “24th IEEE International Conference on Advanced Information Networking and Applications (AINA)”. Perth, Australia, April 2010.
- [16] J. L. R. Souza; J. Rufino. “Characterization of Inaccessibility in Wireless Networks-A Case Study on IEEE 802.15.4 Standard”. “3th IFIP International Embedded Systems Symposium (IESS) - IFIP Advances in Information and Communication Technology”. vol. 310. pages 290-301. Springer, September 2009.
- [17] P. Veríssimo; J. Rufino; L. Rodrigues. “Enforcing Real-Time Behaviour on LAN-Based Protocols”. “10th IFAC Workshop on Distributed Computer Control Systems”. Semmering, Austria, September 1991.
- [18] P. Veríssimo; L. Rodrigues; M. Baptista. “AMp: A Highly Parallel Atomic Multicast Protocol”. “ACM Symposium on Communications Architectures & Protocols (SIGCOMM)”. Austin, USA, September 1989.
- [19] P. Veríssimo; J. A. Marques. “Reliable Broadcast for Fault-Tolerance on Local Computer Networks”. “9th Symposium on Reliable Distributed Systems”. Alabama, USA, October 1990.
- [20] H. Karl; A. Willig. “Protocols and Architectures for Wireless Sensor Networks”. ISBN-13 978-0-470-09510-2. John Wiley and Sons. Ltd. England, 2005.
- [21] J. L. R. Souza; J. Rufino. “An Approach to Enhance the Timeliness of Wireless Communications”. “5th International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM)”. Lisbon, Portugal, November 2011.
- [22] IEEE 802.15.4. “Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-rate Wireless Personal Area Networks (WPANs) -IEEE standard 802.15.4”. IEEE P802.15 Working Group. Revision of IEEE Standard 802.15.4-2006. New York, USA, June 2011.
- [23] J. L. R. Souza; A. Guerreiro; J. Rufino. “Characterizing Inaccessibility in IEEE 802.15.4 through Theoretical Models and Simulation Tools”. “4th Simpósio de Informática (INFORUM)”. Lisbon, Portugal, September 2012.

2.2 Self-* communication and synchronization primitives

During the first year, we have focused on the communication fundamentals that are related to medium access control. We show a way to increase the degree predictability and resilience of wireless embedded networks. As Figure 8 illustrates, the discussed components are fundamental for supporting the entire system. In this figure, we highlight in black and bold the technologies

on which we focused during the first year, namely algorithms for network medium access control and algorithms for increasing synchrony among nearby transmitting stations.

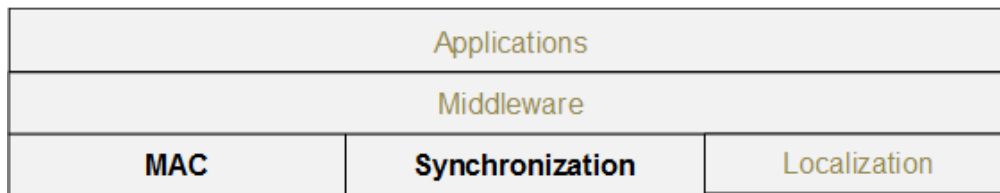


Figure 8: The system structure of the supporting technology perspective.

We focus on advance provision with respect to enforcing predictability and resilience in a relevant set of wireless network settings by suggesting our design for self-stabilizing MAC algorithms that provides a greater predictability degree than existing ones. The main results of this work are presented in Section 2.2.1, while the complete details are available in [LS12], provided in Annex A.1.5. In Section 2.2.2 we describe an algorithmic design for TDMA alignment, with further details provided in [Mus12, MPS12] and included in Annex A.1.6. Such algorithms are required for autonomous implementation of [LS12], i.e., without the use of external time sources, such as GPS. Finally, in Section 2.2.3 we look at protocols that directly use the MAC protocol, i.e., the data link layer and the end-to-end communications in dynamic networks. The results in Section 2.2.3 are detailed in [DHS12], which is provided in Annex A.1.7.

MAC protocols for VANETs need to be autonomous and robust as well as have high bandwidth utilization, high predictability degree of bandwidth allocation, and low communication delay in the presence of frequent topological changes to the communication network. We propose a self-stabilizing MAC algorithm that guarantees satisfying these severe timing requirements. Besides the contribution in the algorithmic front of research, we expect that our proposal can enable quicker adoption by practitioners and faster deployment of VANETs, such as the IEEE 802.11p.

The problem of local clock synchronization is studied in the context of TDMA protocols for dynamic and wireless ad hoc networks. In the context of TDMA, local pulse synchronization mechanisms let neighbouring nodes align the timing of their packet transmissions, and by that avoid transmission interferences between consecutive timeslots. Existing implementations for VANETs assume the availability of common (external) sources of time, such as base-stations or GPS time sources. We are the first to consider autonomic design criteria, which are imperative when no common time sources are available, or preferred not to be used, due to their cost and signal loss and use self-* pulse synchronization strategies. Their implementing algorithms consider the effects of communication delays and transmission interferences. We demonstrate the algorithms via extensive simulations in different settings including node mobility. We also validate these simulations in the MicaZ platform, whose native clocks are driven by inexpensive crystal oscillators. The results imply that the studied algorithms can facilitate autonomous TDMA protocols for VANETs.

End-to-end communication over the data link layer (or overlay networks) is one of the most important communication tasks in every communication network, including mobile ad hoc networks, and VANETs. We study data link layer and end-to-end algorithms that exchange packets to deliver (high level) messages in FIFO order without omissions or duplications. We present a self-stabilizing end-to-end algorithm that can be applied to networks of bounded capacity that omit, duplicate and reorder packets. The algorithm is network topology independent, and hence suitable for always changing dynamic networks with any churn rate.

2.2.1 Self-Stabilizing TDMA algorithms for Dynamic Wireless Ad-hoc Networks

We consider Medium Access Control (MAC) protocols for dynamic wireless ad-hoc networks that need to be autonomous, robust, and have high bandwidth utilization, a high predictability degree of band-width allocation, and low communication delay in the presence of frequent changes to the network topology. We propose an algorithmic design for self-stabilizing MAC protocols with a provable short convergence period, and by that, it can facilitate the satisfaction of severe timing requirements and possesses a greater predictability degree, while maintaining low communication delays and high throughput. We show that the algorithm facilitates the satisfaction of severe timing requirements for Dynamic wireless ad-hoc networks. We consider transient faults and topological changes to the communication network, and demonstrate self-stabilization.

Algorithm description

The MAC algorithm in Figure 9 **Error! Reference source not found.** assigns timeslots to nodes after the convergence period. The system consists of a set, P , of N anonymous communicating entities, which we call nodes. Denote every node $p_i \in P$ with a unique index, i . We assume that the MAC protocol is invoked periodically by synchronized common pulses that align the starting time of the TDMA frame.

```

Constants, variables, macros and external functions
2   $n$  : integer = maximal number of competition rounds
    $s$  :  $[0, T-1] \cup \{\perp\}$  = next timeslot to broadcast or null,  $\perp$ 
4   $signal$  : boolean = trying to acquiring the channel
    $unused[0, T-1]$  : boolean = marking unused timeslots
6   $unused\_set$  =  $\{ k : unused[k] = true \}$  : unused timeslot set

8  Upon timeslot( $t$ )
   if  $t = 0 \wedge s = \perp$  then  $s := uniform\_select(set)$ 
10 ( $unused[t], signal$ ) := (true, false) (* remove stale info. *)
   if  $s \neq \perp \wedge t = s$  then send(fetch new message from upper layer)
12
14  Upon receive(< DATA,  $m$  >) deliver <  $m$  > to upper layer

16  Function send( $m$ ) (* send message  $m$  to  $p_i$ 's neighbors *)
   for (( $signal, k$ ) := (true, 0);  $k := k + 1$ ;  $k \leq MaxRnd$ ) do
   if  $signal$  then with probability  $\rho(k) = 1/(MaxRnd - k)$  do
18    $signal := false$  (* quit the competition *)
   transmit(< BEACON >) (* try acquiring the channel *)
20   wait until the end of competition round (* exposure period alignment *)
   if  $s \neq \perp$  then transmit(< DATA,  $m$  >) (* send the data packet *)
22
24  Upon carrier_sense( $t$ ) (* defer transmission during  $t$  *)
   if  $s = t \wedge signal$  then  $s := \perp$  (* mark that the timeslot is not unique *)
   ( $signal, unused[t]$ ) := (false, false) (* quit the competition *)

```

Figure 9: Self-stabilizing TDMA-based MAC algorithm.

The term (broadcasting) timeslot refers to the period between two consecutive common pulses. In our pseudo-code, we use the event $timeslot(t)$ that is triggered by the common pulse. Nodes raise the event $carrier_sense()$ when they detect that the received energy levels have reached a threshold in which the radio unit is expected to succeed in carrier sense locking. We assume

that timeslots allow the transmission of DATA packets using the *transmit()* and *receive()* primitives. Moreover, we consider signalling (beacons) as short packets that include no data load; rather their carrier sense delivers important information. Before the transmission of the DATA packet in timeslot t , the scheme uses beacons for singling the node intention to transmit the packet within t .

During the convergence period several nodes can be assigned to the same timeslot. The algorithm solves such timeslot allocation conflicts by letting the node p_i and p_j to go through a (listening/signalling) competition before transmitting in its broadcasting timeslot. The competition rules require each node to choose one out of n listening/signalling period for its broadcasting timeslot.

This implies that among all the nodes that attempt to broadcast in the same timeslot, the ones that select the earliest listening/signalling period win this broadcasting timeslot and access the communication media. Before the winners access their timeslots, they signal to their neighbours that they won by sending beacons during their chosen signalling periods.

When a node receives a beacon, it does not transmit during that timeslot, because it lost this competition. Instead, it randomly selects another broadcasting timeslot and competes for it on the next broadcasting round.

Discussion

Thus far, MAC algorithms could not consider timing requirements within a provably short recovery period that follows (arbitrary) transient faults and network topology changes. This work proposes the first self-stabilizing TDMA algorithm for Dynamic wireless ad-hoc networks that has a provably short convergence period. Thus, the proposed algorithm possesses a greater degree of predictability, while maintaining low communication delays and high throughput.

$$\text{Convergence time: } k(n, N) = 1 + \frac{\log(1 - \sqrt[N]{1 - \alpha})}{\log\left(1 - \left(\frac{n-1}{2n}\right)^{\frac{d}{T}}\right)} \quad (1)$$

The analysis shows that when there are N nodes in the network and $\alpha \in (0, 1)$, the network convergence time is bounded by equation (1) with probability $1 - \alpha$, where d is the maximal node degree in the interference graph and T is the size of the TDMA frame. This means that with probability α all nodes are allocated with timeslots in maximum $k(n, N)$ broadcasting rounds (see Figure 10).

Figure 10 illustrates the contour plot of equation (1) for $s = d/T = 1$. The contour lines connect values of $k(n, N)$ that are the same (see the text tags along the line). When N nodes attempt to access the medium, the convergence time is stable in the presence of a growing number, n , of listening/signalling periods. This figure shows that when allowing merely a small fraction of the bandwidth to be spent on frame control information, say three listening/signalling periods, and when considering 99% probability to convergence within a couple of dozen TDMA frames, the proposed algorithm demonstrates a low dependency degree on the number of nodes in the network even when considering 10,000 nodes. Recently, we have implemented the proposed algorithm, extensively validated our analysis via computer simulation, and tested it on a platform with more than two dozen nodes. These results indeed validate that the proposed algorithm can indeed facilitate the implementation of MAC protocols that guarantee satisfying these severe timing requirements.

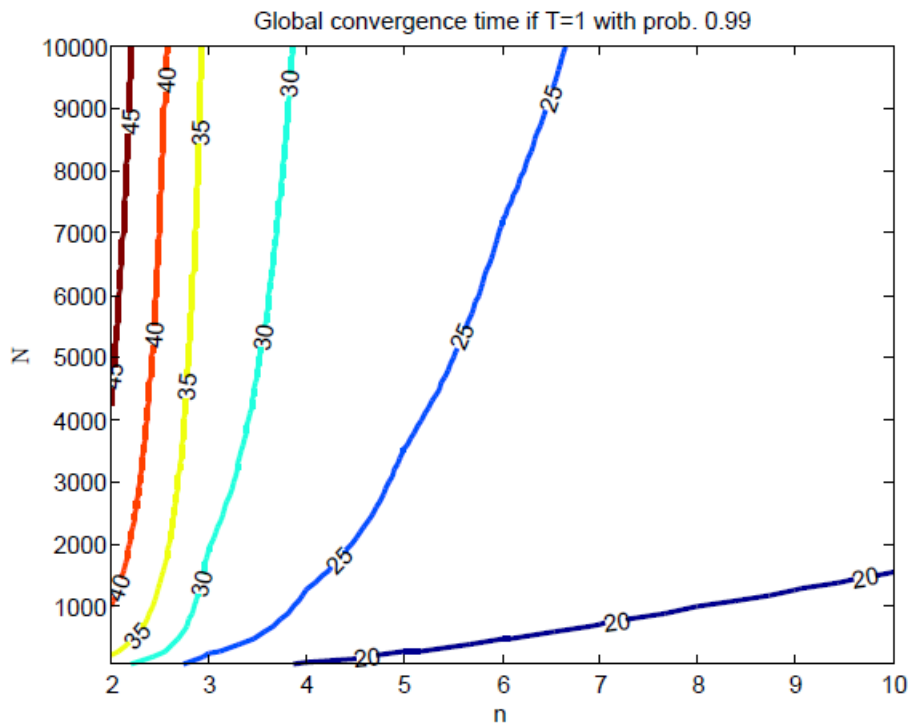


Figure 10: Contour plot of equation (1) for $s = d/T = 1$.

The costs associated with predictable communications, say, using base-stations, motivate the adoption of new networking technologies, such as MANETs and VANETs. In the context of these technologies, we expect that our proposal would contribute to the development of MAC protocols that can be used by applications that needs guarantees for severe timing requirements.

This work has been published in [LS12], which is provided in Annex A.1.5.

In annex: "Self-Stabilizing TDMA algorithms for Dynamic Wireless Ad-hoc Networks". Pierre Leone and Elad Michael Schiller. The 8th International Symposium on Algorithms for Sensor Systems, Wireless Ad Hoc Networks and Autonomous Mobile, 2012.

2.2.2 Self-Stabilizing TDMA Alignment Algorithms for Dynamic Wireless Ad-hoc Networks

Recent work on vehicular systems explores a promising future for vehicular communications. They consider innovative applications that reduce road fatalities, lead to greener transportation, and improve the driving experience, to name a few. The prospects of these applications depend on the existence of predictable communication infrastructure for dynamic networks. We consider TDMA protocols that can divide the radio time regularly and fairly in the presence of node mobility, such as Chameleon-MAC [LPSZ10]. The studied problem appears when neighbouring nodes start their broadcasting timeslots at different times. It is imperative to employ autonomous solutions for timeslot alignment when no common (external) time sources are available, or preferred not to be used, due to their cost and signal loss. We address the timeslot alignment problem by considering the more general problem of (decentralized) local pulse synchronization. Since TDMA alignment is required during the period in which communication links are being established, we consider non-deterministic communication delays, the effect of transmission interferences and local clocks with arbitrary initial offsets. We propose autonomous and self-* algorithmic solutions that guarantee robustness and provide an important level of abstraction as they liberate the system designer from dealing with low-level problems, such as availability and cost of common time sources. Our contribution also facilitates autonomous TDMA protocols for Vehicular Ad-Hoc Networks (VANETs).

Let us illustrate the problem and the challenges of possible strategies using an example. Consider three neighbouring stations that have unique timeslot assignment, but their timeslots are not well-aligned. This is illustrated in Figure 11, where solid and dashed lines stand for transmission and, respectively, idle radio times.

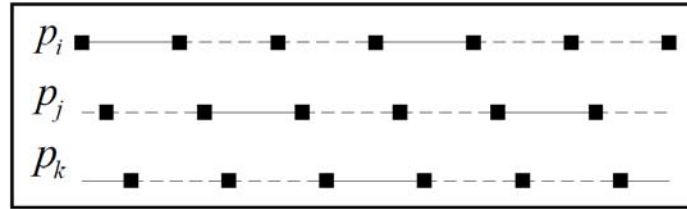


Figure 11: Unaligned TDMA timeslots.

Packet transmissions collide in the presence of such concurrent transmissions. Suppose that the stations act upon the intuition that gradual pairwise adjustments are most preferable. Station pk is the first to align itself with its closest neighbour, pj . This is observable in Figure 12, where solid and dashed lines stand for transmission and idle radio times (like in Figure 11) and where the circles above the solid boxes represent the node's view on its neighbors' TDMA alignment at the start of its broadcasting timeslot. Gaps between two solid boxes represent alignment events.

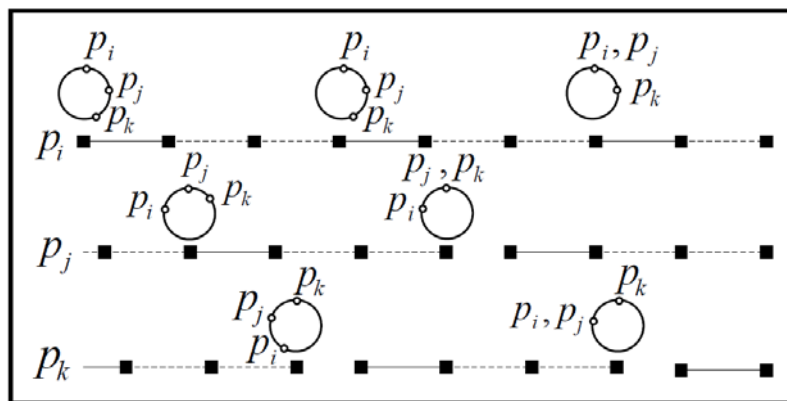


Figure 12: The cricket strategy.

Next, pj aligns itself with pi and by that it opens a gap between itself and pk . Then, pk aligns itself with pi and pj . The end result is an all aligned sequence of timeslots. We call this algorithmic approach the *cricket* strategy.

Observe that the convergence process includes chain reactions, i.e., node pk aligns itself before and after pj 's alignment. One can foresee the outcome of such chain reactions and let pj and pk to concurrently adjust their clock according to pi . This algorithmic approach, named the *grasshopper*, is faster than the cricket. This improvement comes at the cost of additional memory and processing requirements. We integrate the proposed algorithms with the Chameleon-MAC [LPSZ10], which is a self-*, mobility resilient, TDMA protocol. After extensive simulations with and without mobility, we observe tight alignment among the timeslots, and high MAC throughput. Additional testbed experiments appear in [Mus12].

Task definition

The system consists of a set, $N = \{pi\}$, of n anonymous communicating entities, which we call *nodes*. The radio time is divided into fixed size TDMA frames and then into fixed size timeslots [as in LPSZ10]. The nodes' task is to adjust their local clocks so that the starting time of frames and timeslots is aligned. They are to achieve this task in the presence of: (1) a MAC layer that is in the process of assigning timeslots, (2) network topology changes, and (3) message omission,

say, due to topological changes, transmission interferences, unexpected change of the ambient noise level, etc.

We consider the MAC layer of Section 2.2.1 and of [LS12]. Each node has hardware supported timer for generating (*periodic*) pulses every P (phase) time units. We denote the operations' time notation (timestamp) in the format (*timeslot, phase*), where $timeslot \in [0, T - 1]$ and $phase \in [0, P - 1]$ is the node timer's value between two pulses. We assume the existence of efficient mechanisms for timestamping packets at the MAC layer that are executed by the transmission operations, as in [HZ06]. We assume the existence of an efficient upper-bound, $\alpha \ll P$, on the communication delay between two neighbours that, in this work, has no characterized and known distribution.

The problem of (*decentralized*) *local pulse synchronization* considers the rapid reduction of all *local synchrony bounds* ψ between the phase values of any pair of nodes that can communicate directly. Given the synchrony bound $\psi \geq 0$, we look at the *convergence (rate bound)*, $f\psi$, which is the number of TDMA frames it takes to reach ψ .

Pulse Synchronization Strategies

Pulse synchronization solutions require many considerations, e.g., nondeterministic delays and transmission interferences. Before addressing the implementation details, we simplify the presentation by first presenting (*algorithmic*) strategies in which the nodes learn about their neighbours' clock values without delays and interferences.

We present two strategies that align the TDMA timeslots by calling the function $adjust(aim)$ immediately before their broadcasting timeslot. The first strategy, named *Cricket*, sets aim 's value according to neighbours that have the most similar phase values. The second strategy, named *Grasshopper*, looks into a greater set of neighbours before deciding on aim 's value. Both strategies are based on the relations among nodes' phase values, see the annex for details.

Experimental Evaluation

Computer simulations and the MicaZ platform are used for showing that: (1) both proposed algorithms achieve a small synchrony bound, and (2) the grasshopper, which has a higher resource consumption cost, converges faster than the cricket. In addition to extensive experimentation in system settings in which node mobility is not considered, we also consider Mobile Ad Hoc Networks by borrowing two mobility models from [LPZ10]. We observed that the grasshopper was able to show a shorter recovery time and a greater resiliency degree.

Discussions

The prospects of safety-critical vehicular systems depend on the existence of predictable communication protocols that divide the radio time regularly and fairly. This section presents autonomous and self-* algorithmic solutions for the problem of TDMA timeslot alignment by considering the more general problem of (*decentralized*) local pulse synchronization. The studied algorithms facilitate autonomous TDMA-based MAC protocols that are robust to transient faults, have high throughput and offer a greater predictability degree with respect to the transmission schedule. These properties are often absent from current MAC protocol implantations for VANETs, see [SUS11, BUSB08].

We saw that avoiding clock update dependencies can significantly speed up the convergence and recovery processes. In particular, the grasshopper algorithm foresees dependencies among the clock updates, which the cricket cannot. However, dependency avoidance requires additional resources.

Existing vehicular systems often assume the availability of common time sources, e.g., GPS. Autonomous systems cannot depend on GPS services, because they are not always available, or preferred not to be used, due to their cost. Arbitrarily long failure of signal loss can occur in underground parking lots and road tunnels. Moreover, some vehicular applications cannot

afford accurate clock oscillators that would allow them to maintain the required precision during these failure periods.

By demonstrating the studied algorithms on inexpensive MicaZ motes, we have opened up the door for *hybrid-autonomous* designs. Namely, nodes that have access to GPS, use this time source for aligning their TDMA timeslots, whereas nodes that have no access to GPS, use the studied strategies as dependable fallback for catching up with nodes that have access to GPS.

We expect applicability of the hybrid-autonomous design criteria to other areas of VANETs. E.g., spatial TDMA [SUS11] protocols base their timeslot allocation on GPS availability. As future work, we propose dealing with such dependencies by adopting the hybrid-autonomous design criteria.

This work has been published in [MPS12], which is provided in Annex A.1.6.

In annex: “Autonomous TDMA Alignment for VANETs”. M. Mustafa, M. Papatriantafidou, E. M. Schiller, A. Tohidi and P. Tsigas, in the proceeding of the IEEE 76th Vehicular Technology Conference (VTC'12-Fall), Quebec City, Canada, September 2012.

2.2.3 Self-Stabilizing End-to-End Communication in (Bounded Capacity, Omitting, Duplicating and non-FIFO) Dynamic Networks

End-to-end communication is a basic primitive in communication networks. A sender must transmit messages to a receiver in an exactly once fashion, where no omissions, duplications and reordering are allowed. Errors occur in transmitting packets among the network entities – one significant source of error is noise in the transmission media. Thus, error detection and error correcting techniques are employed as an integral part of the transmission in the communication network. These error detection and correction codes function with high probability. Still, when there is a large volume of communication sessions, the probability that an error will not be detected becomes high, leading to a possible malfunction of the communication algorithm. In fact, it can lead the algorithm to an arbitrary state from which the algorithm may never recover unless it is self-stabilizing [Dol00]. By using packets with enough distinct labels infinitely often, we present a self-stabilizing end-to-end communication algorithm that can be applied to dynamic networks of bounded capacity that omit, duplicate and reorder packets.

Contemporary communication and network technologies enhance the need for automatic recovery and interoperability of heterogeneous devices and the means of wired and wireless communications, as well as the churn associated with the totally dynamic communication networks. Having a self-stabilizing, predictable and robust basic end-to-end communication primitive for these dynamic networks facilitates the construction of high-level applications. Such applications are becoming extremely important nowadays where countries' main infrastructures, such as the electrical smart-grid, water supply networks and intelligent transportation, are based on cyber-systems. Defining the communication network as a bounded capacity network that allows omissions, duplications and reordering of packets and building (efficient) exactly once message transmission using packets, allows us to abstract away the exact network topology, dynamicity and churn.

The dynamic and difficult-to-predict nature of electrical smart-grid and intelligent transportation systems give rise to many fault-tolerance issues and require efficient solutions. Such networks are subject to transient faults due to hardware/software temporal malfunctions or short-lived violations of the assumed settings for the location and state of their nodes. Fault-tolerant systems that are self-stabilizing [Dol00, Dij74] can recover after the occurrence of transient faults, which can drive the system to an arbitrary system state. The system designers consider all configurations as possible configurations from which the system is started. The self-stabilization design criteria liberate the system designer from dealing with specific fault scenarios, the risk of neglecting some scenarios, and having to address each fault scenario separately.

Problem definition

We consider a distributed system that includes *nodes*, p_1, p_2, \dots, p_N . We represent a distributed system by a *communication graph* that may change over time, where each node is represented as a node. Two *neighbouring* nodes, p_i and p_j , that can exchange packets directly are connected by a link in the communication graph. Packet exchange between neighbours is carried via (directed) communication links, where packets are sent from p_i to p_j through the directed link (p_i, p_j) and packets are sent from p_j to p_i through (p_j, p_i) , the opposite directed link. End-to-end communication among non-neighbour nodes, p_s and p_r , is facilitated by packet relaying from one node to neighbours. Thus, establishing a (virtual) communication link between p_s and p_r in which p_s is the sender and p_r is the receiver. We assume the communication graph is dynamic, and is constantly changed, while respecting N as the upper bound on the number of nodes in the system. Packets are exchanged by the sender and the receiver in order to deliver (high level) messages in a reliable fashion. We assume that the entire number of packets in the system, at any given time, does not exceed a known bound. We allow any churn rate, assuming that joining nodes reset their own memory, and by that assist in respecting the assumed bounded packet capacity of the entire network.

The communication links are bidirectional. Namely, between every two nodes, p_i and p_j , that can exchange packets, there is a unidirectional *channel (set)* that transfers packets from p_i to p_j and another unidirectional channel that transfer packets from p_j to p_i . We model the (*communication*) *channel*, from node p_i to node p_j as a (non-FIFO order preserving) packet set that p_i has sent to p_j and p_j is about to receive. When p_i sends a packet m to p_j , the operation *send* inserts a copy of m to the channel from p_i to p_j as long as the upper bound of packets in the channel is respected. Once m arrives, p_j triggers the *receive* event and m is deleted from the set. The communication channel is non-FIFO and has no reliability guarantees. Thus, at any time the sent packets may be omitted, reordered, and duplicated, as long as the link capacity bound is not violated. We note that transient faults can bring the system to consist of arbitrary, and yet capacity bounded, channel sets from which convergence should start. We assume that when node p_i sends a packet, *pckt*, infinitely often through the communication link from p_i to p_j , p_j receives *pckt* infinitely often. We intentionally do not specify (the possible unreliable) routing scheme that is used to forward a packet from the sender to the receiver, e.g., flooding, shortest path routing. We assume that the overall network capacity allows a channel from p_i to p_j to contain at most *capacity* packets at any time, where *capacity* is a known constant. However, it should be noted that although the channel has a maximal capacity, packets in the channel may be duplicated infinitely many times because even if the channel is full, packets in the channel may be either lost or received. This leaves places for other packets to be (infinitely often) duplicated and received by p_j .

Self-stabilizing algorithms do not terminate (see [Dol00]). The non-termination property can be easily identified in the code of a self-stabilizing algorithm: the code is usually a do forever loop that contains communication operations with the neighbors.

The *self-stabilizing end-to-end communication* (S^2E^2C) algorithm provides FIFO guarantee for bounded networks that omit duplicate and reorder packets. Moreover, the algorithm considers arbitrary starting configurations and ensures error-free message delivery. In detail, given a system's execution R , and a pair, p_s and p_r , of sending and receiving nodes, we associate the message sequences $s_R = m_0, m_1, m_2, \dots$, of messages fetched by p_s , with the message sequence $r_R = m'_0, m'_1, m'_2, \dots$ of messages delivered by p_r . Note that we list messages according to the order they are fetched (from the higher level application) by the sender, thus two or more (consecutive or non-consecutive) messages can be identical. The S^2E^2C task requires that for every legal execution $R \in LE$, there is an infinite suffix, R' , in which infinitely many messages are delivered, and $s_{R'} = r_{R'}$. It should be noted that packets are not actually received by the receiver in their correct order but eventually it holds that messages are delivered by the receiver (to higher level application) in the right order.

The End-to-End Algorithm

Dynamic networks have to overcome a wide range of faults, such as message corruption and omission. It often happens that networking techniques, such as retransmissions and multi-path routing, which are used for increasing robustness, can cause undesirable behaviour, such as message duplications and reordering. We present a self-stabilizing end-to-end communication algorithm that uses the network's bounded capacity, to cope with packet corruptions, omissions, duplications, and reordering. We abstract the entire network by two directed channels, one from the sender to the receiver and one from the receiver to the sender, where each abstract channel is of a bounded capacity. These two abstract channels can omit, reorder and duplicate packets. We regard two nodes, ps , pr , as sender and receiver, respectively. Sender ps sends packets with distinct labels infinitely often until ps receives a sufficient amount of corresponding distinct acknowledgment labels from the receiver pr .

For the sake of readability, we start describing the algorithm using large overhead, before showing ways to dramatically reduce the overhead. The sender repeatedly sends each message m with a three state *alternating index*, which is either 0, 1 or 2. We choose to discuss, without the loss of generality, the case of a message with alternating index 0, where $(0, m)$ is repeatedly sent in $(2 \cdot \text{capacity} + 1)$ packet types. Each type uses a distinct label in the range 1 to twice the capacity plus 1. Namely, the types are: $(0, 1, m)$, $(0, 2, m)$, ..., $(0, 2 \cdot \text{capacity} + 1, m)$. The sender waits for an acknowledgment of the packet arrival for each of the $(2 \cdot \text{capacity} + 1)$ distinct labels, and an indication that the receiver delivered a message due to the arrival of $(\text{capacity} + 1)$ packets with alternating index 0. The receiver accumulates the arriving packets in an array of $(2 \cdot \text{capacity} + 1)$ entries, where each entry, j , stores the last arriving packet with distinct label j . Whenever the receiver finds that $(\text{capacity} + 1)$ recorded array entries share the same alternating index, for example 1, the receiver delivers the message m encapsulated in one in-coming packet recorded in the array – this packet has the alternating index of the majority of recorded packets; 1 in our example. Then, the receiver resets its array and starts accumulating packets again, until $(\text{capacity} + 1)$ recorded copies, with the same alternating index reappear. The receiver always remembers the last delivered alternating index, $ldai$, that caused the reset of its array, and does not deliver two successive messages with the same alternating index. Each packet (ai, lbl, m) that arrives to the receiver is acknowledged by $(lbl, ldai)$. The sender accumulates the arriving packet in an array of $(2 \cdot \text{capacity} + 1)$ entries and waits to receive a packet for each entry, and to have a value of $ldai$ that is equal to the alternating index the sender is currently using in the sent packets in at least $(\text{capacity} + 1)$ of the recorded packets. Once such a packet set arrives, the sender resets its array, fetches a new message, m' , to be delivered, and increments the alternating index by 1 modulo 3 for the transmission process of the next message, m' .

The correctness considers the fact that the receiver always acknowledges incoming packets, and hence the sender will infinitely often fetch messages. Following the first fetch of the sender, the receiver follows the sender's alternating index, records it in $ldai$, and acknowledges this fact. We consider an execution in which the sender changes the alternating index in to $x, x + 1, x + 2, x$ (all modulo 3). In this execution, the sender is acknowledged that the receiver changes $ldai$ to $x + 1$ and then to $x + 2$, while the sender does not send packets with alternating index x , thus, the last x delivery in the sequence must be due to fresh packets, packets sent after the packets with alternating index $x + 2$ were sent, and cause a delivery.

In the preceding text a simplified algorithm with a large overhead was presented – a more efficient algorithm is described in the following. The basic idea is to enlarge the arrays to have more than $n > (2 \cdot \text{capacity} + 1)$ recorded packets. Roughly speaking, in such a case the minority of the distinct label packets accumulated in the arrays are erroneous, i.e., packet copies that were accumulated in the network prior to the current fetch (maximum capacity).

The other $(n - \text{capacity})$ distinct label accumulated packets are correct. Thus, as we know the maximal amount of unrelated packets, we can manipulate the data so that the $n - \text{capacity}$

correct packets, each of length p_l will encode, by means of error correcting codes, p_l messages each of length m_l , a length slightly shorter than n . The sender fetches a window of p_l messages each of length m_l , where p_l is the maximal packet length beyond the header. The sender then uses error correcting codes so that a message of length m_l is coded by a word of length n , such that the encoded word can tolerate up to capacity erroneous bits. The p_l encoded messages of length n are then converted to n packets of length p_l in a way that the i th message out of the m_l fetched messages is encoded by the i th bits of all the n distinct packets that are about to be transmitted. So eventually, the first bit of all distinct labelled packets, ordered by their distinct labels, encode, with redundancy, the first message, and the second bit of all distinct labelled packets, encode, with redundancy, the second message, etc.

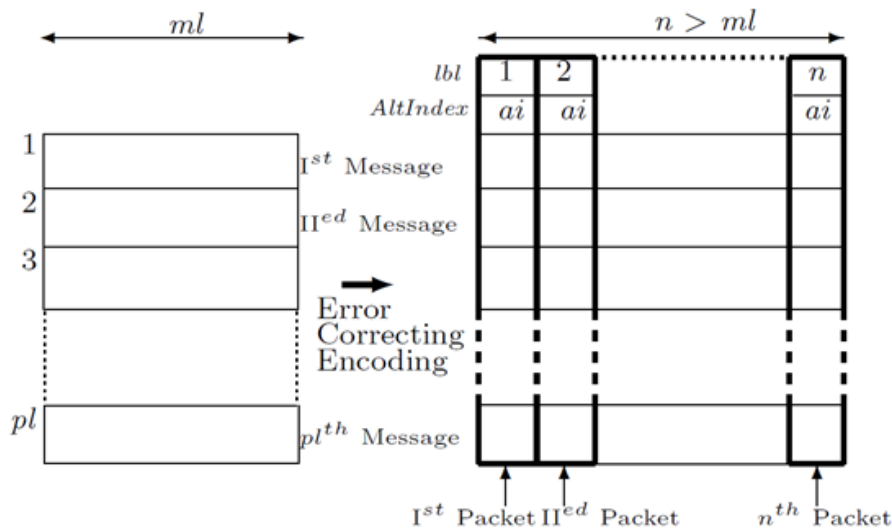


Figure 13: Packets formation from the messages in S2E2C algorithm.

Figure 13 shows the formation of the n packets from the p_l messages. When the receiver accumulates n distinct label packets, the *capacity* of the packets may be erroneous. However, since the i th packet, out of the n distinct packets, encodes the i th bits of all the p_l encoded messages, if the i th packet is erroneous, then the receiver can still decode the data of the original p_l messages each of length $m_l < n$. The i th bit in each encoded message may be wrong, in fact, capacity of packets maybe erroneous yielding capacity of bits that may be wrong in each encoded message, however, due to the error correction, all the original p_l messages of length m_l can be recovered, so the receiver can deliver the correct p_l messages in the correct order.

In this case, the sender repeatedly sends n distinct packets and the receiver keeps sending (*capacity* + 1) packets each with a distinct label in the range 1 to (*capacity* + 1). In addition, each of these packets contains the receiver's current value of *ldai*. The packets from the receiver are sent infinitely often, not necessarily as a response to its received packets. When the receiver accumulates n distinct label packets with the same alternating index, it recovers the original p_l messages, delivers them, resets its received packets array and changes its *ldai* to the alternating index of the packets that it just delivered. We note that these received packets must be different from its current *ldai* because the receiver does not accumulate packets if their alternating index is equal to its current *ldai*. The sender may continue sending the n packets with alternating index *ldai*, until the sender accumulates (*capacity* + 1) distinct label acknowledging packets with alternating index *ldai*. However, because now the packets' alternating index is equal to its current *ldai*, the receiver does not accumulate them, and hence does not deliver a duplicate. Once the sender accumulates (*capacity* + 1) packets with *ldai* equal to its alternating index, it will fetch p_l new messages, encode and convert them to n distinct label packets and increase its alternating index by 1 modulo 3.

The correctness arguments use the same facts mentioned above in the majority based algorithm. Eventually, we will reach an execution in which the sender fetches a new set of messages infinitely often and the receiver will deliver the messages fetched by the sender before the sender fetches the next set of messages. Eventually, every set of pl fetched messages is delivered exactly once because after delivery the receiver resets its packets record array and changes $ldai$ to be equal to the senders alternating index. The receiver stops accumulating packets from the sender until the sender fetches new messages and starts sending packets with a new alternating index. Between two delivery events of the receiver, the receiver will accumulate n distinct label packets of an identical alternating index, where $(n - capacity)$ of them must be fetched by the sender after the last delivery of messages by the receiver. The fact, which reflects such behaviour at the receiver node, is that the sender only fetches new messages after it gets $(capacity + 1)$ distinct packets with $ldai$ equal to its current alternating index. When the receiver holds n distinct label packets with maximum capacity erroneous packets, it can convert the packets back to the original messages by applying the error correction code capabilities and deliver the original message correctly.

Conclusions

Self-stabilizing end-to-end data communication algorithms for bounded capacity dynamic networks have been presented in this extended abstract. The proposed algorithms inculcate error correction techniques for the delivery of messages to their destination without omissions, duplications or reordering. We consider two nodes, one as the sender and the other as the receiver. In many cases, however, two communicating nodes may act both as senders and receivers simultaneously. In such situations, acknowledgment piggybacking may reduce the overhead needed to cope with the capacity irrelevant packets that exist in each direction, from the sender to the receiver and from the receiver to the sender. Using piggybacking, the overhead is similar in both directions. The obtained overhead is proportional to the ratio between the number of bits in the original message, and the number of bits in the coded message, which is a code that withstands capacity corruptions. Thus, for a specific capacity, assuming the usage of efficient encoding, the overhead becomes smaller as the message length grows.

This work has been published in [DHS12], which is provided in Annex A.1.7.

In annex: "Self-Stabilizing End-to-End Communication in Bounded Capacity, Omitting, Duplicating and Non-FIFO Dynamic Networks". S. Dolev, H. Ariel, E. M. Schiller and S. Sharma, 14th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS'12), Toronto, Canada, October 2012.

2.2.4 References

- [BUSB08] Katrin Bilstrup, Elisabeth Uhlemann, Erik G. Ström, and Urban Bilstrup. Evaluation of the IEEE 802.11p MAC method for vehicle-to-vehicle communication. In VTC Fall, pages 1–5. IEEE, 2008.
- [Dij74] Edsger W. Dijkstra. Self-stabilizing systems in spite of distributed control. Commun. ACM, 17(11):643–644, 1974.
- [Dol00] Shlomi Dolev. Self-Stabilization. MIT Press, 2000.
- [DHS12] Shlomi Dolev, Ariel Hanemann, Elad M. Schiller, and Shantanu Sharma Self-Stabilizing End-to-End Communication in (Bounded Capacity, Omitting, Duplicating and non-FIFO) Dynamic Networks to appear in the 14th International Symposium Stabilization, Safety, and Security of Distributed Systems (SSS'12).
- [HZ06] Ted Herman and Chen Zhang. Best paper: Stabilizing clock synchronization for wireless sensor networks. In Stabilization, Safety, and Security of Distributed Systems, pages 335–349, 2006.

-
- [LLW10] Christoph Lenzen, Thomas Locher, and Roger Wattenhofer. Tight bounds for clock synchronization. *J. ACM*, 57(2), 2010.
- [LLWC03] Philip Levis, Nelson Lee, Matt Welsh, and David E. Culler. TOSSIM: accurate and scalable simulation of entire TinyOS applications. In *ACM SenSys*, pages 126–137, 2003.
- [LPSZ10] Pierre Leone, Marina Papatriantafidou, Elad Michael Schiller, and Gongxi Zhu. Chameleon-mac: Adaptive and self-* algorithms for media access control in mobile ad hoc networks. In *Stabilization, Safety, and Security of Distributed Systems*, pages 468–488, 2010.
- [LS12] Pierre Leone and Elad Michael Schiller. Self-Stabilizing TDMA algorithms for Dynamic Wireless Ad-hoc Networks The 8th International Symposium on Algorithms for Sensor Systems, Wireless Ad Hoc Networks and Autonomous Mobile, 2012.
- [Mus12] Mohamed Hassan Mustafa. Self-* pulse synchronization for autonomous TDMA MAC in VANETs. Master's thesis, CSE, Chalmers Univ. of Tech., 2012.
- [MPS12] Mohamed Hassan Mustafa, Marina Papatriantafidou, Elad M. Schiller, Amir Tohidi, and Philippos Tsigas, Autonomous TDMA alignment for VANETs. In *IEEE 76th Vehicular Technology Conference (VTC'12-Fall)*, 2012.
- [SUS11] Katrin Sjöberg, Elisabeth Uhlemann, and Erik G. Ström. Delay and interference comparison of CSMA and self-organizing TDMA when used in VANETs. In *IEEE Wireless Communications and Mobile Computing Conference*, pages 1488–1493, 2011.

3. Adaptive Middleware for Advanced Control Systems

Cooperating vehicles form a system-of-systems incorporating a large collection of heterogeneous hardware components, diverse operating systems, and embedded and general purpose networks, reaching from local interconnection busses like LIN, CAN, SafeBus or FlexRay to wireless communication media. Additionally, applications may be programmed in different programming languages ranging from C to domain-specific languages like Matlab/Simulink. It is obvious that this creates a problem during system integration and a harder problem when the systems need to cooperate spontaneously. One way to master the problem is defining abstract component and communication models that hide the various levels of heterogeneity and putting respective interfaces and communication mechanisms in a middleware layer. As an example from the automotive industry, the AUTOSAR standard [1] has been introduced as a middleware to manage the problems of interoperability in a single car. OEMs may define their hardware and software components against the AUTOSAR Runtime Environment (RTE) that decouples the application from the details of the underlying heterogeneous hardware and system software. This is the basis for re-usable and easy to integrate application modules. The middleware for KARYON has to accommodate dynamic cooperation in a mobile application scenario. This puts additional needs on the middleware services. Dynamic cooperation means that data is spontaneously used, which results in additional information describing how to deal with the data. Secondly, the quality of network connections may change that requires dynamic adaptation to new conditions. Thirdly, interactions are not a priori known. This requires dynamic discovery and assessment of available information and robust coordination.

The main objectives and properties of middleware therefore are:

- Hiding the heterogeneity of the underlying system infrastructure to shield the programmer from the details of hardware, operating systems and communication networks;
- Providing convenient abstractions that ease the programming of complex applications. The abstractions are encapsulated in a set of services relevant for a specific application domain offering a uniform Application Programming Interface (API);
- Supporting programming of applications by factoring out common, frequently used services and putting them into the framework of the middleware;
- Assessing the state of the network and the system dynamically as a basis for adaptation.

Some of the basic functionalities that are required for any such system are a seamless communication, easy access to and assessment of sensor data and support for interpreting these sensor data in the context of the environment. While seamless communication is one of the typical objectives of general-purpose middleware, environment perception middleware is more specific and less common. Nevertheless moving from a static sensor setting with an a priori known set of sensors and actuators to a dynamic, ad-hoc and spontaneously used set of perception and actuation components requires a radical change of perspective and services. Central to this kind of system is adaptation.

Adaptation comes in many ways and on many levels. On the communication layer, adaptation may need to deal with uncertainty of network connection and the changing set of communicating entities when supporting mobility. Respective analysis and techniques for monitoring and assessing the network status were already described in Section 2. These will provide resilience and predictability. The adaptation to the actual network state is further described in Section 3.1, about lightweight and dependable adaptation for wireless sensor networks below.

On the level of distributed sensing, the problems arise from the spontaneous use of sensor data that is not known a priori but is discovered dynamically. Related to this is the assessment of the quality of this sensor data coming from previously unknown sources. Fusing very unreliable sensor data may corrupt the entire perception quality. Thus we identified and factored out important functionality to assess, select and fuse sensor data in a distributed perception process and provide it as respective services in a dedicated middleware layer. Adaptation here means dealing with changing sets of sensors and a way to handle value failures, delays and omissions. Exploiting the investigations that are described in Deliverable D2.2., this middleware layer will encapsulate the required redundancy and analysis components in a programming abstraction and services supported by the respective middleware layer.

Finally, using directed sensor data made available by another vehicle or the infrastructure like radar, laser distance meters or, for smaller distances, ultrasound and infrared components, needs information about the position and the direction to which these sensors are "looking to". Interpreting this sensor information requires at the first place a geometric model of the environment. The data of a remote distance sensor mounted on a mobile entity cannot be interpreted and exploited without the geometric context in which the sensor is situated. We call this a situated sensor. The adaptive middleware will provide a perception layer which offers such an environment model and encapsulate it in a set of well defined services needed to support the concept of a situated sensor. Further the environment modelling layer prepares sensor information in an application dependent way. It generates views that are well described and can be used by the application to access the needed aspects of the environment. The conceptual work on the environment models again is elaborated in WP2. A rough sketch of the middleware layers is provided in Figure 14 below.

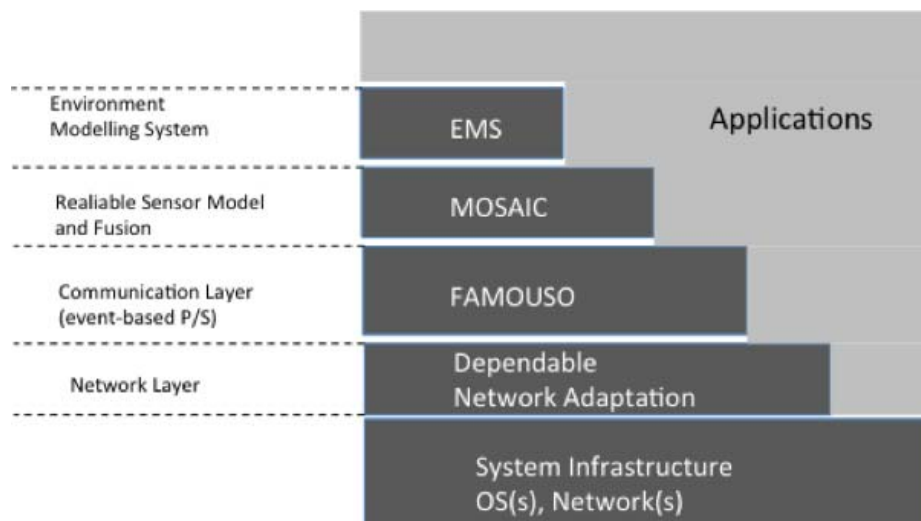


Figure 14: Layers of middleware.

The network monitoring and lightweight adaptation is described in Section 3.1. The adaptation will be exploited for the guarantees provided by the event channel concept of FAMOUSO.

For FAMOUSO we will mainly describe background work and the main properties of the event-based approach. Additionally, we will describe how to exploit this layer in a simulation environment and for mixed reality, i.e. the interaction of simulated and real system components.

The “Reliable Sensor Model and Fusion” layer implemented by MOSAIC constitutes a middleware layer that supports abstract sensors. It provides a programming model and an internal sensor structure to build reliable sensor systems. MOSAIC has a strong relation to WP2 Task 2.2. It constitutes the realisation of what has been elaborated there. MOSAIC supports composition and fusion of sensor information, considering the validity of individual sensors.

The GEMS layer (geometric environment modelling system) deals with the geometric modelling of the environment needed to interpret (directed and location dependent) sensor data correctly and deriving a geometric model of the environment. This has a direct relation to T2.3 in WP2. Because T2.3 only started at month 7, only basic ideas on services and abstractions will be presented in this report.

3.1 Lightweight Dependable Adaptation for Wireless Sensor Networks

In this section we describe a technique for dependable monitoring of, and adaptation to, uncertain and varying network conditions. This technique has been designed to be lightweight, which allows its implementation in resource constrained devices, such as those used to create Wireless Sensor Networks and Smart Environments, which vehicles can exploit for an augmented awareness of their operational environments and, therefore, improve their performance.

Due to the assumed open nature of the environment and the characteristics of the wireless network, in the considered environments it has generally not been possible to offer real-time inter-vehicle timeliness communication guarantees in a meaningful way. Existing work has mostly focused on improving some metrics of timeliness, but has never allowed the wireless network to provide guarantees comparable with those of real-time buses. In this context, the KARYON project takes a different (and complementary) approach: the network is seen as an abstract resource with no predictability guarantees. By being aware of, and continually adapting to, the environment conditions we can maintain high functional levels, despite the uncertainties.

We present a statistical technique that allows an awareness of the network conditions in such a way that applications and services can reason in terms of probabilities of timeliness. Using such technique, vehicles can adapt their behaviour to provide the best possible performance, without sacrificing safety.

For instance, in virtual traffic light scenario considered in Deliverable D1.1, motor vehicles have to adapt their level of service according to their awareness of the state of the intersection, which will depend on the availability, timeliness and accuracy of remote sensor information. By being aware of the probability of such sensor information being made available to the vehicle within certain deadlines, the vehicle can optimize the transitions between the levels of service, and therefore improve the performance of intersection crossing, without sacrificing safety.

3.1.1 Overview

Ever more, we are dealing with environments where it is not possible to define a priori guarantees for the quality of network communication. As an alternative, we have to adapt the system behaviour to the encountered and varying environment network conditions. Such adaptation should be done in a way that does not jeopardize system assumptions and safety, but that at the same time should make the best use of the available resources. This implies both an efficient use of scarce resources (e.g. wireless medium use, energy consumption, etc.) and the maximization of application performance and functional levels for a given amount of available and used resources. We provide a technique that allows the assessment of network conditions and which, therefore, allows this optimization and trade-off to be made.

The provided technique considers the operational environment as a stochastic process. That is, the state of the environment, and in particular the network, changes over time but at a given moment its properties are defined by some probability distribution. By considering that this state changes slowly over time (relative to our ability to recognize such state, an assumption we tested), we are able to derive what the probably distribution for the current state is. In particular,

we applied this technique to a characterization of the end-to-end latency of various scenarios and showed that the technique is effective: it allowed systems to maintain an adequate probability of meeting their deadlines with an efficient trade-off for performance (in our case, the time required to wait to receive the expected information).

3.1.2 Technique

We assume that participating systems are able to compute network latencies. Specifically, we consider that this will likely be achieved through the use of synchronized clocks (e.g. global time), and that messages are stamped with their send time.

Every time a message is received, we use the message timestamp to compute the end-to-end latency and add that latency to a FIFO list of values. When the number of values exceeds a maximum number n we discard the oldest value. This sliding window of network latencies is our unordered sample.

An ordered sample is created from the unordered one, either by sorting a copy of the sliding window or by using one of the alternatives discussed in Annex A.2.1, which have different time and space complexities. From this ordered sample we choose the value which best predicts the optimum deadline – i.e. the one which is respected with the desired average probability, and therefore optimizes the timeliness/performance trade-off.

The value is chosen from the sample using nonparametric order statistics, as described in Annex A.2.1. This approach is simple and well performing, yet highly effective. We estimate that even a very basic microcontroller is able to perform over 8000 of such bound estimations per second. Being both efficient and effective, this technique is able to scale from simple sensor nodes to more powerful systems, such as those that might be found in vehicles or supporting infrastructure.

3.1.3 Evaluation

We evaluated this technique by applying it to a variety of network traces, from very different scenarios. We simulated the adaptation process for a range of different target timeliness probabilities and determined the effective probability of deadlines being fulfilled. The evaluation allowed us to take several key conclusions.

We concluded that, even only by itself, the technique is effective in approximating the desired timeliness probability. Given the variety of network scenarios tested, this means we can be confident that our assumption of a slow-changing stochastic process easily holds in realistic networks, and that we can achieve adaptations very close to the best timeliness/performance trade-off.

We also determined that, by using larger sample sizes, we were able to achieve high probabilities of timeliness (over 99%), which could have been a weakness of the proposed technique, due to the limitations of nonparametric order statistics.

3.1.4 Conclusion

The proposed technique can effectively support a continuous adaptation to varying network conditions, which will allow achieving high functional levels, despite the uncertainties of the operating environment and without jeopardizing safety.

A detailed description of this work is provided in Annex A.2.1.

In annex: “Lightweight Dependable Adaptation for Wireless Sensor Networks”. L. Marques and A. Casimiro, Technical Report DI/FCUL, September 2012, Lisbon, Portugal.

3.2 Communication middleware

The basic concepts of the FAMOUSO communication middleware have been developed in the context of the IST project CORTEX (CO-operating Real-time senTient objects: architecture and EXperimental evaluation, IST-2000-26031) [2] as COSMIC middleware [3]. It has been considerably improved and extended with respect to adaptability and configurability during all stages of the development process and the support for heterogeneous systems and programming languages. Key to this adaptability are machine exploitable descriptions of the service requirements and the provided functionality and performance of the underlying processors and the communication system. Because many papers have been published about COSMIC and FAMOUSO [4], [5], we will here only briefly sketch the main achievements important for KARYON.

In KARYON we aim for predictable and safe coordination of smart vehicles. This requires a spontaneous communication system in which communication end-points may dynamically need to dynamically use information from other vehicles or from the available infrastructure. Another challenging property results from the system-of-systems property in a KARYON scenario. This means that we have to deal with heterogeneous networks concerning data formats and addressing schemes. With some effort, heterogeneity can be made transparent by the respective middleware abstractions. However, Quality of Service (QoS) will be a problem, because latencies, Jitter and other inherent uncertainties cannot be removed easily.

The publish/subscribe communication model is well known to support spontaneous, many to many communication relations and reflect autonomy of communicating entities [6], [7], [8]. This approach prevents control flow dependencies between the communication participants. However, QoS is a major problem. In a dynamic communication scenario the quality of service will change over time and requirements need to be checked dynamically whenever the communication link is established and during run-time. However, AUTOSAR [1] does not address spontaneous communication across system boundaries. AUTOSAR enables communication by using the publish/subscribe interface in a local system where QoS issues can be solved statically.

FAMOUSO provides event-based communication that is explicitly designed for dynamic, distributed control. We propose the concept of event channels that address the problem of assessing and maintaining QoS in such a cooperative system. Figure 15 sketches the channel concept.

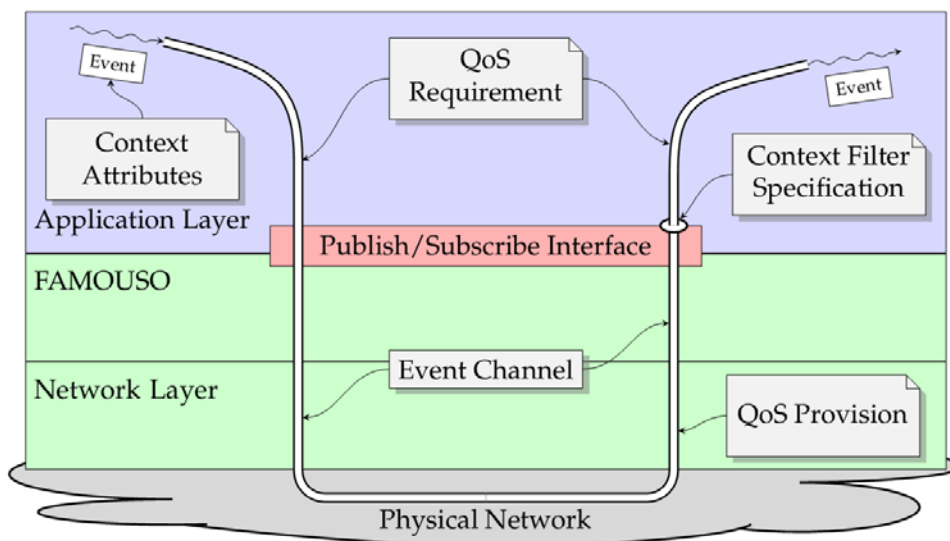


Figure 15: Channel concept implemented in FAMOUSO.

In FAMOUSO all disseminated information is encapsulated in typed message objects called *events*. An event is composed from three parts:

- a *subject*,
- *attributes*, and
- *content*.

A *subject* identifies to the content of an event and is represented by a unique identifier (UID). The UIDs span a global name space across all networks. Subjects are used to route an event to the interested subscribers. The binding between a publisher and a subscriber is performed dynamically over network boundaries.

Attributes specify quality requirements and the context of an event. Quality attributes provide information like timeliness and dependability parameters. Context attributes supply information like location or time. As illustrated in Figure 15, the publisher may add a context attribute to an event. The subscriber may specify a set of context attributes by using the context filter specification. The subscriber will only get those events which pass the context filter. As an example, a subscriber is interested in events from a specific location.

An event channel provides a unidirectional communication channel connecting multiple publishers to multiple subscribers. Before a publisher can disseminate an event, it has to announce the respective event channel that is identified by the subject of events that will be disseminated. The notion of an event channel allows specifying and enforcing QoS attributes. The publisher may specify the QoS that is needed, e.g. a maximal latency, a bandwidth, a rate of events or a delivery guarantee. It is obvious, that in a static system these requirements can be checked against what the network is able to provide e.g. at configuration time, i.e. before the system is actually in operation. In a system-of-systems in which spontaneous communication is needed, the information about the underlying network properties have to be acquired dynamically during run-time. Nevertheless any guarantee involves some assessment and subsequent resource reservation before communication can start. The dynamic assessment of the underlying network properties is part of the announcement process when a publisher creates a new event channel. The monitoring and dynamic adaptation concepts of wireless networks as described in this document acquire the knowledge that is needed to check whether the requirements match what the networks can provide (e.g. the inaccessibility analysis which is described in Section 2.1).

A key concept for providing QoS for spontaneous and dynamic established event channels are descriptions about the application requirements and the network provisions that can be interpreted by the system. FAMOUSO supports these cross checks by its Multi-Level Composability Check Architecture (MLCCA [9]), an integrated component that detects a misconfiguration or an application demand that cannot be achieved. After accepting the QoS configuration, events can be transferred through an event channel which then ensures the required dissemination quality.

One substantial benefit of the uniform interface to communication has been found when using FAMOUSO in mixed reality systems. Before putting control systems to operation it is common to model the crucial parts in a domain-specific language and simulate the behaviour. As a standard for many areas Matlab/Simulink is used. Often it is of particular interest to have hardware-in-the-loop configurations in which part of the system is real target hardware while some of the system parts are simulated. Communication support for such applications is very poor in Matlab/Simulink and does not exceed mere TCP/IP connections. This means that substantial adaptation efforts (going deep into the component implementation) must be made when changing the communication configuration.

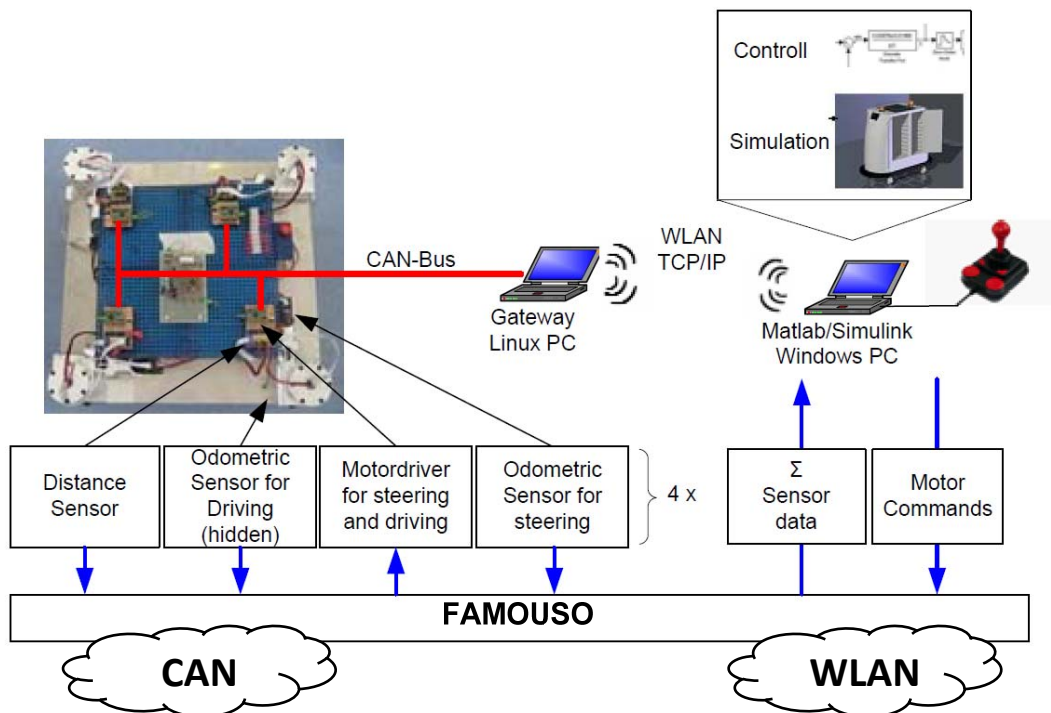


Figure 16: Cooperative vehicle development platform.

An example is shown in Figure 16. Here the capabilities of FAMOUSO are demonstrated in a practical communication scenario. The figure shows an autonomous vehicle that is controlled by simulating the control algorithm on an external PC. This vehicle is equipped with four independent drives that can speed and turn each wheel independently. This enables very complex movement pattern. It is by far more convenient and safe to develop these mobility patterns in virtual reality rather than directly on a physical vehicle. Therefore, a model of the vehicle is simulated on the external PC. This simulation is controlled by real sensor information from the physical vehicle. After a simulation loop is performed, the drive commands are transferred to the four drives of the vehicle. Here the advantages of using the uniform interface to the network become apparent. We interpreted the event channel concept for Matlab/Simulink. The functional blocks of Simulink can communicate via the event interface using the network independent subject addressing scheme. This example demonstrates two advantages. Firstly, it demonstrates the diversity of network media that can be managed by FAMOUSO. To establish a communication link between the vehicle and the external PC, the underlying network is based on WLAN and TCP/IP. The application only needs to know subjects to communicate and is not forced to implement details of the network communication. This is particularly helpful if networks will change. When moving from the simulation to the implemented functions, the second advantage becomes noticeable. When changing the configuration, e.g. the control algorithm is implemented in the local system and simulation is not needed any further there will be no need to change the communication. The event channels established between the sensors, the actuators and the control block will remain. What will change is the underlying network. These changes will not affect the application but will be handled automatically by the middleware layers.

FAMOUSO is able to support a broad variety of different hardware platforms ranging from low-end 8-Bit micro-controllers up to high-end 64-Bit server systems and enables interaction over different communication media like the CAN field-bus [3], Wireless Sensor Networks like IEEE 802.15.4, Wireless Mesh Networks [10] and Ethernet like UDP broad- and multicast. FAMOUSO can be used from different programming languages (C/C++, Python, Java, .NET) as well as from engineering tools (LabVIEW, MATLAB/Simulink) simultaneously [4]. A paper describing the background work on FAMOUSO is allocated in the Annex A.2.2.

In Annex: "Programming abstractions and middleware for building control systems as networks of smart sensors and actuators". Sebastian Zug, Michael Schulze, Andre Dietrich, Joerg Kaiser, September 2010, ETFA 2010 - 15th IEEE International Conference on Emerging Technologies and Factory Automation, Bilbao, Spain

3.3 Sensor middleware

An application coping with the interpretation of dynamically accessed sensor data requires an appropriate abstraction of the exchanged information. The middleware FAMOUSO defines the basic abstractions for a seamless communication. MOSAIC exploits this communication layer and specifies the smart sensor components. The MOSAIC framework thus extends the concepts of monitored and self-describing Smart Sensors (described for instance in [11], [12] or [13]). MOSAIC was inspired by concepts as the Abstract Sensor of Marzullo [14] and the Instrumented Logical Sensors of Henderson [15]. However, MOSAIC extends these approaches for a more dynamic and spontaneous use of sensor information. Particularly MOSAIC:

1. Offers a uniform interface to the rich set of diverse sensors.
2. Provides a machine interpretable description of data formats, physical units and other properties of a sensor. Additionally it includes context information related to spatial and temporal conditions as well as the associated uncertainties.
3. Integrates a validity estimate based on a well-defined failure semantics that enables a distributed monitoring and a validation of the environment perceptions. It should be noted that this validity estimate is the outcome of a failure detection process that is provided in an abstract sensor. To achieve a well-defined failure semantics, the respective checking components have to be added to the nominal sensor.
4. Provides an adaptive processing chain, ready to interpret, validate, filter and process sensor information.

These properties are needed to establish an adaptive processing that is able to select the best fitting sensor data set for a specific task. The main ideas of MOSAIC were published in [5], [16] and [17]. In the following sections we summarize the core ideas, approaches and concepts using a multi-modal sensing scenario.

The correct interpretation of a data set requires a large number of attributes in addition to the actual (measurement) value. During the development of MOSAIC we analyzed the required information and determine a set of mandatory parameters needed for further processing. Information about a sensing component that that does not change over time as uncertainty of data, a sampling interval, the respective physical units and failure detection methods are comprised in a data structure that is kept with the sensor.

We call it a sensor data sheet. The dynamic information like validity estimation, timestamp and uncertainty assessment is assigned to every event as an attribute. The sensor data sheet uses XML syntax to define the sensor properties. The description format adapts existing standards like SensorML [18] or TEDS [19] and enhances them by additional information. One important innovation was the abstract description that is particularly important for directed sensor like infrared or laser beams in MOSAIC. The geometrical representation is necessary to evaluate the respective monitoring area and relevance of a data set. Additionally, the node data sheets refer to the description of the assigned FAMOUSO communication channels responsible for the transmission of the related events. This format hides the individual low-level characteristics of a sensor and its communication but offers all binding information needed in dynamic scenarios.

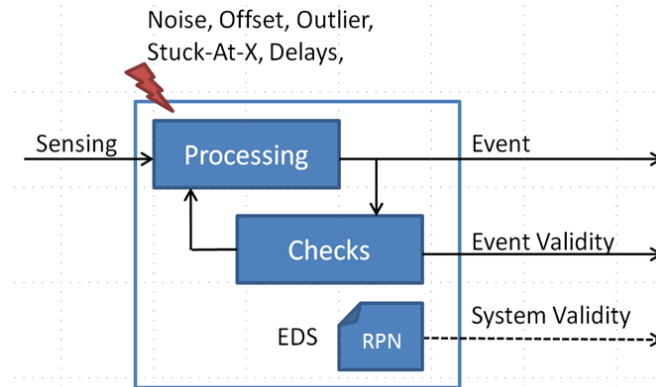


Figure 17: Fault tolerant sensor node providing an abstract output interface that includes a system and event validity value.

The validation of a data set is particularly important in dynamically changing applications. It requires an elaborate evaluation concept that is able to cope with the diverse sensors with different qualities, unknown environment conditions and network delays. As described in [20] and [21] we presented a failure semantics encapsulating the individual low-level failures and providing a uniform interface for dealing with sensor failures. The failure semantics quantifies the validity of an event and integrates two aspects, event validity and system validity. The event validity quantifies the quality level of a certain data set. It is represented by a single value combining the output of local failure checks. As depicted in it is assigned to each event. This approach is described in different variants e.g. [12] [13]. In an adaptive application the validity value is not sufficient without the knowledge about the sensor system and its processing chain. For information coming from different remote sources, an additional indicator is needed that describes the suitability and precision of the local detection methods related to the expected failure modes of the connected sensor. We call this system validity. Figure 17 illustrates the enhanced concept that integrates a system validity summarizing the fault detection capabilities of the node. The system validity of a component is defined statically and is stored in the electronic data sheet of the sensor. For representing system validity in a heterogeneous, multi-modal sensor system in a uniform way, we adapted the Failure Modes and Effects Analysis (FMEA) commonly used in automotive contexts [22] as further described in D2.2 [21]. The fundamental parameters considered are the amplitude of a failure (A), its occurrence probability (O) and on the probability of a correct detection (D). The main idea is to map each individual failure parameter {A,O,D} on an scale between 1 and 10 and multiply the individual estimations for deriving the system validity. In FMEA this product is known as Risk Priority Number (RPN).

The integration in the data processing chain is illustrated in Figure 18. It is divided into 5 steps *Allocation, Interpretation, Validation, Adaptive Fusion* and *Transmission*. The first stage is responsible for collecting and preparing sensor data. It provides a large number of possibilities to access local or remote sensor information. This stage even allows the transparent use of simulated sensors. Thus a sensor and a simulator interface may collect information from (real or simulated) local sensors. Likewise, a network interface may receive events from remote sensors. Network interfaces can be divided into two subclasses. The first one provides a direct access on all events belonging to a specific class of sensors. The second variant of a network interface offers adaptive data aggregation. In the allocation step (see Figure 18) the network filters relevant events from the stream of events disseminated via the communication system. Subsequently, the sensor, simulation or network interfaces support individual filter and aggregation methods, which can be configured by the user. At the outcome of the allocation stage, each interface type presents all received events in a uniform way. Thus, collecting and filtering is independent and separated from further processing steps.

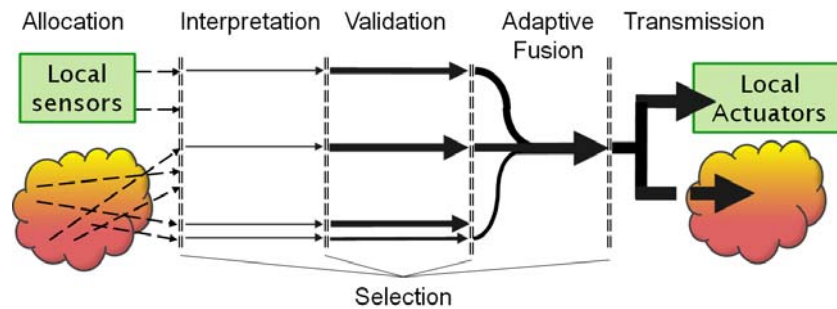


Figure 18: Abstract representation of the 5-step processing and selection chain inside a MOSAIC node.

The second stage – *Interpretation* – takes the received events and evaluates their individual relevance. For instance, for a localization scenario the filter exploits the knowledge about sensor and vehicle positions as well as the information about the monitoring area of the sensor and decides about the state of the related events. Another interpreter may filter information according to the uncertainty level or the age of an event.

The third selection step – *Validation* – evaluates the quality of a data set based on the developed failure semantics. Figure 18 illustrates the resulting weight of information by differently bold arrows. Related to the requirements of the following applications a large range of selection mechanisms is possible.

The *Adaptive Fusion* level represents a generic internal structure of the processing chain provided by MOSAIC. It defines the necessary building blocks for a reliable sensor and includes interfaces that provide:

- Different access methods to the processed sensor information;
- A failure estimate based on the individual evaluation of the processing modules;
- Access on the transmission level that disseminates the results either directly to local actuators or via FAMOUSO middleware.

The user may customize these interfaces (with individual filter rules, aggregation methods, etc.) and to add the individual processing modules as fusion algorithms, filters or failure detectors.

Figure 19 illustrates the implementation of the modular structure in a MOSAIC component.

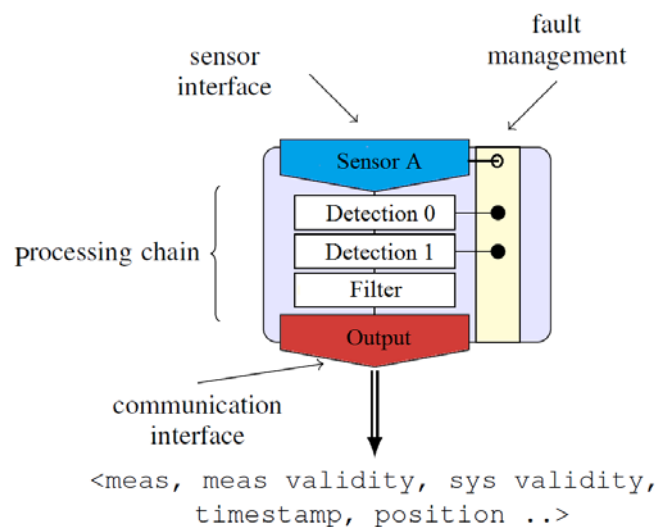


Figure 19: Modular structure of a MOSAIC sensor node that implements the concepts of Figure 18.

The sensor interface (Sensor A) has to fulfill three tasks. Firstly, the sensor interface controls the sensor element, configures the transducer and collects sensor data. The second task is to transform all specific sensor data into events as described in the sensor data sheet. It is important to realize that the sensor interface can be a real sensor, a network or a simulator interface. The results of the processing chain are validated by the fault management system. The user defined validation algorithm evaluates the individual validity estimation of the detection modules and calculates a general validity. The validity, timestamps, localization information and other relevant attributes are delivered to the output interface which is responsible to marshal the results into events and to check their consistency related to the node data sheet.

Due to the common abstraction of all data sets it is possible to combine information from different perception devices in an automated way. Consequently, we were able adapting established filters like Kalman or Bayes on scenarios with a varying number of sensors. Traditionally these algorithms are defined on design-time for a static sensor configuration. A description of the extensible and modular concept of the MOSAIC processing chain is given in [17] and [23].

Complex distributed application can be implemented by combining different MOSAIC nodes. Figure 20 shows the distance control of an autonomous vehicle that follows a wall.

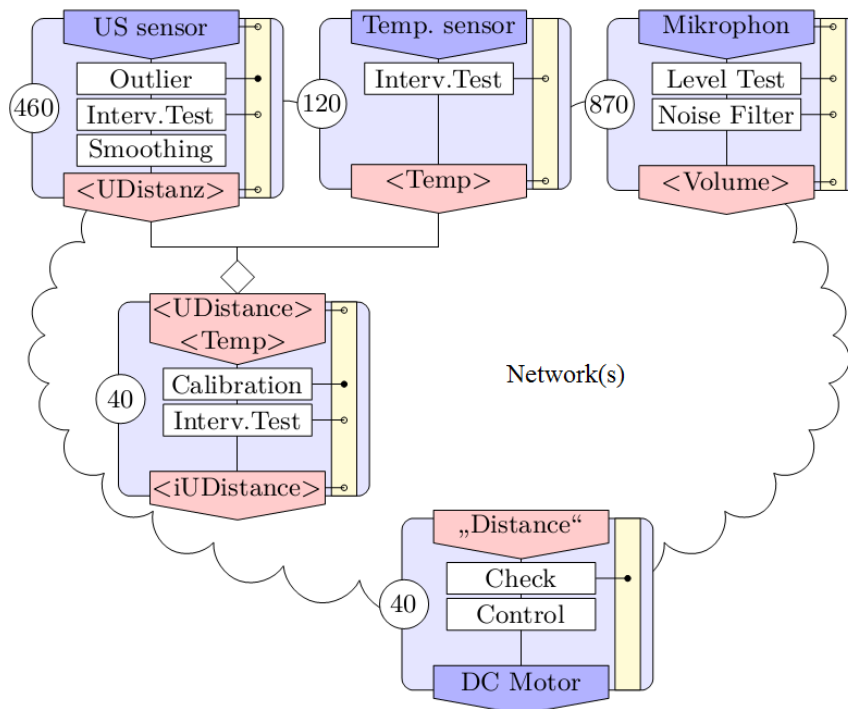


Figure 20: Example of a MOSAIC application implementing the distance control of a mobile robot.

Three different sensor nodes, one processing and one actuator node are included. The system validity of each node is indicated by the (RPN-) number inside the circles. According to the definitions done in [21] this value represents a measure of correctness for the sensor information. Obviously, the detection methods implemented on the sensors do not cover all failure modes and may not be able to guarantee a sufficiently high validity. Especially the quality of data from the ultra-sonic sensor strongly depends on the environment temperature. Hence, the application engineer may integrate a calibration node, which provides more precise results. The system validity of the calibration node is determined to be 40. The input interface of the calibration node is statically connected to the temperature and the ultra-sonic sensor. In fact, event channels of the FAMOUSO middleware are used here for communication.

The input interface of the actuator nodes is just configured for a general “Distance” pattern. It receives all distance information (identified by their physical unit), checks the consistency and calculates a respective motor command. Consequently, the microphone is not relevant for this node. In the scenario two types of distance events are available, the raw ultra-sonic measurements (“UDistance”) and calibrated ultra-sonic measurements (“iUDistance”). Related to the higher validity, the calibrated measurements obtain a much higher weight compared to the raw values. If no temperature information would be available, the calibration node would also not be able to provide a result. In this case the actuator node will work with the raw distance measurements.

A paper describing the MOSAIC approach in detail is allocated in the Annex A.2.3.

In Annex: “A fault-aware sensor architecture for cooperative mobile applications”. Joerg Kaiser, Sebastian Zug, May 2012, 26th IEEE International Parallel and Distributed Processing Symposium, Shanghai, China

3.4 Environment model

In recent years, a lot of research has been expended in developing autonomous vehicles. Controlling these autonomous vehicles in a real scenario requires information from their environment. The requirements on the perception have to cover a large spectrum of applications ranging from simple distance measurement to complex object recognition. In several applications the key issue is to represent or to interpret the neighborhood environment. Many researchers are concentrating on representing the immediate vicinity of an autonomous vehicle. This has been initially evolved from autonomous robotics applications representing a dynamic environment [24]. The environment representation of the autonomous vehicle needs an adequate integration of sensor information. Hence an intensive research is also going on in evaluating the information obtained from the available sensors [25] [26]. Predicting the environment and the environmental changes is also inspired from Advanced Driver Assistant systems (ADA’s) which are used in current vehicles. These predictions are made more accurate by using the information available from multiple and different sensors [27] [28].

For the better perception and to take decisions accordingly we need a model which can represent the environment adequately. One way of representing the environment is using mathematical models [29]. Kinematic equations describe the properties of the vehicle behavior. Several mathematical models are used for guiding the autonomous vehicles in different environments for example in 3D planes [30]. There also exist mathematical models for the navigation control of autonomous vehicles [31].

Autonomous vehicle has to adapt itself for a continuous change in environment. The representation of the environment is done by the environment model. Currently there exist some environment models to represent the vicinity of the vehicle [32] [33]. But all of them are application specific i.e. these environment models can represent the environment for implementing an application. Our approach makes the environment model provided by a middleware, which represents the environment based on the obtained information in such a way that it can be used to accomplish every aspect of the autonomous vehicle. Most of the applications require the information in different levels of detail as well as the diverse aspects of the environment. The Environment Modelling System (EMS) provides different views needed by different applications. These views are derived from a single, complex environment model.

Consider a parking assistant application as depicted in Figure 21. The vehicle evaluates 3 distance measurement sensors on its rear which are looking at center, right and left. In Figure 21, this is represented by the red beams. The information available from the sensors helps in inferring the environment at the back side of the vehicle. This is same with the sensors available in front if necessary. The autonomous vehicle has to construct its environment model from the information obtained from local sensors.

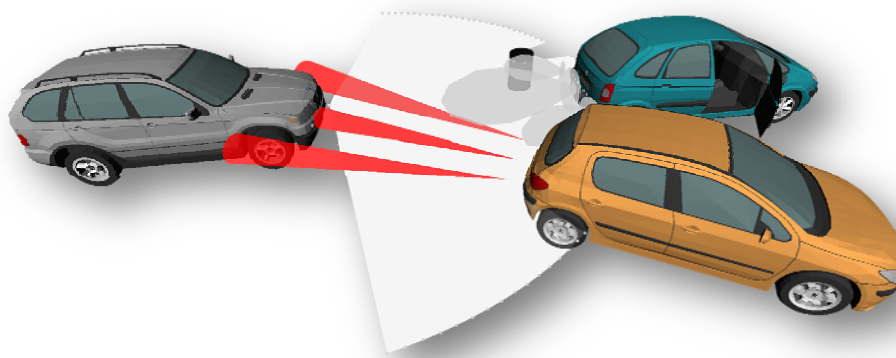


Figure 21: Parking assistance scenario.

As seen in the example, the local sensors may have blind spots in their perception and may not recognize the pole or the car close by the side. Therefore, it will be beneficial to also exploit external sensors that may cover these obstacles. Information may be obtained from other vehicles in the vicinity or from the road infrastructure. In general we have to address a number of questions which will help the autonomous vehicle to get the best possible perception of its environment like:

- What sensors are available at all?
- Are these sensors relevant for controlling the next movement?
- What is the distance to the obstacles close by?

The first point refers to the dynamic discovery and use of sensors. The MOSAIC system described above already supports this task considerably. EMS adds the possibility to interpret the sensor information in the spatial context, e.g. it allows inferring the location and the direction of an external sensor. This is a prerequisite for the dynamic use. The second point is important to filter sensor information. Those sensors that are not relevant can be neglected. Again, this filter is only possible because of environment knowledge. The third point says that we know can build a spatial model of the scene in terms of environment objects and distances.

How the environment model is exploited in the perception-action-loop is sketched in Figure 22.

As depicted in the Figure 22, each sensor has its own Electronic Data Sheet (EDS) which represent the characteristics like physical units, possible failures and behaviour at different conditions etc. This firstly helps to adapt the model to the actual environment and also is the basis that the information coming from the sensors can be handled properly by the interpreter. The interpreter shields the environment model against the details of the sensor system and provides information in terms of the model, like distance, orientation, location etc. rather than mere physical units.

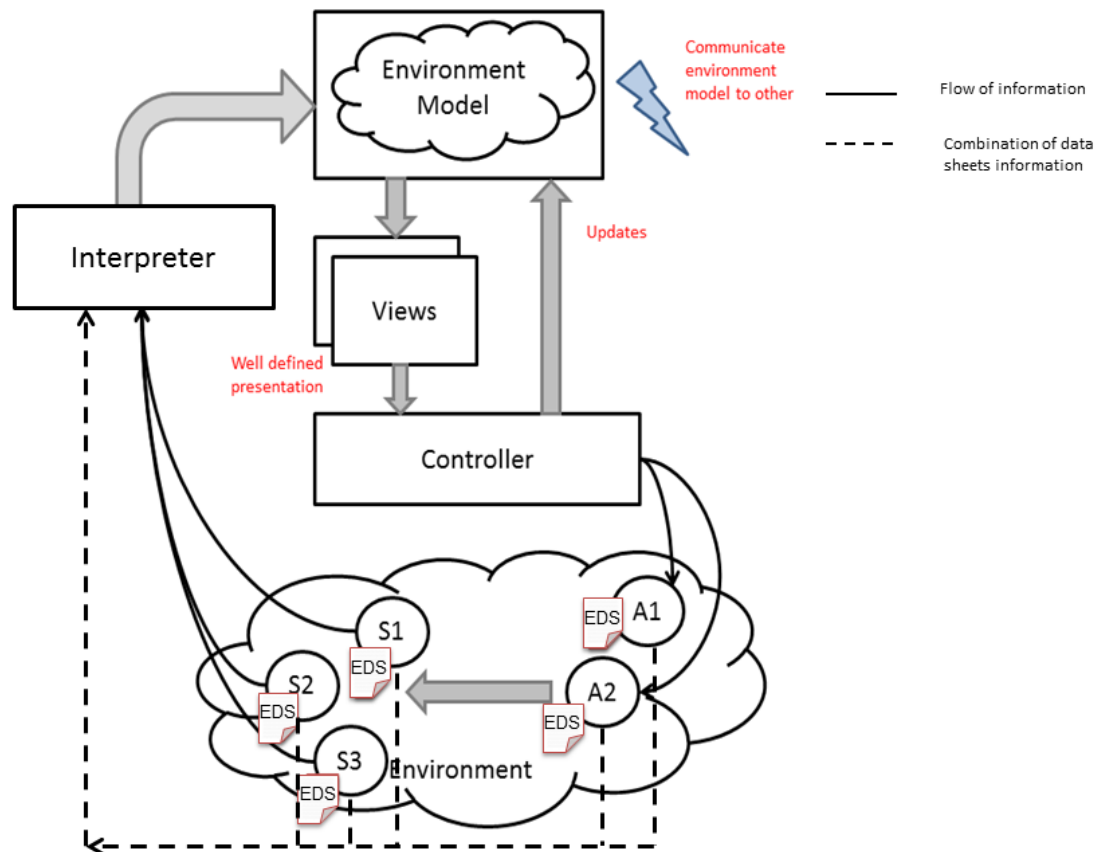


Figure 22: Environment model.

As indicated above, the environment modelling layer will serve multiple applications. A simple parking assistant may only need a 2D map of its vicinity. Recognizing pedestrians in an automatic braking system may need 3-D environment representation and also some model that predicts movements. These multiple diverse applications may rely on the same sensor set. The EMS therefore allows generating application specific views that meet the requirements of an application. In the case of cooperating vehicles like in a car platooning scenario, the environment model must also provide the views at extended level. That is each individual vehicle will share knowledge with other vehicles. The EMS enables local views to be exchanged and aggregated with other vehicles to extended views. We will distinguish between the microscopic view that just reflects the vicinity of a car and meso- and macroscopic views that model larger areas.

At this early stage of the environment modeling task in KARYON we have elaborated a model that covers geometric properties of the environment and which is continuously updated by sensor information. It builds a 3-D model and allows generating restricted views. A respective paper has been submitted for publication.

3.5 References

- [1] “AUTOSAR AUTomotive Open System ARchitecture,” [Online]. Available: <http://www.autosar.org/>, Oct 15, 2012.
- [2] P. Verissimo, V. Cahill, A. Casimiro, K. Cheverst, A. Friday and J. Kaiser, “CORTEX: Towards Supporting Autonomous and Cooperating Sentient Entities,” in Proceedings of

European Wireless 2002, Florence, Italy, February 2002.

- [3] J. Kaiser, C. Brudna, and C. Mitidieri, "COSMIC: A real-time event-based middleware for the CAN-bus,," in *Journal of Systems and Software* volume 77, pp. 27-36, Special issue: Parallel and distributed real-time systems (ISSN: 0164-1212), July 2005.
- [4] Jörg Kaiser, L.B. Becker, Sebastian Zug, and Michael Schulze, "Supporting independent development, deployment and co-operation of autonomous objects in distributed control systems,," in *9th International Symposium on Autonomous Decentralized Systems (ISADS 2009)*, pp. 195-200, , Athens Greece, March 23--25, 2009,.
- [5] S. Zug, M. Schulze, A. Dietrich und J. Kaiser, „Programming abstractions and middleware for building control systems as networks of smart sensors and actuators,“ in *s Proceedings of Emerging Technologies in Factory Automation (ETFA '10)*, Bilbao, Spain, 2010.
- [6] Rajkumar, R. and Gagliardi, M. and Sha, Lui, "The real-time publisher/subscriber inter-process communication model for distributed real-time systems: design and implementation,," in *Proceedings of the Real-Time Technology and Applications Symposium*, IEEE Computer Society, 1995, pp. 66--.
- [7] Eugster, Patrick Th. and Felber, Pascal A. and Guerraoui, Rachid and Kermarrec, Anne-Marie, "The many faces of publish/subscribe,," *ACM Comput. Surv.*, vol. 35, no. 2, pp. 114--131, June 2003.
- [8] J. Kaiser, M. Mock, "Implementing the real-time publisher/subscriber model on the controller area network (CAN),," in *In Proceedings of the 2nd International Symposium on Object-oriented Real-time distributed Computing (ISORC99)*, 1999.
- [9] M. Schulze and G.Lukas, "MLCCA -Multi-Level Composability Check Architecture for Dependable Communication over Heterogeneous Networks,," in *Proceedings of 14th International Conference on Emerging Technologies and Factory Automation* (pp. 859--866), Palma de Mallorca, Spain, 22-26 September 2009.
- [10] André Herms, Michael Schulze, Jörg Kaiser, and Edgar Nett, "Exploiting Publish/Subscribe Communication in Wireless Mesh Networks For Industrial Scenarios,," in *13th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 48-655, Hamburg, Germany, 15--18, 2008.
- [11] E. Song und K. Lee, „Understanding IEEE 1451-Networked smart transducer interface standard-What is a smart transducer?,“ *Instrumentation & Measurement Magazine, IEEE*, Bd. 11, Nr. 2, pp. 11-17, 2008.
- [12] H. Piontek und J. Kaiser, „Self-Describing Devices in COSMIC,“ in *s Proceedings of the 10th IEEE International Conference on Emerging Technologies and Factory Automation*, Catania, 2004.
- [13] W. Elmenreich, S. Pitzek und M. Schlager, „Modeling Distributed Embedded Applications on an Interface File System,“ in *s Proceedings of the Seventh IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC'04)*, Vienna, Austria, 2004, pp. 175-182.
- [14] K. Marzullo, „Tolerating Failures of Continuous-Valued Sensors,“ *ACM Transactions on Computer Systems (TOCS)*, Bd. 8, Nr. 4, pp. 284-304, 11 1990.
- [15] T. C. Henderson und M. Dekhil, „Instrumented Sensor System Architecture,“ *The International Journal of Robotics Research*, Bd. 17, Nr. 4, pp. 402-417, 1998.

- [16] S. Zug und J. Kaiser, „An approach towards smart fault-tolerant sensors,“ in s Proceedings of the International Workshop on Robotics and Sensors Environments (ROSE 2009), Lecco, Italy, 2009.
- [17] S. Zug, A. Dietrich und J. Kaiser, „An Architecture for a Dependable Distributed Sensor System,“ IEEE Transactions on Instrumentation and Measurement, Bd. 60 Issue 2, pp. 408-419, 2 2011.
- [18] O. G. Consortium, „OpenGIS Sensor Model Language (SensorML) Implementation Specification Version 1.0.0,“ 2007.
- [19] K. Lee, „IEEE 1451: A standard in support of smart transducer networking,“ in s Proc. 17th IEEE Instrumentation and Measurement Technology Conference IMTC 2000, Bd. 2, 2000, pp. 525--528.
- [20] J. Kaiser und S. Zug, „A fault-aware sensor architecture for cooperative mobile applications,“ in s 2012 IEEE 26th International Parallel and Distributed Processing Symposium (IPDPS), Shanghai, China, 2012.
- [21] S. Zug, A. Dietrich und J. Kaiser, „Fault-Handling in Networked Sensor Systems,“ in s Fault Diagnosis in Robotic and Industrial Systems, G. Rigatos, Hrsg., St. Franklin, Australia, Concept Press Ltd., 2012.
- [22] D. H. Stamatis, Failure Mode and Effect Analysis: FMEA from Theory to Execution, 2 Hrsg., {ASQ} Quality Press, 2003.
- [23] S. Zug und A. Dietrich, „Examination of Fusion Result Feedback for {Fault-Tolerant} and Distributed Sensor Systems,“ in s IEEE International Workshop on Robotic and Sensors Environments (ROSE 2010), Phoenix, AZ, USA, 2010.
- [24] H. NOLTEMEIER, „DYNAMIC ENVIRONMENTAL MODELING BY THE C-TREE,“ in s World Scientific.
- [25] Saegusa, R. and Nori, F. and Sandini, G. and Metta, G. and Sakka, S., „Sensory prediction for autonomous robots,“ in s Humanoid Robots, 2007 7th IEEE-RAS International Conference, IEEE, 2007, pp. 102--108.
- [26] Saegusa, R. and Sakka, S. and Metta, G. and Sandini, G. and others, „Sensory prediction learning-how to model the self and environment,“ in s Proceedings of the 12th IMEKO TC1-TC7 joint Symposium on" Man Science and Measurement"(IMEKO2008), 3rd-5th September, 2008, pp. 269--275.
- [27] Hermann, A. and Matzka, S. and Desel, J., „Using a proactive sensor-system in the Distributed Environment Model,“ Intelligent Vehicles Symposium, 2008 IEEE, Nr. IEEE, pp. 703--708, 2008.
- [28] Hofmann, U. and Rieder, A. and Dickmanns, ED, „Radar and vision data fusion for hybrid adaptive cruise control on highways,“ Machine Vision and Applications, Bd. 14, Nr. Springer, pp. 42--49, 2003.
- [29] P. Petrov, „A mathematical model for control of an autonomous vehicle convoy,“ Lateral (steering), Bd. 5, 2008.
- [30] Georgiev, G. and Penev, V., „Mathematical Model of Fuzzy Control System for Autonomous Guided Vehicle in 3D Space,“ INFORMATION AND SECURITY, Bd. 12, Nr. PROCON LTD, pp. 195--207, 2003.

-
- [31] K. Moriwaki, K. Tanaka, „Mathematical Modelling of an autonomous vehicle for navigation control,“ 2005.
- [32] Wang, H. and Kearney, J.K. and Cremer, J. and Willemsen, P., „Steering behaviors for autonomous vehicles in virtual environments,“ in s Virtual Reality, 2005. Proceedings. VR 2005. IEEE, IEEE, 2005, pp. 155--162.
- [33] Petrovskaya, A. and Thrun, S., „Model based vehicle tracking for autonomous driving in urban environments,“ Proceedings of Robotics: Science and Systems IV, Zurich, Switzerland, Bd. 34, 2008.

4. Reliable Assessment of Global State

We aim at developing solutions for reliable cooperation between mobile nodes. This should allow us to design reliable coordination algorithms. In order to have a consistent view about the operational state of cooperating entities and their intentions, we look into algorithms that can learn about the distributed system state of the vehicular system and its network and by that facilitate application at the higher level.

Agreement protocols are needed as building blocks for application at the higher level. For example, a vehicle should not go for a lane change until an agreement with the vehicles in close proximity can be achieved. We consider the challenging settings of wireless sensor networks. We are working toward understanding what may not be possible to achieve within a predictable time bound because it heavily relies on the inherent uncertainties of the wireless network. Therefore, we consider both Byzantine and benign system settings. We look into the necessary building block for wireless implementation of communication primitives that are needed for the reliable assessment of the distributed system state and its network. During this period of the project, we study the problem of topology discovery that is needed as a building block for the problem of Byzantine agreement. We study the possibilities of algorithmic solutions that have constant costs. Moreover, we study the possibilities of having full-scale implementations for the well-known agreement problem in benign system settings of vehicular systems.

Traditional Byzantine resilient (agreement) algorithms use $2f+1$ vertex disjoint paths to ensure message delivery in the presence of up to f Byzantine nodes. The question of how these paths are identified is related to the fundamental problem of topology discovery. Distributed algorithms for topology discovery cope with a never ending task, dealing with frequent changes in the network topology and unpredictable transient faults. Therefore, algorithms for topology discovery should be self-stabilizing to ensure convergence of the topology information following any such unpredictable sequence of events. We provide an overview of the first such algorithm that can cope with Byzantine nodes in Section 4.1, which is described in detail in [DLS12], included in Annex A.3.1. Starting in an arbitrary global state, and in the presence of f Byzantine nodes, each node is eventually aware of all the other non-Byzantine nodes and their connecting communication links.

Using the topology information, nodes can, for example, route messages across the network and deliver messages from one end user to another. We present the first deterministic, cryptographic-assumptions-free, self-stabilizing, Byzantine-resilient algorithms for network topology discovery and end-to-end message delivery.

In Section 4.1 we also consider another aspect of reliable assessment of the distributed system and its network, which is the task of r -neighbourhood discovery for the case in which r and the degree of nodes are bounded by constants. The use of r -neighbourhood discovery facilitates polynomial time, communication and space solutions for the above tasks.

In Section 4.2 we briefly look into efficient implementation of agreement protocols in wireless ad hoc networks. This is discussed in more detail in [LFZ12], which is included in Annex A.3.2. We introduce CaptureCom that omits the need for a central controller and for distinct collection and dissemination phases. The network converges to a stable state where all nodes agree on the same data.

4.1 Self-Stabilizing Byzantine Topology Discovery and Message Delivery

Self-stabilizing Byzantine resilient topology discovery is a fundamental distributed task that enables communication among parties in the network even if some of the components are compromised by an adversary. Such topology discovery is becoming extremely important nowadays where countries main infrastructures, such as the electrical smart-grid, water supply networks and intelligent transportation systems are subject to cyber-attacks. Self-stabilizing Byzantine resilient algorithms naturally cope with mobile attacks [e.g., OY91]. Whenever the set of compromised components is fixed (or dynamic, but small) during a period that suffice for convergence of the algorithm the system starts demonstrating useful behaviour following the convergence. For example, consider the case in which nodes of the smart-grid are constantly compromised by an adversary while local recovery techniques, such as local node reset and/or refresh, ensure the recovery of a compromised node after a bounded time. Once the current compromised set does not imply a partition of the communication graph, the distributed control of the smart grid automatically recovers. Self-stabilizing Byzantine resilient algorithms for topology discovery and message delivery are important for systems that have to cope with unanticipated transient violations of the assumptions that the algorithms are based upon, such as unanticipated violation of the upper number of compromised nodes and unanticipated transmission interferences that is beyond the error correction code capabilities

The dynamic and difficult-to-predict nature of electrical smart-grid and intelligent transportation systems give rise to many fault-tolerance issues and require efficient solutions. Such networks are subject to transient faults due to hardware/software temporal malfunctions or short-lived violations of the assumed settings for the location and state of their nodes. Fault-tolerant systems that are *self-stabilizing* [Dol00] can recover after the occurrence of transient faults, which can drive the system to an arbitrary system state. The system designers consider *all* configurations as possible configurations from which the system is started. The self-stabilization design criteria liberate the system designer from dealing with specific fault scenarios, risking neglecting some scenarios, and having to address each fault scenario separately.

We also consider Byzantine faults that address the possibility of a node to be compromised by an adversary and/or to run a corrupted program, rather than merely assuming that they start in an arbitrary local state. Byzantine components may behave arbitrarily (selfishly, or even maliciously) as message senders and/or as relaying nodes. For example, Byzantine nodes may block messages, selective omit messages, redirect the route of messages, playback messages, or modify messages. Any system behaviour is possible, when all (or one third or more of) the nodes are Byzantine nodes. Thus, the number of Byzantine nodes, f , is usually restricted to be less than one third of the nodes [Lyn96, Dol00].

The task of *r-neighbourhood network discovery* allows each node to know the set of nodes that are at most r hops away from it in the communication network. Moreover, the task provides information about the communication links attached to these nodes. The task *topology discovery* considers knowledge regarding the node's entire connected component. The *r-neighbourhood network discovery* and *network topology discovery* tasks are identical when r is the diameter of the communication graph.

This work presents the first deterministic self-stabilizing algorithms for *r-neighbourhood discovery* in the presence of Byzantine nodes. We assume that every *r-neighbourhood* cannot be partitioned by the Byzantine nodes. In particular, we assume the existence of at least $2f+1$ vertex disjoint paths in the *r-neighbourhood*, between any two non-Byzantine nodes, where at most f Byzantine nodes are present in the *r-neighbourhood*, rather than in the entire network. Note that by the self-stabilizing nature of our algorithms, recovery is guaranteed after a temporal violation of the above assumption. When r is defined to be the diameter of the communication graph, our assumptions are equivalent to the standard assumption for Byzantine

agreement in general (rather than only complete) communication graphs. In particular the standard assumption is that $2f+1$ vertex disjoint paths exist and *are known* (see e.g., [Lyn96]) while we present distributed algorithms to find these paths starting in an arbitrary state.

Our contribution

We present two cryptographic-assumptions-free yet secure algorithms that are deterministic, self-stabilizing and Byzantine resilient.

We start by showing the existence of deterministic, self-stabilizing, Byzantine resilient algorithms for network topology discovery and end-to-end message delivery. The algorithms convergence time is in $O(n)$. They take in to account every possible path and requiring bounded (yet exponential) memory and bounded (yet exponential) communication costs. Therefore, we also consider the task of r -neighbourhood discovery, where r is a constant. We assume that if the r -neighbourhood of a node has f Byzantine nodes, there are $2f+1$ vertex independent paths between the node and any non-Byzantine node in its r -neighbourhood. The obtained r -neighbourhood discovery requires polynomial memory and communication costs and supports deterministic, self-stabilizing, Byzantine resilient algorithm for end-to-end message delivery across the network. Unlike topology update, the proposed end-to-end message delivery algorithm establishes message exchange synchronization between end-users that is based on message reception acknowledgments.

Extensions and Conclusions

As extension, we suggest to combine the algorithms for r -neighbourhood network discovery and the end-to-end capabilities in order to allow the use of end-to-end message delivery within the r -neighbourhoods. These two algorithms can be used by the nodes, under reasonable node density assumptions, for discovering their r -neighbourhoods and then extending the scope of their end-to-end capabilities beyond their r -neighbourhood, as we sketch next. We instruct further remote nodes to relay topology information, and in this way collect information on remote neighbourhoods. One can consider an algorithm for studying specific remote neighbourhood that are defined, for example, by their geographic region, assuming the usage of GPS inputs; a specific direction and distance from the topology exploring node defines the exploration goal. The algorithm nominates $2f+1$ nodes in the specific direction to return further information towards the desired direction. The sender uses end-to-end communication to the current $2f+1$ nodes in the *front* of the current exploration, asking them for their r -neighbourhood, chooses a new set of $2f+1$ nodes for forming a new front. It then instructs each of the current nodes in the current front to communicate with each node in the chosen new front, to nominate the new front nodes to form the exploration front.

To ensure stabilization, this interactive process of remote information collection should never stop. Whenever the current collection process investigates beyond the closest r -neighbourhood, we concurrently start a new collection process in a pipeline fashion. The output is the result of the last finalized collection process. Thus, having a correct output after the first time a complete topology investigation is finalized.

In this work we presented two deterministic, self-stabilizing Byzantine resilience algorithms for topology discovery and end-to-end message delivery. We have also considered an algorithm for discovering r -neighbourhood in polynomial time, communication and space. Lastly, we mentioned a possible extension for exploring and communicating with remote r -neighbourhoods using polynomial resources as well.

The obtained end-to-end capabilities can be used for communicating the public keys of parties and establish private keys, in spite of f corrupted nodes that may try to conduct man-in-the-middle attacks, an attack that the classical Public key infrastructure (PKI) does not cope with. Once private keys are established encrypted messages can be forwarded over any specific $f+1$ node independent paths, one of which must be Byzantine free. The Byzantine free path will forward the encrypted message to the receiver while all corrupted messages will be discarded.

Since our system should be self-stabilizing, the common private secret should be re-established periodically.

In annex: “Self-Stabilizing Byzantine Resilient Topology Discovery and Message Delivery”. S. Dolev, O. Liba, E. M. Schiller, CoRR abs/1208.5620, August 2012. <http://arxiv.org/abs/1208.5620>

4.2 Capture Effect Based Communication Primitives

In this section we briefly look into efficient implementation of agreement protocols in wireless ad hoc networks. Wireless control systems consist of sensing and actuating devices that are commonly driven by a central controller. Wireless communication protocols for Cyber-Physical Systems (CPS) match this design by employing a “sense → collect → process → disseminate → actuate” flow [PSN+12], where typically different protocols are employed for collecting sensor data and disseminating actuation signals. In this paper, we depart from this traditional design and introduce CaptureCom. By relying on capture effects and in-network processing, it omits the need for a central controller and for distinct collection and dissemination phases. In CaptureCom, each node transmits its current data (e.g., temperature reading). Upon receiving, nodes integrate (e.g., aggregate) the received data with previously received data and concurrently forward the result. Due to capture effects, neighbouring nodes correctly receive one of the concurrently sent packets with high probability. Repeating this process, the network converges to a stable state where all nodes agree on the same data. The impact of our approach is threefold:

1. CaptureCom closes the loop in CPS: data are processed within the network. Upon completion, all nodes have received with high probability the same data as the base for actuation. Thus, it departs from the widespread architecture of collecting information at a central controller for processing and then disseminating the results.
2. CaptureCom exploits spatial diversity in low-power wireless networks: Consecutive, concurrent transmissions spread out across the network allow for data distribution at very low delays and high energy efficiency.
3. Relying solely on concurrent forwarding and capture effects for communication, CaptureCom simplifies the networking stack by obviating the need for link estimation, neighbour discovery, and routing protocols.

A preliminary evaluation results with a prototype implementation of CaptureCom appears in [LFZ12], included in Annex A.3.2.

Summary

We argue that CaptureCom closes the control loop in wireless CPS: it provides efficient collection, processing (in our example aggregation), and dissemination within the network, making it suitable for widespread control applications. Our initial evaluation shows that CaptureCom efficiently exploits spacial diversity. It converges within 125 ms at high reliability, leading to low communication delays and high energy efficiency. Motivated by these promising initial results, we are currently investigating the theoretical and experimental foundations of CaptureCom as a novel primitive for wireless CPS.

In annex: “Capture effect based communication primitives”. O. Landsiedel, F. Ferrari, and M. Zimmerling, In SenSys’12: Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, Toronto, Canada, November 2012. <ftp://ftp.tik.ee.ethz.ch/pub/people/marcoz/LFZ2012.pdf>

4.3 References

- [DCA11] Manjunath Doddavenkatappa, Mun Choon Chan, and Akkihebbal L. Ananda. Indriya: A low-cost, 3d wireless sensor network testbed. In Thanasis Korakis, Hongbin Li, Phuoc Tran-Gia, and Hong-Shik Park, editors, *TRIDENTCOM*, volume 90 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 302–316. Springer, 2011.
- [DLS12] Shlomi Dolev, Omri Liba, and Elad Michael Schiller. Self-stabilizing Byzantine resilient topology discovery and message delivery. *CoRR*, abs/1208.5620, 2012.
- [Dol00] Shlomi Dolev. *Self-Stabilization*. MIT Press, 2000.
- [FZTM12] Federico Ferrari, Marco Zimmerling, Lothar Thiele, and Luca Mottola. The low-power wireless bus: simplicity is (again) the soul of efficiency. In Zhao et al. [ZTW12], pages 93–94.
- [FZTS11] Federico Ferrari, Marco Zimmerling, Lothar Thiele, and Olga Saukh. Efficient network flooding and time synchronization with glossy. In Xenofon D. Koutsoukos, Koen Langendoen, Gregory J. Pottie, and Vijay Raghunathan, editors, *IPSN*, pages 73–84. IEEE, 2011.
- [LF76] K. Leentvaar and J. Flint. The capture effect in FM receivers. *Communications, IEEE Transactions on*, 24(5):531–539, 1976.
- [LFZ12] Olaf Landsiedel, Federico Ferrari, and Marco Zimmerling. Poster abstract: Capture effect based communication primitives. In *SenSys'12: Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, Toronto, Canada, November 2012. Poster and Extended Abstract.
- [LW09] Jiakang Lu and Kamin Whitehouse. Flash flooding: Exploiting the capture effect for rapid flooding in wireless sensor networks. In *INFOCOM*, pages 2491–2499. IEEE, 2009.
- [Lyn96] Nancy Lynch. *Distributed Computing*. Morgan Kaufmann Publishers, 1996.
- [NT09] Mikhail Nesterenko and Sebastien Tixeuil. Discovering network topology in the presence of byzantine faults. *IEEE Trans. Parallel Distrib. Syst.*, 20(12):1777–1789, 2009.
- [OY91] Rafail Ostrovsky and Moti Yung. How to withstand mobile virus attacks (extended abstract). In Luigi Logrippo, editor, *PODC*, pages 51–59. ACM, 1991.
- [PSN+12] Miroslav Pajic, Shreyas Sundaram, Jerome Le Ny, George J. Pappas, and Rahul Mangharam. Closing the loop: a simple distributed method for control over wireless networks. In Zhao et al. [ZTW12], pages 25–36.
- [ZTW12] Feng Zhao, Andreas Terzis, and Kamin Whitehouse, editors. The 11th International Conference on Information Processing in Sensor Networks (co-located with CPS Week 2012), IPSN 2012, Beijing, China, April 16-19, 2012. ACM, 2012.

Annex A Papers and Reports

A total of 12 papers and reports are provided as annex to the deliverable. Therefore, in order to make it easier handling the deliverable, this annex is provided as a separate document. We just provide here a listing of the sections in the annex and the respectively included documents.

A.1 Predictability and Resilience in Embedded Networks

A.1.1 An Approach to Enhance the Timeliness of Wireless Communications

“An Approach to Enhance the Timeliness of Wireless Communications”. J. L. R. Souza and J. Rufino, Fifth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM 2011), November 2011, Lisbon, Portugal.

A.1.2 Characterization of Network Inaccessibility in IEEE 802.15.4 Wireless Networks

“Characterization of Network Inaccessibility in IEEE 802.15.4 Wireless Networks”. J. L. R. Souza and J. Rufino, Technical Report DI/FCUL, September 2012, Lisbon, Portugal.

A.1.3 Characterizing Inaccessibility in IEEE 802.15.4 Through Theoretical Models and Simulation Tools

“Characterizing Inaccessibility in IEEE 802.15.4 Through Theoretical Models and Simulation Tools”. J. L. R. Souza, A. Guerreiro and J. Rufino. INForum 1012 Simpósio de Informática – Embedded and Real-Time Systems Track. September 2012, Caparica, Portugal.

A.1.4 Reducing Inaccessibility in IEEE 802.15.4 Wireless Communications

“Reducing Inaccessibility in IEEE 802.15.4 Wireless Communications”. J. L. R. Souza and J. Rufino, (submitted for publication).

A.1.5 Self-Stabilizing TDMA algorithms for Dynamic Wireless Ad-hoc Networks

“Self-Stabilizing TDMA algorithms for Dynamic Wireless Ad-hoc Networks”. Pierre Leone and Elad Michael Schiller. The 8th International Symposium on Algorithms for Sensor Systems, Wireless Ad Hoc Networks and Autonomous Mobile, 2012.

A.1.6 Autonomous TDMA alignment for VANETs

“Autonomous TDMA alignment for VANETs”. Mohamed Hassan Mustafa, Marina Papatriantafidou, Elad M. Schiller, Amir Tohidi, and Philippas Tsigas, In IEEE 76th Vehicular Technology Conference (VTC’12-Fall), 2012.

A.1.7 Self-Stabilizing End-to-End Communication in Bounded Capacity, Omitting, Duplicating and Non-FIFO Dynamic Networks

“Self-Stabilizing End-to-End Communication in Bounded Capacity, Omitting, Duplicating and Non-FIFO Dynamic Networks”. S. Dolev, H. Ariel, E. M. Schiller and S. Sharma, 14th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS’12), Toronto, Canada, October 2012.

A.2 Adaptive Middleware for Advanced Control Systems

A.2.1 Lightweight Dependable Adaptation for Wireless Sensor Networks

“Lightweight Dependable Adaptation for Wireless Sensor Networks”. L. Marques and A. Casimiro, Technical Report DI/FCUL, September 2012, Lisbon, Portugal.

A.2.2 Programming abstractions and middleware for building control systems as networks of smart sensors and actuators

“Programming abstractions and middleware for building control systems as networks of smart sensors and actuators”. Sebastian Zug, Michael Schulze, Andre Dietrich, Joerg Kaiser, September 2010, ETFA 2010 - 15th IEEE International Conference on Emerging Technologies and Factory Automation, Bilbao, Spain.

A.2.3 A fault-aware sensor architecture for cooperative mobile applications

“A fault-aware sensor architecture for cooperative mobile applications”. Joerg Kaiser, Sebastian Zug, May 2012, 26th IEEE International Parallel and Distributed Processing Symposium, Shanghai, China.

A.3 Reliable Assessment of Global State

A.3.1 Self-Stabilizing Byzantine Resilient Topology Discovery and Message Delivery

“Self-Stabilizing Byzantine Resilient Topology Discovery and Message Delivery”. S. Dolev, O. Liba, E. M. Schiller, CoRR abs/1208.5620, August 2012. <http://arxiv.org/abs/1208.5620>

A.3.2 Capture effect based communication primitives

“Capture effect based communication primitives”. O. Landsiedel, F. Ferrari, and M. Zimmerling, In SenSys’12: Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, Toronto, Canada, November 2012. <ftp://ftp.tik.ee.ethz.ch/pub/people/marcoz/LFZ2012.pdf>