Innovation Report

On-Board and Off-Board Data Platforms

Submitted by: Mx. Liz James *MPhys*, November 2019

Abstract

The automotive industry is facing significant technological barriers with the shift toward Connected and Autonomous Vehicles (CAVs). With technology trends such as Big Data and the Internet of Things (IoT), demonstrating that effective utilisation of data can lead to a competitive advantage, the question becomes: what is the best approach to gathering, disseminating and utilising the data available within the vehicular platform? Key challenges that have arisen during this adoption period include: how should access to the data existing within current vehicular platforms be managed and how can existing application development paradigms, such as Publish/Subscribe and Service Orientation, be employed to ease development of the increasingly diverse range of systems integrating with vehicles. The initial stage of research aimed to address these challenges associated with Vehicle-to-Cloud (V2C) applications, whose requirements cannot be precisely known in advance, by introducing a multi-broker Publish/Subscribe system. In order to ensure effective coupling between the Vehicle and Cloud components, synchronisation between the two brokers was managed through an example optimisation function laying the framework for delivering cost constrained applications. Through exploring the capabilities of the proposed multi-broker Publish/Subscribe system several limitations were highlighted - unexposed signal data; and a high, often fixed signal latency. Emerging complementary trends in Automotive Networking include the adoption of Automotive-compliant Ethernet PHYs, Quality of Service (QoS) standards (802.1Qav and 802.1Qbv [1]) and Service Orientated Architectures (enabled through IP abstraction). Service Orientated Architectures could address the challenges associated with unexposed signal data when supported by higher speed networking by allowing services to request the data dynamically rather than requiring firmware changes to expose new data. This led to proposing a new non deterministic method, that can run alongside statically configured safety-critical streams, for delivering applications on an Ethernet-based network utilising a dynamic priority allocation model, which assigns Ethernet frame parameters at run-time, in order to meet its application latency requirements whilst remaining responsive to changes in application topology. With the contributions from this research, the automotive industry now has: an alternative to the traditional bottom-up network development approach that applies the same constraint model to all traffic; and a Vehicle-to-Cloud application development framework that allows for applications to be developed completely independently of vehicular hardware while supporting vehicle-side decision making. By establishing these alternative approaches early, the adoption of Automotive Ethernet, IP, Service Orientated Architectures and integrated Vehicle-to-Cloud solutions, the automotive industry can make informed decisions with regard to vehicular platform design.

WMG
THE UNIVERSITY OF WARWICK

Contents

List of Figures

Declaration

I hereby declare that all work contained in this report was produced by the author and that none of the work has been previously submitted for an academic degree.

All sources of quoted work have been referenced accordingly.

Signed: [redacted]

Liz James

WARWICK
THE UNIVERSITY OF WARWICK
Department - WMG
November 2019

Acknowledgements

I would like to formally acknowledge the sponsor of the RACeD Engineering Doctoral research programme - Jaguar Land Rover.

My academic mentors have been entirely supportive throughout the process, ensuring timely feedback and an ever-useful external frame of reference.

- Professor Paul Jennings
- Professor Mehrdad Dianati (Years 3 - 4)
- Dr Harita Joshi (Years 1 - 2)

I have been fortunate to have been provided with highly experienced and technically gifted mentors from industry whose insight has been invaluable throughout the process.

- Mr Tom Mizutani
- Mr David Oxtoby
- Dr Harita Joshi (Years 3 - 4)

I would like to acknowledge the following people for their unwavering support that has definitely played a key role in my getting to this stage.

- Ms Lee-Rose Jordan - Project Management
- Mr Mike Oldridge - Pastoral + ASD Mentor
- Ms Despina Weber - Disability Services

And whilst their contributions are significantly wider reaching and harder to quantify, I'd like to thank my parents, younger brother and foster brother for their unending desire to support and assist pastorally during periods of extended stress.

Acronyms

2G 2nd Generation Cellular.

3G 3rd Generation Cellular.

3GPP 3rd Generation Partnership Project.

4G 4th Generation Cellular.

5G 5th Generation Cellular.

5G NR-V2X 5G New Radio (NR) continues the development from LTE-V2X.

AFDX Avionics Full-Duplex Switched Ethernet.

ARINC Aeronautical Radio, Incorporated.

AUTOSAR AUTomotive Open System ARchitecture.

AVB Audio Video Bridging.

CAN Controller Area Network.

CAN-FD CAN FD (Controller Area Network Flexible Data-Rate).

CAV Connected and Autonomous Vehicle.

CBD Credit-based Shaper.

CRC Cyclic Redundancy Check.

CSMA-CD Carrier Sense Multiple Access with Collision Detection.

DEI Drop Eligible Indicator.

DES Discrete Event Simulation.

ECU Electronic Control Unit.

EMC Electromagnetic compatibility.

EV Electric Vehicle.

GCL Gate Control List.

GDPR General Data Protection Regulation.

GMII Gigabit media-independent Interface.

IEEE Institute of Electrical and Electronics Engineers.

IoT Internet of Things.

IPC Instrument Panel Cluster.

ISM Industrial, Scientific and Medical.

IVN In-Vehicle Network.

LAN Local Area Network.

LIN Local Interconnect Network.

LLC Logical Link Control.

MAC Medium Access Control.

MDI Medium Dependent Interface.

MISRA Motor Industry Software Reliability Association.

MOST Media Oriented System Transport.

NGA Next Generation Architecture.

NIT Network Idle Time.

OEM Original Equipment Manufacturer.

OMNeT++ Objective Modular Network Testbed in C++.

OPC Open Platform Communications.

OPC-UA Open Platform Communications Unified Architecture.

OSI Open Systems Interconnection.

PC5 Name of 3GPP device-to-device interface proposed for C-V2X.

PCP Priority Code Point.

PHY Physical Layer.

PLCA PHY-Level Collision Avoidance.

PLS Physical Coding Sublayer.

PMA Physical Medium Attachment.

POF Plastic Optical Fibre.

PoI Point of Interest.

QoS Quality of Service.

RSU Road Side Unit.

SoA Service Orientated Architecture.

TAS Time Aware Scheduler.

TDMA Time Division Multiple Access.

TSN Time Sensitive Network.

UMTS Universal Mobile Telecommunications System.

UNECE United Nations Economic Commission for Europe.

UTP Unshielded Twisted Pair.

Uu UMTS air interface.

V2C Vehicle to Cloud.

VID VLAN identifier.

VLAN virtual LAN.

XGMII 10 Gigabit Media Independent Interface.

1 Introduction

The vehicles being designed and manufactured for use on the road today are, in general, more complex than ever before.

Electronic Control Units (ECUs) have been employed commercially by the automotive industry since the wide-spread introduction of integrated circuits in the 1970s.

In general, ECUs of this era were highly specialised units designed to perform a single task such as providing precise timing for efficient operation of the engine. Over-time new applications required their own electronic control units such as electronically controlled Anti-Lock Braking Systems (ABS).

Eventually, new applications began requiring access to the same sensors and actuators and it became apparent that running these typically safety or primary functionality orientated, features using the same sets of sensors and actuators would quickly become problematic.

There are numerous potential issues with having several independent systems being connected to the same Input/Output (I/O), collectively referred to as bus contention for digital signals. For analogue devices that are particularly sensitive to noise, having many additional wires and current sinks connected could lead to variation in measured sample based around the number of devices currently taking a measurement which is unwanted.

The solution adopted by the automotive industry, in the situation where it was cost prohibitive to duplicate the sensors entirely, was to develop data networks between ECUs to have only a single ECU+sensor that would exchange the important measurements to other devices that need them. With the introduction of ECUs into the vehicular platform came the introduction of firmware, which was developed by manufactures/suppliers to perform the numerous and diverse tasks/features that had begun to find their way into these units.

Original Equipment Manufacturers (OEMs) are looking to provide consumers with product ranges that have meaningful differentiation while simultaneously aiming to reduce development costs and improve reliability. To develop a wider range of variants, it became necessary to reuse solutions from existing products, which introduced additional complexity when working with suppliers who could change the electrical pin-outs and firmware interfaces for their own solutions between generations of product.

In 2002, AUTomotive Open System ARchitecture (AUTOSAR) partnership was formed between OEMs, automotive suppliers, software suppliers, and various other organisations within the supply chain to address these challenges with a focus on [2]:

- Supporting the transferability of software
- Supporting scalability to different vehicles and platform variants
- Support a broad variety of functional domains

- Support collaboration between various partners

With a 2015 study [3], common benefits of AUTOSAR were listed as standardisation, reusability, and interoperability with some of the downsides being listed being complexity, initial investment and the steep learning curve which align strongly with the original partnership objectives.

Until relatively recently, the discussion around vehicular technologies has treated the vehicular platform as an isolated system with interaction with external systems being limited to diagnostic tools and external media sources. This approach has allowed OEMs to focus upon the restricted number of permutations of hardware and software that they officially support however, with increasing pressure to provide new functionality, such as improved servicing, collision avoidance and parking assistance [4] [5] and an ever present to exploit the value of the data contained within vehicular platforms with the European Commission funded research proposing a market value of €122 billion shared amongst OEMs, suppliers, service providers etc. [6] [7].

With market forecasts for Connected and Autonomous Vehicles (CAVs) highlighting the potential for a £907 billion global market for CAVs and a £63 billion global market for various supporting technologies [8], it has become a key area for actors in the automotive industry to try to gather competitive advantage in key areas. A study by the Society for Motor Manufacturers and Traders (SMMT) has identified that the United Kingdom is currently ahead of its international competitors in regard to the regulatory environment and market demand for the numerous technologies associated with CAVs [9].

This position of strength provides the background justification for the research project discussed within as a collaboration between academia and industry to exploit the situation and address future technological challenges that face the automotive industry during the period of transition from traditional vehicles toward CAVs.

2 Methodology

One of the assessment criteria of the Engineering Doctorate is innovation. Given the numerous challenges discussed in Section 1, the industrial research sponsor, Jaguar Land Rover, and academic institution University of Warwick through its department WMG wanted a technology-focused research project. The culmination of which is presented here in the Innovation Report as the evidence of the innovative output of the project.

Figure 1 presents a diagrammatic layout of the various documents, or submissions, that when presented here, provide an insight/'story' of the contributions of the portfolio. The research area attached to the title of "On-Board and Off-Board Data Platforms" is extremely broad straddling several huge fields of specialism including data collection, network technologies, decentralised applications, telematics and analytics before moving into automotive industry specific variants of the above.

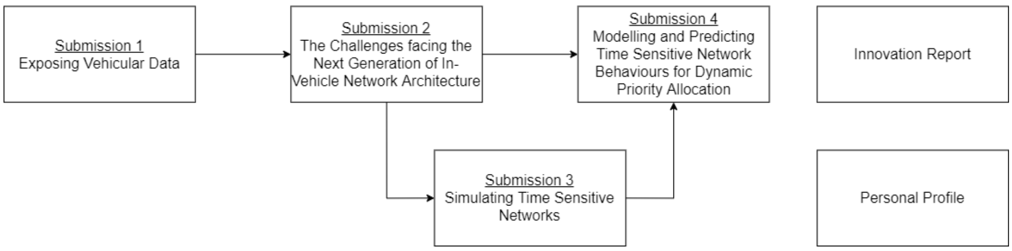


Figure 1: Portfolio Layout

In Section 3, the document will deep-dive into both the commercial activities of OEMs and suppliers as well as provide insight into the numerous technical elements that represent the state-of-art from the various technologies and sector-specific knowledge utilised throughout this project. This deep dive provides and opportunity for the reader to gain an insight into the various challenges facing the automotive industry.

Section 3 is broken down into various different sections, strongly aligned with the topics associated with each portfolio submission, with Section 3.2 explores the state of the art in terms of both commercial and academic Vehicle-to-Cloud (V2C) research which strongly aligns with the portfolio submission "Exposing Vehicular Data"

The document moves into sub-section 3.3 which explores relevant research and industry trends with regard to software architectures and development paradigms, with sub-section 3.4 presenting research pertaining to In-Vehicle network technologies. These sections are closely tied to the content of the second portfolio submission "The Challenges Facing the Next Generation of In-Vehicle Network Architectures" with the final sub-section, Section 3.5, exploring the State of the Art research associated with the final submission, "Modelling and Predicting Time Sensitive Network Behaviours for Dynamic Priority Allocation". The final portfolio contribution explores various approaches to configuring Ethernet-based networks utilising Time Sensitive Networking standards.

With the relevant State of the Art knowledge having been presented, the document turns to the portfolio contributions, or 'story of innovation', with Section 4 that begins with the research contributions of the first portfolio entry named "Exposing Vehicular Data".

The research area: "On-board and Off-board Data platforms" was extremely ill defined at the beginning of the research project with the potential to expand into the plethora of existing research areas associated with Smart, Connected and Automated Vehicles.

With the project stakeholders being unable to formally express the challenges they were facing in this space, one approach (Exploratory Research) could have been to look toward academic literature and attempt to identify gaps in the existing research. The scope was so broad, however that it is likely that there would have been numerous career-defining areas of research as starting points. Instead, a more Applied Research approach was taken as the starting point of the portfolio, this was to ensure that the research portfolio was tied to fundamental problems that were being faced by stakeholders within Jaguar Land Rover, a brownfield approach rather than addressing the problem greenfield.

To this end, Submission 1 started with a known problem space of the next generation of vehicular architecture, Vehicle-to-Cloud. An applied research approach or continuing the above analogy, a brownfield approach to the research being performed by an academic institution, enables an exploration of the problem space without the restrictions that typically accompany the traditional automotive product development models, and as such new challenges could be discovered.

By combining the ideal requirements of a Vehicle-to-Cloud system that include the ability for local applications to continue to operate when there isn't a cellular network and be able to respond to local contextual changes that would have an impact on what information should be captured/recorded dynamically. These properties led to the proposal of a dual broker system, in which applications can sit On-board, Off-board, or both that, when connected to the various automotive networks on the vehicle. Drawing Section 4 to a close will explore the various contributions, its strengths and weaknesses and provide the basis for the next contribution discussed in Section 5, "The Challenges Facing the Next Generation of In-Vehicle Network Architectures", which explores current In-Vehicle Networking technologies and Automotive-compliant Ethernet which represents a key area of interest in potentially addressing the limitations identified in Section 4.

While the applied research led to some conclusions around the data available in the vehicle the jump to Discrete Event Simulation and Automotive Ethernet might not be so clear. The initial exposure to stakeholders throughout the organisation provided insight into the long-term priorities of various departments, the changes to electrical/electronic (E/E) architectures, and the evolving conversation around software complexity in vehicles both had stakes in the ongoing developments in Networking and Software architectures.

While at the end of the research, the availability and maturity of these technologies has moved forward significantly, the challenge around access to hardware platforms and robust software frameworks limited the path forward toward a more quantitative research framework. Pulling in other areas of interest to the automotive industry, including Software/Hardware-in-the-Loop and Simulated test environments exposed the need for the ability to perform some degree of testing in this space.

Section 6, "Simulating Time Sensitive Networks", begins to address the findings from Submission 1 and Submission 2 which include: the limited subset of signal data exposed on traditional field buses due to ideal optimisation strategies for low bitrate, shared medium buses and legacy automotive architectures. The document turns toward the development of a simulation environment in which to work with various automotive Ethernet technologies, which could assist with addressing the aforementioned challenges, including a discussion around the benefits and limitations of such an environment in comparison to a hardware implementation. Section 7, "Modelling and Predicting Time Sensitive Network Behaviours for Dynamic Priority Allocation", continues to explore how the simulation environment discussed in Section 6 can be used to present an alternative approach for configuring applications and Time Sensitive Networks focusing more upon the applications running on the vehicular platform rather than the physical hardware (top-down) which in turn is one way of achieving the desirable publish/subscribe for all signals, and run time dynamism that would complement the proposed architecture from Submission 1.

Once a complete exploration of the various elements of the project has been provided Section 8, the Discussion, looks to tie it all together and discuss the benefits to the wider industry of the various contributions of this work.

Section 9, the Conclusion, looks to tie together all of the thematic elements that have been brought up in the document and provide a comprehensive summary of the document.

3 State of Art

As first introduced in Section 2, this section draws out various components of literature that is relevant for the numerous different technologies and themes that have arisen throughout the portfolio.

While using a traditional cellular modem is the current state of the art for Vehicle to Cloud applications, a more general framework for V2X is being expanded upon in the 3GPP facilitated by Cellular Vehicle-to-Everything (C-V2X).

3.1 Cellular Vehicle-to-Everything (C-V2X) and 5G

The cellular part of C-V2X is used to differentiate this set of technologies from earlier approaches such as Dedicated Short Range Communications (DSRC) which was built using a specialist PHY+MAC combination derived from the wireless communications standards developed as part of 802.11. 802.11p looked to shrink channel sizes to enable a more robust handling of channel effects such as multi-path, Doppler shift and other forms of interference that are more common in a vehicular environment whilst adding management frames that enabled a shared timing reference for all the device. However in 2020 the Federal Communications Commission (FCC) deemed the lack of adoption of DSRC as sufficient reason to reallocate the spectrum to the ISM band and to the rest to C-V2X.

A Cellular Vehicle to Everything (C-V2X) system, has three defined operating modes for different types of communication [10]:

- Device-to-Device
- Device-to-Cell tower
- Device-to-Network

Of these Device-to-Network is likely the most familiar as it is the underlying mechanism that enables internet access on consumer electronic devices such a mobile phones. It utilises the the Uu interface, with Device-to-Device communication being facilitated using the PC5 interface [11].

A comparison between 4G V2X specification and the 5G version of the same functionality can be seen in Figure 2.

When operating outside a cellular network, in Mode 4 (under the LTE-V2X framework) and Mode 2 (under the 5G NR-V2X) two categories of device can typically exist: Road Side Units (RSU) and On-Board Units (OBU) with compliant chipsets these can be leveraged to facilitate various applications such as:

- Signal Phase and Timing

Table 1. Difference between LTE-V2X and NR-V2X.

Items	LTE-V2X	NR-V2X
Specification	3GPP Rel-14/Rel-15	3GPP Rel-16/Rel-17
Latency	Low latency: 10~100 ms	Ultra-low latency: 1 ms [17]
PC5 Message Type	Broadcast	Broadcast, Unicast and Groupcast
Application Scenario	Safety related/Enhanced	Advanced Application
Cellular Network Coverage	Uu and PC5 Mode3	Uu and PC5 Mode1
Out of Cellular Network Coverage	PC5 Mode4	PC5 Mode2

Figure 2: Table comparing the LTE-V2X and the 5G NR-V2X communication standards [11]

- Traffic Signal Preemption
- Warnings
  - Vulnerable Road User Collision warning
  - Intersection Collision warning
  - Emergency Brake warning
  - Do Not Pass warning
  - Hazardous Location warning

The automotive industry has traditionally struggled with standardisation efforts for a variety of reasons including the significant latency from the inception of an idea to the time it becomes standardised and ensuring that industry specific considerations are taken into account. In more recent times, this has led to the creation of a variety of industry consortia aiming to accelerate the development of new standards by actively specifying and then beginning the formal development of industry specific standards. Some relevant examples include the OPEN (One-Pair Ether-Net) Alliance; The Connected Vehicle Systems Alliance (COVESA) and the 5G Automotive Association (5GAA).

The 5GAA is actively working in the domain of Cooperative Intelligent Transportation Systems (C-ITS) and looking to promote the adoption of (Vehicle to Everything) V2X communications. As with all things improving adoption includes promotion (case studies), improving and expanding the capabilities of these systems (standardisation and regulation), establishing a robust certification and testing process to build trust.

3.2 Vehicle-to-Cloud Applications (V2C)

Vehicle-to-Cloud (V2C) refers to a specific type of CAV communication framework in which the vehicle utilises some form of network technology to communicate with systems that exist independently, elsewhere on the internet.

When discussing V2C applications the network technology in question is typically taken to be an integrated vehicle cellular modem connecting to an Mobile Network Operator (MNO) or Mobile Virtual Network Operator (MVNO). As discussed above, this type of system is likely to shift toward C-V2X in which V2C is just part of the device-to-network functionality. The similarity between vehicles and mobile phones in this regard is important because it relegates much of the research domain to optimising the Physical Layer (PHY) parameters of the cellular modems to operate effectively: when moving at speed; with complex antenna position constraints; with multiple signal paths and ensuring broad signal coverage.

In order to explore the State of the Art for Vehicle-to-Cloud applications then, it becomes important to identify the key trends in the development of cellular technologies. At the current time, in the United Kingdom, the adoption of the 5th Generation (5G) of Cellular Networking technologies is picking up pace with several mobile manufacturers beginning to include the functionality in their latest/upcoming products [12].

With the GSM Association (GSMA) identifying a projected global contribution of millimetre wave 5G for Next Generation Transport connectivity being 14% of a total $565 billion in 2034 [13] it highlights the focus of the automotive industry to identify the best approach to integrate it with their products.

However for the purposes of this literature review it is important to consider the technologies being targeted by the sector at the time. Which looked to utilise both 2G and 4G cellular technologies and build upon the capabilities of eCall-compliant vehicles that are designed to provide quick emergency response in case of a road accident [14]. eCall is an automotive emergency response system that is now mandated for all new models of passenger cars within the European Union by March 2018 exchanging at least a Minimum Set of Data (MSD) to a Public Safety Answering Point (PSAP) and/or establishing a call to an operator [14]. Given that the functionality of eCall builds upon Global System for Mobile Communications (GSM) the coverage is critical [15] with the UK regulator OFCOM finding that 97% of the UK population are covered and 91% of the land mass were covered by GSM service (2G) in 2010 [16] with significant disparity between Scotland, Wales and Northern Ireland [16].

As discussed before, the most common type of network technology for Vehicle to Cloud is to utilise existing cellular technologies however an alternative approach utilising road-side infrastructure, that has a wired network connection, could be utilised to deploy V2C applications via an Vehicle to Infrastructure (V2I) intermediary.

[17] presents a solution for vehicle monitoring that combines road side units (RSUs) and a cloud-based application to provide non-automotive devices access to the information contained within the RSUs. Whilst within [17] the focus is primarily upon applications requiring low-latency such as Time of

Arrival localization to the RSUs, collision avoidance and speed based lane changes many of the other applications including vehicle/accident detection, video surveillance and emergency message propagation all fit under the domain of Vehicle to Cloud.

More traditional approaches to V2C applications can be seen in [18] in which the 'Cloud' is utilised as an external computing platform and as part of a wider system such as seen in Figure 3.

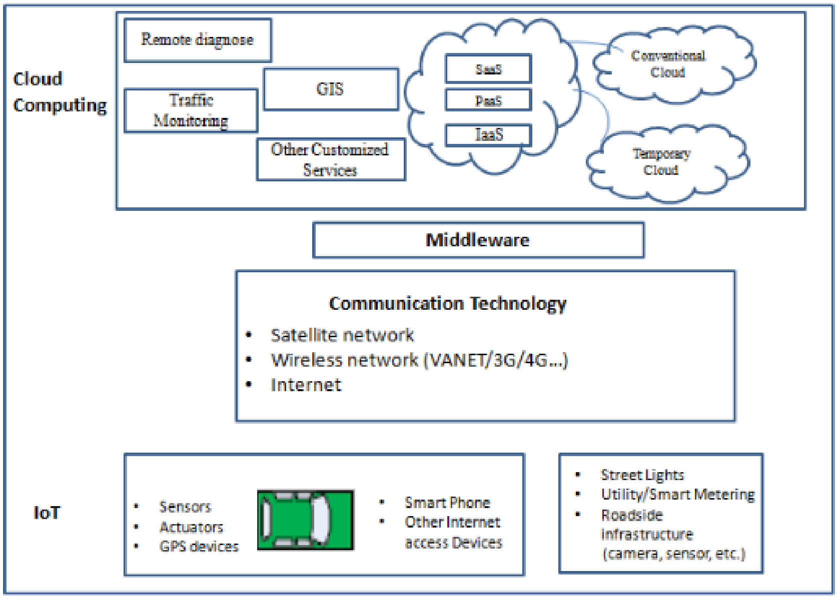


Figure 3: Architecture proposed for IoT-based Vehicular Data Clouds [18]

[18] breaks down the three cloud-based services as follows: Infrastructure as a Service (IAAS); Storage as a Service (SAAS) and Platform as a Service (PAAS) in which a vehicle provides some of its compute capability to other vehicles, or requires cloud-based storage to off-load information, or contributes information alongside other vehicles to provide new functions respectively.

Given that from an Internet of Things (IoT) perspective, a vehicle is just a source of information and compute capability it becomes important to examine how an IoT framework fits into the V2C paradigm.

[19] presents a survey of the different types of IoT system architectures in use across a variety of domains and highlights many technical challenges associated with the design of Service Orientated Architectures and the common lack of a Service Description Language makes the implementation of such systems increasingly complex.

3.2.1 Commercial Implementations and Applications of V2C

Vehicle to Cloud (V2C) is a highly broad field with nuances in the network technologies used to provide the connectivity to the 'Cloud'. In order to establish how the Automotive industry is currently utilising these technologies the document now aims to introduce and discuss the strengths and benefits of commercial V2C implementations.

Ericsson, a supplier for IoT automotive solutions, presents it's "Connected vehicle telematics service" with key features such as remote safety which builds upon eCall with optional support for breakdowns, remote control and status , remote diagnostic code handling and remote security [20]. It's important to note here that the implementation specifics are not available for consideration and it is entirely feasible that features such as "Remote safety" are entirely contained as a minor enhancement to eCall functionality rather than being part of a fully integrated cloud-base suite of applications. Given that Ericsson is a supplier it is entirely likely however that this functionality does exist within as an option for OEMs such as Volvo who provide digital vehicle services and over-the-air updates through the Ericsson offering [21].

Easy examples for existing commercial V2C deployments can be found throughout the automotive industry, from Infotainment through the integrated modem, PoI data for the navigation services and real-time traffic data [22] [23] [24] [25]. Many of these features are enhanced by or solely delivered by a connection to a cloud service that is providing information that can be delivered to the customer through their infotainment or instrument panel cluster (IPC).

Another successful demonstration of the power of V2C applications can be seen in City Data Solutions by Ford Smart Mobility [26] who utilised 160 light commercial vehicles with smart on-board computers alongside private-use vehicles (model: Ford Fiesta). It is important to note here that the information was made available using plug-in devices that captured data through the diagnostic port. Through this relatively small study they were able to collect 15,000 days of vehicle operation and over 500 million data records [26] which was used to examine: optimal EV charge point placement based around the inactivity of the commercial vehicles; optional shift timings by identifying optimal times to drive specific routes and explored the impact of events and times of year of traffic flow [26]. As part of this study Ford Smart Mobility utilised a framework to facilitate their data capture Autonomic.ai and the Autonomic Transportation Mobility Cloud (TMC) [27] who aims to provide a unified set of APIs for the wider automotive industry in an attempt to provide the solution to implementing V2C applications for OEMs that might wish not to develop V2C as a core competence instead utilising a 3rd party [27].

3.3 Embedded Software in the Automotive Sector

The automotive sector has been an industry that has long utilised electronic devices to improve the performance of various systems throughout the vehicle. Since the introduction of Integrated Circuits and Microprocessors to the industry the complexity of the functionality that they provide has been on an upward trend.

MISRA C and C++ With the introduction of firmware and software to the vehicle it has been important to ensure that software is developed to facilitate code safety, reliability and security. MISRA C and MISRA C++ exist as guidelines to assist in the development of software embedded systems for use in safety-related applications [28][29] [30]. In order to do this the guidelines present a subset of the languages such that developers minimise the number of non-definite behaviours that can occur in the language [31] with four classes of non-definite behaviours being: implementation-defined behaviour, locale-specific behaviour, undefined behaviour and unspecified behaviour [31]. These non-definite behaviours can lead to software crashes and ill-defined states which is highly undesirable for safety-critical applications. MISRA utilises Directives which are more broad aspects of compliance that is not necessarily discernible from the source code and Rules that are specific requirements and properties that source-code should follow. There are three categories of Rules and Directives: Mandatory - must comply with every guideline; Required - shall comply unless a formal deviation is recorded as part of the development process and Advisory - which are recommendations whose non-compliance should be documented but not necessarily through a formal deviation [31].

MISRA C and C++ exist to ensure that code that is utilised within vehicles is not going to unintentionally end in an indeterminate state.

AUTOSAR The AUTOSAR Classic platform for all intents and purposes is a set of abstraction libraries that, amongst a wide range of automotive OEMs and suppliers, have been adopted for use in the industry in order to increase the re-usability of code and allow for compliant hardware solutions from a variety of different suppliers to be used interchangeably on the vehicle without changing to much of the underlying software [2]. Within the Classic platform, there are three predominate layers of abstraction: application, run-time environment (RTE) and basic software (BSW) [2]. The basic software abstraction layer allows for abstraction away from specific micro-controllers and ECUs enabling a developer to utilise the drivers of compliant hardware from the supplier to work directly with AUTOSAR functions. With studies into the benefits of using AUTOSAR being listed as Standardisation, Reuse, Interoperability, Improved communication [3] with the biggest downside of AUTOSAR being shown to be its complexity, initial investment, steep learning curve, term confusion and the risk of it becoming too abstract [3].

Service Orientated Architectures (SOA) and Publish/Subscribe AUTOSAR Adaptive, is a more recent development from the foundation looking at defining an opt-in standardised approach for adaptive applications which are interfaced as services or APIS [32]. Within the AUTOSAR adaptive framework the runtime environment dynamically links services and their various clients together during run time.

[33] presents a succint summary of the other characteristics of an SOA which include higher run-time flexibility, On-Board and Off-Board services can be linked together, extensions and updates are possible, communication is abstracted from software development, start-up and shutdown complexity is reduced and general code maintenance is easier. The weaknesses are obvious, low level system behaviour is harder to predict, more abstraction requires increased computational power and testing becomes increasingly complex.

3.4 In-Vehicle Networks

There are hundreds of different mechanisms for exchanging data over short distances such as LVDS, SPI, UART, I2C etc. It is highly likely that these technologies also exist on the vehicle but utilised not necessarily as networks but interfaces to standalone modules or components rather than for facilitating exchanges between ECUs.

Local Interconnect Network - LIN A Local Interconnect Network enables communications up to 20kbps at a bus length of up to 40 meters as specified in LIN revision 2.2 [34]. It supports upto 16 slaves with no bus-arbitration in a Master-Slave configuration to enable deterministic communication between the various nodes. In Figure 4 it is clearly seen that one node provides the master functionality.

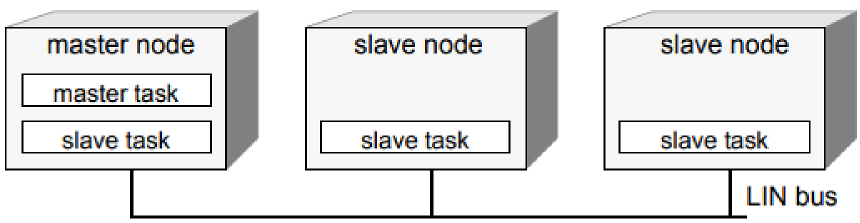


Figure 4: Figure showing a LIN bus with multiple nodes [34]

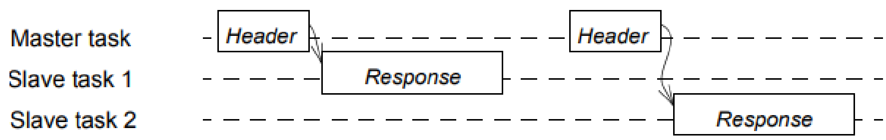


Figure 5: Diagram showing the communication flow of a LIN Master to the various Slaves [34]

Figure 5 shows that a LIN frame is constructed of a Header issued by the Master node and a response section transmitted by the slave node with the appropriate task (relating to the contents of the header section).

The header section contains a: break field, used to signal that a new frame is coming and can only be generated by a master node; a Sync byte field which must be detectable by all slave nodes and facilitates bit rate synchronisation and the Protected identifier field that is used to identify the task or the response required by the master node [34]. This can be seen in Figure 6 as can the break down of the response that is up to a maximum of 8 one byte sections before the checksum.

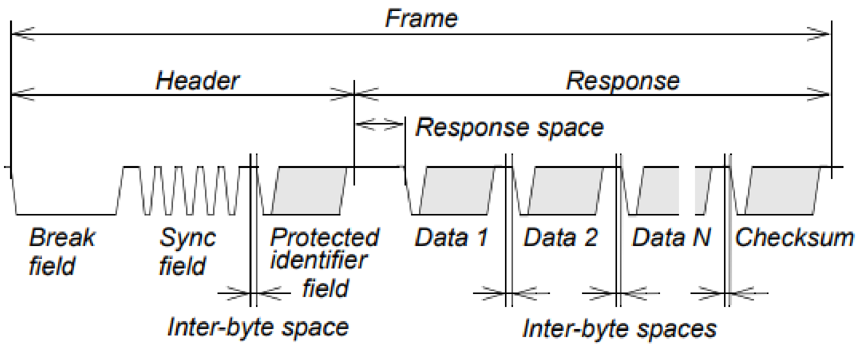


Figure 6: Diagram showing a LIN Frame broken down into Header and Response and other sub-fields [34]

The intended workflow for a LIN system can be seen in Figure 7 where Node capability files are utilised to to generate a LIN description file that contains within it the mapping from the abstract capabilities to the values for the master node to apply into the identifier field.

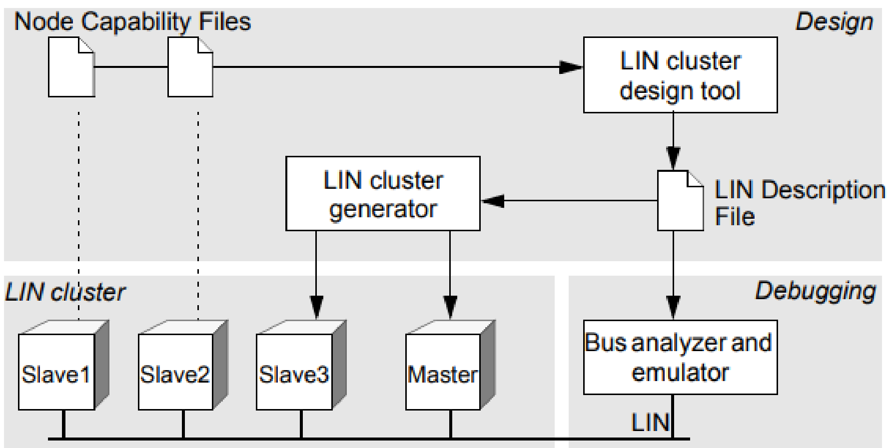


Figure 7: Intended Workflow for a LIN-based System [34]

This workflow allows for highly abstract system constructs to be mapped to the relatively slow speed, low complexity network with it typically being utilised to provide low-cost network functionality for components such as doors and seats [35] [36].

Controller Area Networks - CAN Controller Area Networks are a popular choice for automotive networks with a wide range of configurable options for an automotive network engineer to utilise to deliver features upon. In Figure 8 it is easy to identify where in the Open Systems Interconnection model the CAN standards fit, they provide a Physical layer, single twisted pair on a bus-topology, a priority based arbitration process, as well as interfaces that enable abstraction of the data link layer. Traditionally, CAN was limited to a bit-rate of 1Mbps over a range of in it's high speed mode [37]. Low-speed, fault-tolerant CAN specifies transmission rates above 40kbps up to 125kbps over a 40m segment [38]. CAN-FD a relatively recent addition to the CAN family was designed to enable interoperability with older standards, in most cases, by changing the bitrate of communication in the data segments of the frame [39].

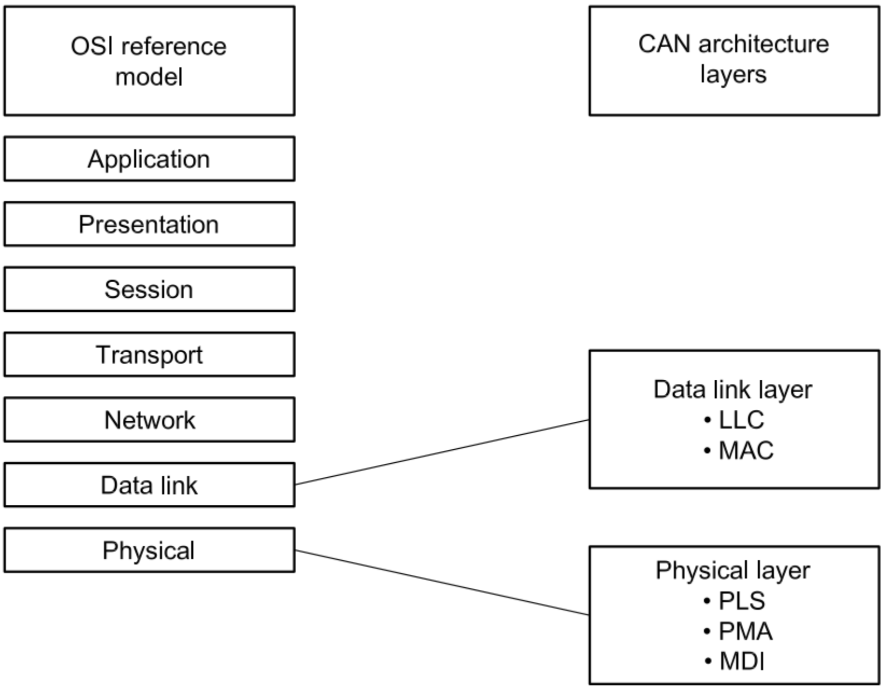


Figure 8: Diagram showing the various elements of the OSI model that the CAN standards look to address

In Figures 9 show the order in which frames are transmitted starting with the dominant Start of Frame bit. The standard transmits the Most Significant Bit (MSB) first meaning that for each byte of the payload, the bits are transferred from bit 7 down to bit 0.

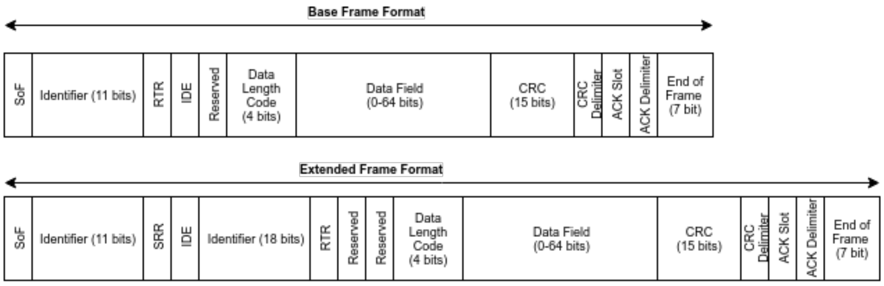


Figure 9: Controller Area Network Classic Base and Extended Frame frame format [40]

The information contained within all CAN frames is the Start of Frame, the message identifier, the data length code, the data field, the Cyclic Redundancy Check (CRC), the Acknowledgement bit and the End-of-Frame. In a classic, un-extended frame the Identifier is 11 bits long and is utilised in the arbitration process by which nodes stop attempting to transmit if their current identifier bit changes state whilst the bus remains in the original state. Extended frames can be sent by turning the identifier extension bit recessive (1) which then introduces and additional 18bits of identifier but with the same restriction on payload size.

CAN-FD, in compatibility mode, differs very little from the classic or extended frames [41] introducing a new parameters called the Bit Rate Switch (BRS) that if transmitted dominant the bit rate is not switched to the higher speed Data Phase bit rate otherwise the system transitions to the higher bitrate. This higher bitrate is derived from a synchronisation process in which

With quantitative analysis showing that CAN-FD provides a significant improvement over CAN, in terms of it's ability to handle scenarios with high bus-utilisation, due to its ability to utilise up 64 Bytes per frame and have a higher-bit rate, around 8Mbps in order to , during the payload period, [42] suggest that CAN-FD is likely to become increasingly common in order to address the challenges associated with the increasingly complex system architectures.

FlexRay FlexRay is a protocol that offers a system designer a broad range of communication options including a single channel at 2.5, 5 or 10Mbps, dual independent channels at the same bit rate allowing for up to 20Mbps, bonding the two channels together for redundant communication for higher throughput or increased availability [43].

In addition to such a wide range of bitrates and channel options, FlexRay supports a wide range of network topologies as shown in Figure 10. Flexray explicitly supports Point-to Point, linear-bus, passive and active start networks as well as hybrid combinations of the aforementioned.

This wide range of options was an important aspect of the networking standard as it aimed to provide system engineers with the widest possible range

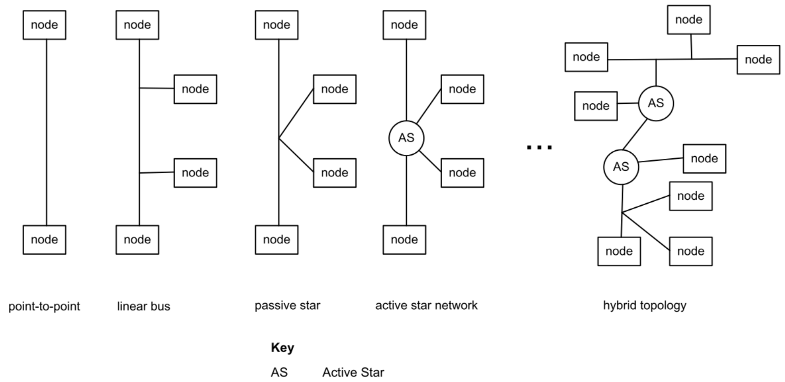


Figure 10: Figure showing the range of network topologies supported by a FlexRay network architecture [43]

of options when implementing the technology. The FlexRay frame format has three segments the header, payload and trailer segment.

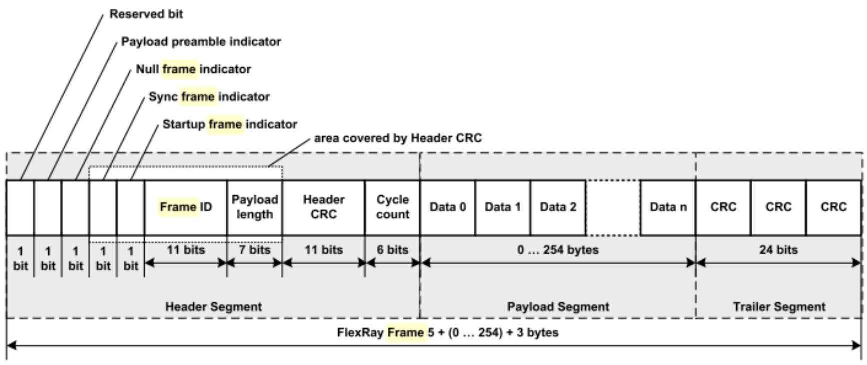


Figure 11: FlexRay Frame format[44]

As depicted in Figure 11, the header segment consists of 5 bytes which contain elements such as the sync frame indicator, startup frame indicator, frame ID, the payload length, the CRC for the header information and the cycle count. The Payload segment is between 0 and 254 bytes wide with each bytes being labelled incrementally from the first byte, which starts at 0. Frames that are transmitted in the static segment may utilise the first 12 bytes of the payload as a network management vector but this is indicated with the inclusion of the payload preamble indicator in the frame header. The first two bytes of the payload segment, for frames transmitted in the dynamic segment, is for the message ID which can be utilised by receiving nodes to filter frames intended for them. The final segment of a FlexRay frame is the trailer segment that contains the CRC over all of the contents of the header segment and payload segment.

FlexRay networks are governed by a communication cycle which represents the media access scheme for the standard. The timing hierarchy shown in Figure 12 shows that within a communication cycle, a variable maintained by all nodes on the network and included within the frame header, there are four segments: the static segment, dynamic segment, the symbol window and the network idle time. Access to the static segment is organised through a Time Division Multiple Access (TDMA) scheme into static slots, the dynamic segment is broken down into minislots whose beginning and end are aligned with macroticks, the symbol window is used to transmit a symbol onto the network and the Network Idle time is used to end a communication cycle [44].

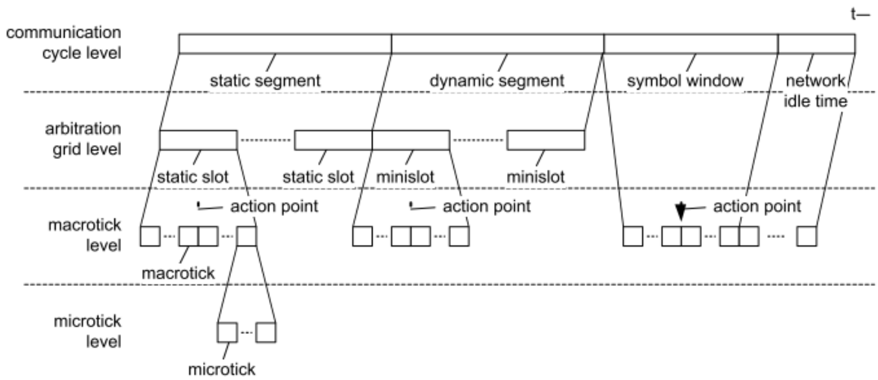


Figure 12: FlexRay timing hierarchy within a single communication cycle [44]

Every communication cycle is executed periodically with a constant number of macroticks and they are numbered from 0 to a maximum number. Access to the medium within the static segment and dynamic segment is based upon the assignment of frame identifiers, with every cycle containing a static segment which is a configurable number of macroticks wide. The dynamic segment does not allow sync frames, startup frames or null frames and the arbitration process utilises minislots and a per-channel counter to allow variably sized frames to compete for access to the available number of slots in the segment [44].

In addition to the wide range of topologies and larger payload sizes than that of LIN or CAN, FlexRay provides the developer with an integrated solution to time-synchronisation, which utilises information within the frame header specifically, the sync frame indicator and the startup frame indicator (depending upon the network configuration), to enable clock drift to be mitigated and cycle numbers to be synchronised. Clock drift and cycle numbers are referred to in the Data link layer specification of the FlexRay standard as Offset (phase) differences and Rate (frequency) differences with rate correction being performed over the entire cycle and offset correction being performed during the Network Idle Time (NIT)during the odd communication cycles. This clock synchronisation can be broken down into three different cluster types: TT-D, TT-E and TT-L. TT-D clusters require that at least one node is set to transmit sync frames and act as a cold-start node that is used for the rest of the network to startup with. TT-E clusters have only coldstart nodes and non-sync nodes with only one node to provide the sync bits in frames. TT-L clusters specify a single node that is configured as a sync node and a cold-start node. AUTOSAR specifies the worst case Time Synchronisation over FlexRay as an accuracy of $10\mu s$ [45].

Automotive Ethernet PHY Automotive Ethernet, typically refers to a category of Physical Layer (PHY) specifications from the 802.3 working group that meeting the stringent EMC requirements of the automotive industry. However, it is important to recognise that the OPEN Alliance (One-Pair EtherNet) Special Interest Group (SIG) an industry alliance attempting to encourage the wide scale adoption of Ethernet-based networks as the standard for the future of the automotive industry was instrumental in the development of 100BASE-T1 as BroadR-Reach [46]. The OPEN Alliance Special Interest Group has thus far worked upon 100BASE-T1, 1000BASE-T1 and 1000BASE-RH which utilise Unshielded Twisted Pair (UTP) and Plastic Optical Fibre (POF) as transmission media.

The IEEE has now standardised 100BASE-T1 [47] and 1000BASE-T1 [48] which both utilise the same physical media as their automotive networking competition, UTP, and 1000BASE-RH [49] which uses POF.

In order to facilitate the reduction from four-pairs in 1000BASE-T to the single-pair in 1000BASE-T1 the standards adopt full-duplex communication, which requires echo cancellation in order to reduce the cabling and adopt Pulse Amplitude Modulation 3 (PAM) in an effort to minimise bandwidth which leads to a reduction in EMI and lower cost cabling [50].

Ethernet Frames There are three types of Ethernet frame specified in 802.3, standard, Q-tagged and envelope [51] but all utilise the same frame format as shown in Figure 13.

Contained within the packet is the Preamble, Start of Frame delimiter (SFD), the Destination Address, the Source Address and the type of data contained within the MAC Client Data field. The size of the MAC Client Data field is 1500 for basic Ethernet Frames, 1504 for Q-tagged Frames ad 1982 for Envelope Frames. Q-tagged frames are a subset of Envelope Frames[51] with the minimum value for the MAC frame size as 64 Bytes.

The Ethernet standard as most recently defined in 802.3-2018 is highly complex while retaining its compatibility with the Open Systems Interconnection (OSI) model framework, as seen in Figure 14. The Data Link layer is broken down into Media Access Control (MAC) and the Locial Link Control (LLC) with numerous physical layer specifications providing a wide range of link speeds and various modular interfaces with standardised Medium Independent Interfaces (MII, GMII, XGMII).

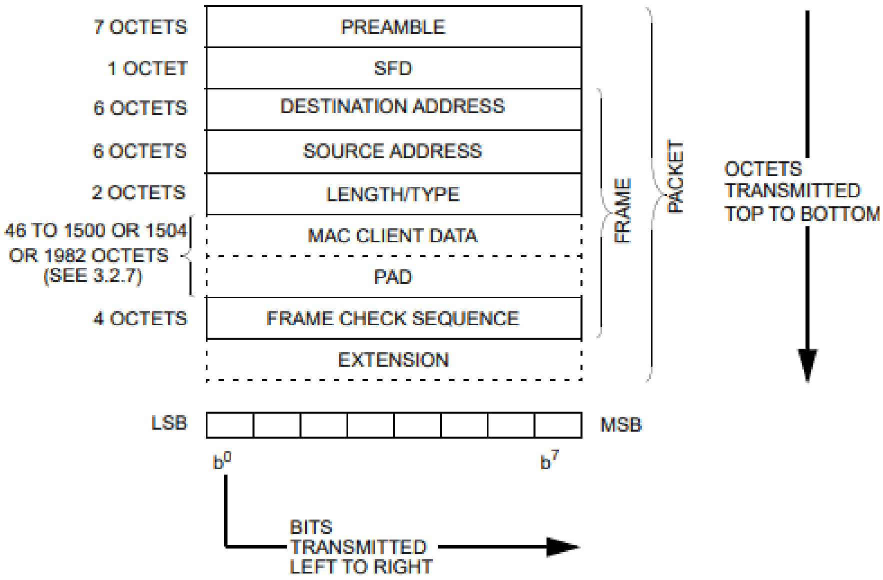


Figure 13: Diagram showing an Ethernet Frame contained within an Ethernet Packet [51]

10BASE-T1S Multi-drop Ethernet Historically the development trend from 802.3 has been toward higher and higher bitrate communication however, the automotive industry not only has a need for higher bitrate but for more cost effective solutions. 802.3cg [52] is exploring the solution to more cost effective Ethernet-based solutions by returning to shared-media or multi-drop segments.

In traditional shared-medium Ethernet PHYs, Carrier Sense Multiple Access with Collision Detection is utilised. Transmission is achieved by a station deferring for a quiet period on the media and then sends the message. If a collision is detected then all nodes involved in the collision intentionally continue broadcasting such that all nodes on the network are aware of the collision, after a random period of time, transmission is attempted again [51]. This approach is similar to the CAN arbitration process in that it introduces non-deterministic behaviours into the latency caused by this process. CAN has one advantage in this situation in that its arbitration process is still capable of transmitting the higher priority message whereas all stations involved in a collision have to try again to re-transmit.

PHY-Level Collision Avoidance (PLCA) [53] is part of 802.3, Clause 148 which attempts to improve the throughput, latency and fairness of CSMA/CD.

The working principle for this approach is to dynamically create transmit opportunities by assigning each PHY on the shared medium a unique node ID. The PLCA coordinator is the node whose $ID = 0$ and transmits a beacon signal which is utilised to synchronise the transmit opportunity timers. With every PLCA cycle containing one beacon and N+1 transmit opportunities

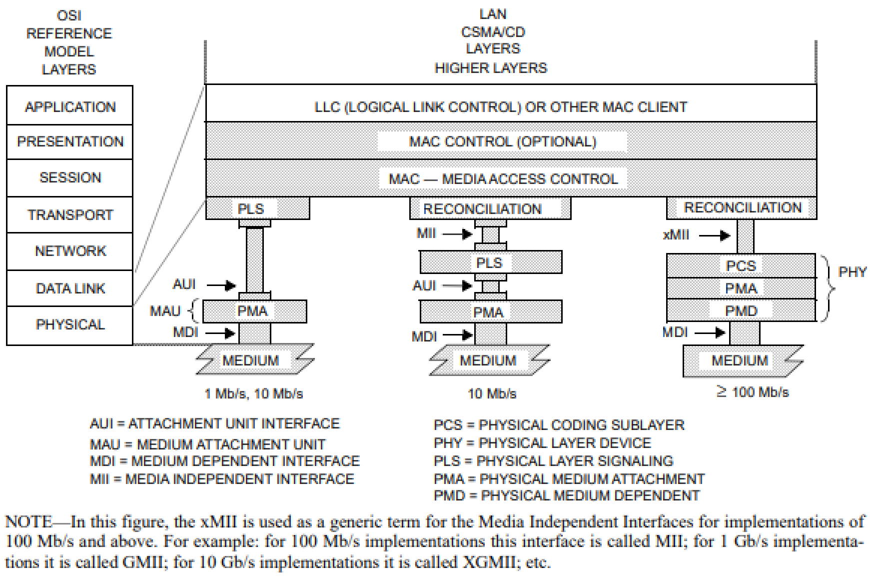


Figure 14: Diagram showing where the Ethernet standard 802.3-2018 fits into the Open Systems Interconnection Model [51]

which if they aren't utilised by their appropriate node within a configurable time period then the next period is started.

In a network with 8 nodes on the shared medium the maximum possible delay is 180 bits until the next transmission opportunity [53] which is approximately 18 microseconds and with the reduction in PHY complexity and no point-to-point wiring the automotive industry has a lower cost Ethernet solution that can be utilised for scenarios where CAN or FlexRay nodes could have one been utilised.

MultiGig Ethernet Taskforce The MultiGig Ethernet Taskforce [54] is looking to develop new PHYs that preserve compatibility with existing 802.3 standards, supporting full-duplex operation only, with a range of data rates including 2.5Gbps, 5Gbps and 10Gbps within an automotive environment. Objectives that were adopted by this task-force include the use of at least one type of automotive cabling such as UTP, STQ, STP,SPP, Coax or Twinax with support for the optional Power over Data Lines for appropriate media [54]. The efforts of the MultiGig taskforce fall inline with the NAV Alliance objectives [55] for applications that will easily saturate 1000BASE-T1 due to increases in sensor resolution. The NAV Alliance, as with the OPEN Alliance are likely to assist with the IEEE standardisation efforts by producing industry supported solutions that require little additional modification to be incorporated into the Ethernet standards.

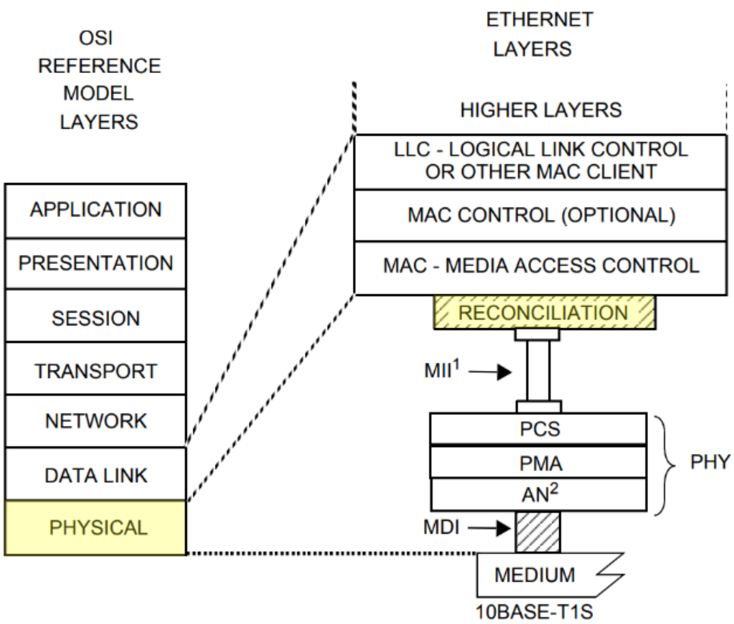


Figure 15: Figure showing where PHY-Level Collision Avoidance fits into the OSI Reference model and the Ethernet standard [53]

Time Sensitive Networking - TSN The Time Sensitive Networking (TSN) Working Group is the successor to the Audio Visual Bridging (AVB) Group in that the original project scope was to, through the use of 802.1Q [56] which relates to the operation of Bridges and Bridged Networks, improve the performance .

The original objective of the AVB Working Group was to improve the abilities of a switched Ethernet network requiring synchronisation, low-latency and reliability beyond that which is covered by the base specifications with the primary focus upon media such as audio and video streams. The Time Sensitive Networking group was created in 2012, out of the AVB group, to broaden its focus onto the mechanisms for time-sensitive transmission of data over Ethernet-based networks, including all forms of time-sensitive application.

There are three basic components to TSN: Time Synchronisation, Scheduling and Shaping and Path selection/reservation. Time Synchronisation for TSN is defined as part of 802.1AS which itself is a highly constrained version of Precision Time Protocol (IEEE1588) that enables sub-microsecond precision [57].

By having a single reference clock, Grand-master clock, on the network initiating the sync messages with all switches along the path being able to append their latency contribution the precision of the synchronisation is extremely high and is maintained throughout the network including on switches that Scheduling and Shaping requires.

An example of scheduling can be seen in the appropriately named Time

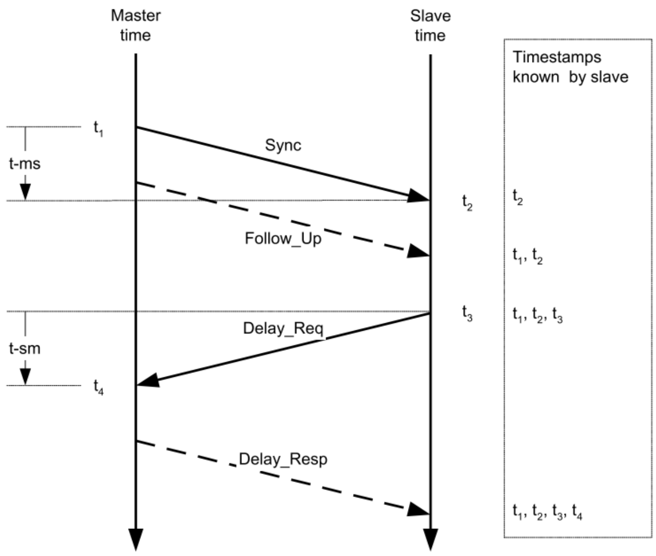


Figure 16: IEEE 1588 Basic Time Synchronisation Message

Aware Scheduler which enables precise synchronised transit of Ethernet frames at specific times, which when coupled with precise time sync allows for specific streams of data to be guaranteed access to the medium without need to queue. The Gate Control List and Transmission Gates are key elements of this Scheduler as shown in Figure 17.

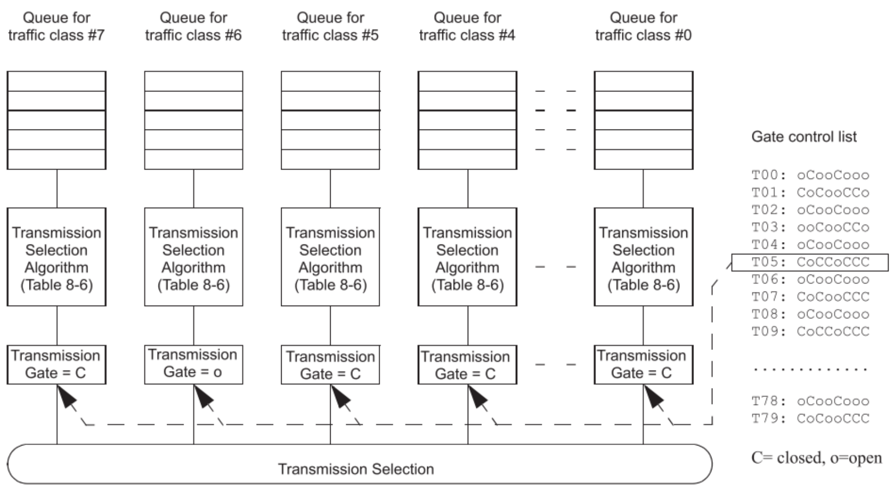


Figure 17: Transmission Selection with Gates per Egress Queue[56]

Traffic Shaping on the other-hand also appears in the Transmission Selection algorithm section an example of which is the Credit Based Shaper which is designed to reduce the impact of bursty traffic by implementing a token bucket system on the appropriate queue.

The final element of TSN standards is the Path Selection and Reservation which is covered in a variety of different ways 802.1CB [58] provides a mechanism for bridged networks with multiple paths to create redundant copies and transparently delete them in order to handle loss of connection.

3.5 Configuring Ethernet-based Networks for Time Sensitive Applications

Time Sensitive Networking standards cover a broad range of different technologies but, in general, the current academic efforts come clustered together in two distinct categories. The first category is in the form of bench-marking either with fixed traffic flows and variable configurations or variable traffic flows with variable configuration looking at trying to provide tools for optimisation or verification efforts. The second category explores the interactions with the various TSN standards with other aspects of the distributed network environment.

Benchmarking TSN standards for Verification [59] explores, from an network calculus perspective, the Worst-Case Latency of 802.1Qbv which is the Time Aware Scheduler and provides a variety of synthetic and realistic test-cases such as those shown in Figures 18 and 19.

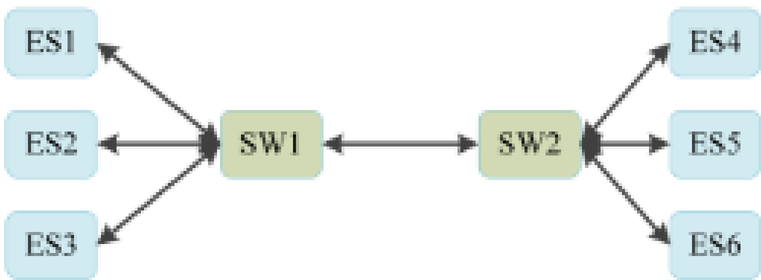


Figure 18: A simple synthetic test-case from [59]

[60] explores mechanisms to improve the Worst-Case latency in an Ethernet-AVB Network that is utilising an additional Stream Reservation class which in turn ends with a methodology that can be used to improve the worst-case delay of a network configured similarly. [61] also explores how to utilise the Credit Based Shaper, though identifies it as the Burst-Limiting shaper,

Interactions with TSN [62] explores how the OPC Unified Architecture (UA), a standard in use in the industrial automation domain, can be coupled together with TSN standards due to their compatible objectives of efficient and timely communication. In this paper, the authors were able to achieve

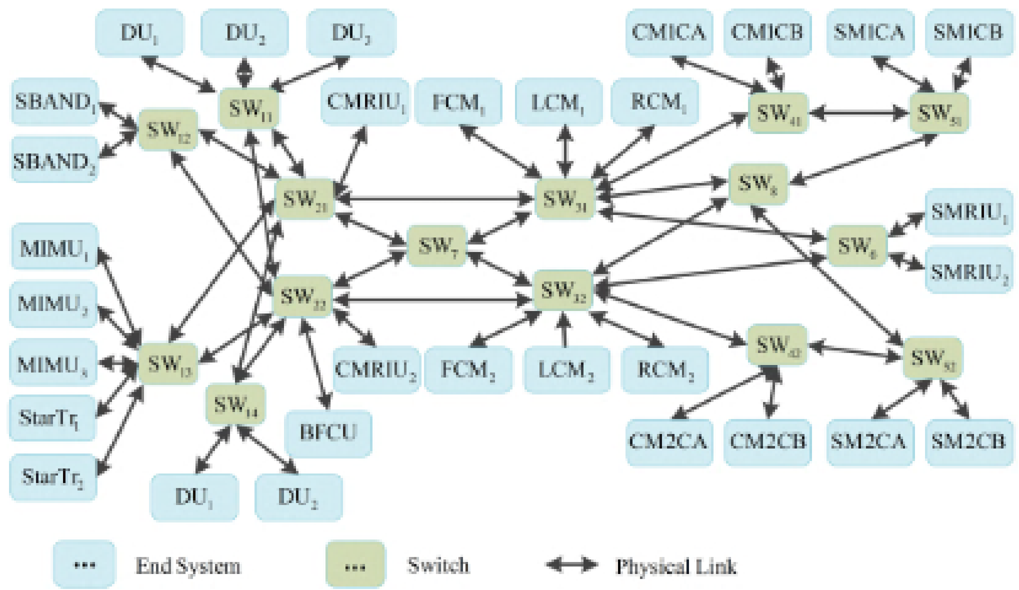


Figure 19: A more complex test-case from [59]

sub-millisecond publication intervals with minimal jitter by incorporating time synchronisation, 802.1AS and 802.1Qbv, the Time Aware Scheduler but has limited discussion on how exactly the network was configured.

[61] proposes a methodology for TSN networks to configure themselves which is strongly aligned with elements of the research project presented here. The proposed approach explores using the YANG data modelling language and NETCONF which is a mechanism that allows networks to be configured by messages from inside their own network, a necessity if self-configuration is desired. Again, an OPC-UA Publish/Subscribe system is adopted allowing information held within a centralised broker to be utilised to assist with the path reservation from TSN.

3.6 Summary

In this section, the document has introduced a wide range of technologies in various phases of Technology Readiness with Vehicle-to-Cloud technologies becoming almost mainstream and having wide success in being utilised to deliver a wide range of new features and insights into both driver and traffic behaviours that would have been infeasible without integration with cloud-based frameworks.

Embedded Software has been a mainstay in the automotive sector for a long time, this section decided to start with the Motor Industry Software Reliability Association's (MISRA) recommendations for embedded C and C++ moving onward to the industry-led interface standardisation, in the form of the AUTOSAR partnership, and how it enabled significantly more robust and modular software. With the introduction of Service Orientation which is a wider software development trend to improve code re-usability and modularity and how it ties into the developments within AUTOSAR to shift toward AUTOSAR Adaptive a platform with Remote Process Calls and Service Orientation at its heart. By exploring recent automotive software trends such as Over-the-Air updates and both Containerisation and Virtualisation which are technologies that are poised to significantly change vehicular architectures in the next product development cycles. Embedded Software is important but following on from Service Orientation, Remote Process Calls and Containerisation it became important to introduce the various in-vehicle network technologies exploring their physical media (PHY) and data frames as well as contrast automotive Ethernet in terms of the PHY and data frames. This section finishes with an exploration of the various standards that enable Ethernet to distinguish itself from more traditional automotive networks and explores the various approaches and techniques that have been used to deliver Time Sensitive Applications on Ethernet-based networks.

There are various themes arising throughout the industry, with the complexity of automotive software driving the shift toward technologies from other industries such as Service Orientation, Containerisation and Virtualisation which are intended to try reduce the coupling between hardware and software which in turn builds upon the significant contributions of the AUTOSAR Classic Platform with the Adaptive Platform almost representing the average industry acceptance of these trends.

Automotive Ethernet was originally pitched as a solution to the ever growing bandwidth requirements of vehicles with PHYs capable of supporting bitrates up to 100 times faster than a single FlexRay channel. With the introduction of a Shared-Medium solution that is designed to address the lower cost networked ECUs the other advantages of having a single unified network architecture become apparent with Ethernet frames being able to transition easily from a multi-drop segment onto a high speed backbone without translation will allow for more complex applications to be developed on the back of reduced latency and increased abstraction.

The general trend being shown here then is an increase in abstraction being leveraged to support the increasing complex vehicular architectures that are required to support Connected and Autonomous Vehicles.

4 Exposing Vehicular Data

Building upon the background information provided in both Section 1 and 3.2, Submission 1 explores the interface between On-Board and Off-Board systems due to the potential industrial impact that could be achieved, challenges faced by the research sponsor, Jaguar Land Rover, and as a good starting point for establishing a clear set of research objectives given the broad nature of the research domain.

As part of this process a comparison between three industries, that are highly motivated by communication efficiency and latency were compared as such On-Board Networking technologies from automotive, aviation and data centres. With the major disparity arising between the technologies in use in data centres being designed to carry data in abstract forms whereas a significant reduction of the abstraction being found in the automotive and aviation industries.

Obvious similarities between the Aviation industry and Automotive industry exist with the physical layer of choice between the twisted pair unidirectional (rx or tx only)interface of ARINC 429 [63] and the single twisted pair half-duplex interfaces of CAN, FlexRay and Automotive Ethernet. With more recent technologies AFDX [64] which can use an optical PHY similar to MOST but unlike most utilises an Ethernet compliant frame.

For Off-Board technologies the distinction between automotive, aviation and datacenters becomes significantly more emphasised with the distinguishing factor being the ability to develop specialised infrastructure.

Datacenters require significant infrastructure investment with extra provision for power and the network connection that is required for such an investment to actually provide its functionality [65]. Within both the aviation and automotive industry the primary product is obviously not a physical building and as such they both rely upon wireless communication technologies to operate with the differences here being the budget allocated for manufacturers to provide their functionality. In the aviation industry, the number of planes is a relatively small number but for safety applications such as Air Traffic Control [66] and Remote Diagnostics [67] special consideration around radio spectrum and dedicated infrastructure such as satellites and ground stations [68].

In order to build off such a broad research area an example was chosen in order to explore some of the strengths and weaknesses of the automotive approach to network technologies.

Telematics Telematics as presented by [69] is the blending of "telecommunication of codified information processed according to Logic". Working upon this definition it becomes apparent that applications discussed above with Remote Diagnostics [67] becomes an application of a telematics system with the logic being some digital version of 'this vehicle is not working as we expect, please help identify the reason'. The automotive industry has long wanted to develop an approach to reducing the costs associated with warranties with estimates of the global spend on warranty claims frequently suggested as falling between $45 billion and $50 billion [70], which incentivises reduction for any manufacturer because it can lead to a competitive advantage and an uptick in consumer perception of your products [71].

The aim at this stage of the project was to provide access to the complete vehicular dataset without foreknowledge of what subset of information is required. The condition around foreknowledge arose from the research sponsor, Jaguar Land Rover, who identified that being able to roll-out changes to the data being collected would enable studies to be more responsive to changing business needs and allow for the prioritisation of data deemed to be more 'valuable'.

The operating constraints provided were to treat the vehicle as a black-box as much as possible and to build atop existing telematics approaches that utilised the cellular networks available in Jaguar Land Rover's primary markets. The challenge introduced here is that the utilisation of a 3rd parties cellular network has an associated cost, with the average cost of 1GB of data being $6.66 in 2019 [72]. It is important to recognise that for a large manufacturer there are likely to be discounts or alternative approaches to paying consumer prices so this can be seen as an upper-bound value for the cost of collecting 1GB of data.

In order to see why having a mechanism to control the expenditure of vehicles with an integrated OEM managed telematics system, exploring the number of newly/first-time registered vehicles in the United Kingdom in 2018 was 2.9million [73], for a OEM with a 2% market share this translates to 58000 new vehicles in a year. It is unrealistic to model the data collected from a vehicle telematics platform to be equal to the total data throughput of the In-vehicle network. With current UK telematics suppliers capturing data such as: fuel utilisation, driver behaviour, acceleration, braking, speeding, location etc. [74] it becomes apparent that there is a huge range of possible frequencies, data encoding/representations and payload layouts that will have an impact upon the amount of data gathered. Fortunately the similarity between automotive platform and mobile phones here can take sample rates and data throughput examples from research into other types of application such as using mobile crowd sensing for traffic prediction as shown [75] which discusses a sample rate from each actor in the simulation every 300 seconds. Assuming that a developer is capable of containing all the relevant information within a single unfragmented IPv4 datagram of 576 Bytes[76], then each vehicle is generating 1.92 Bytes per second of operation. With a conservative daily driving time of 1 hour per day [77] this becomes 6.912KB per vehicle per day, which is 400.896MB per day for this singular application under very conservative estimates. Given that the number of applications that could utilise information from a vehicle in future is unknowable it becomes prudent to develop a system that can optimally deliver the information requested of it without duplication.

Having now discussed the constraints that need be applied to any proposed system the next stage was to explore what information can be gathered from the vehicle and how any proposal would go about gathering it. In this instance, this answer comes in the form of combining solutions from existing telematics solutions and knowledge around the most popular in-vehicle network technology, CAN. In order for ECUs to exchange information they must broadcast it onto the bus as described in Section 3.4 this transmission can be received by all devices connected to the bus (with the appropriate transceiver to decode the physical electrical signals into the bits and bytes that make up the data frame).

What does a result look like? Throughout the inception of the research project, it was apparent that the project sponsor had a desire for information from a vehicle. This was communicated indirectly through involvement with various stakeholders throughout the company looking at: combining accelerometer data and position to create a pothole detection service; preempting increases in warranty claims using remote telemetry and Diagnostic Trouble Codes; allowing a customer to access their drive history and numerous other more blue sky ideas. It became apparent with the sheer variety and range of data that might one day become a requirement of such an application and with the overlaps of data requirements that were apparent, such as location, that the more general problem was: Is it possible to design a data capture/reporting (telematics) system that provides the flexibility for arbitrary data requirements from future off-board applications whilst leveraging the contextual awareness for the specific vehicle to identify and report data that it considers important? The result of answering this question isn't necessarily a comparison between two approaches to identify which is better or worse, nor is it a measure of how performant with respect to less sophisticated approaches to vehicular data capture. Instead a result could be achieved through demonstration - present a system that can automatically report extraneous data, for some definition of extraneous, and allow the 'off-board'/cloud side be able to change what information would be captured.

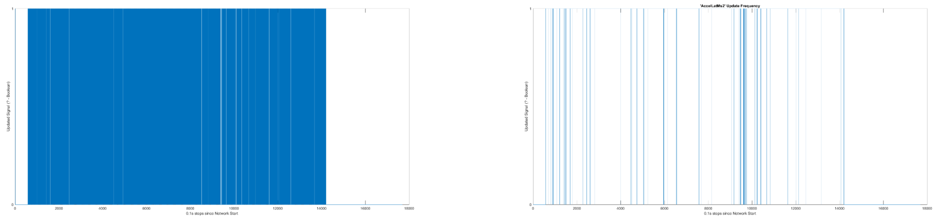
Acquiring telemetry data By utilising the telematics data captured from vehicles performing other research projects it was possible to gather a dataset of the signals transmitted on the various buses that were made available to the network. The telematics system utilised by the research project in question provided the researchers the ability to decode and record in a variety of file-formats the signal data, once parsed by utilising the manufacturer, model and potentially variant specific, '.dbc' file. With a variety of journeys from a representative sample of Jaguar Land Rover management during their commute, such as the one shown in Figure 20, provided a useful dataset to work off. This dataset was especially useful because it represented information that an OEM had collected for research, and as such represented something that could be useful in future applications. The primary limitation with the dataset is that the recordings are from the entire vehicular platform, which is useful for the development of the proposed solution but doesn't provide an insight into what the dataset was actually being utilised for which would have assisted in the development of a test scenario by providing a case-study.

Figure 20: Sample of a Journey being utilised to provide a reference dataset

A signal, is an individual variable that can be encoded, alongside others, into the data segment of a CAN frame. The layout of which as discussed above is encoded in a '.dbc' file. By utilising the captured information it became possible to fill in the gaps in signal value to explore the state, in terms of what the value of various signals are, of a hypothetical telematics system would be in at any given time.

Up until this point, the discussion has not included the latency of individual signals but it quickly became apparent that not all signals were updated equally which would potentially impact the capabilities of Cloud-based applications utilising the data, either being pushed at regular intervals, polled by the cloud application or transmitted when a new update is available.

This disparity in signal update frequency can be seen in Figures 21a and 21b and introduces the first firm limitation that the vehicle network architecture itself imposes on a generic V2C application framework. The ideal scenario from the cloud-side, independent of the mechanism that triggers the data exchange, is for every signal in the vehicle to have the lowest latency possible when it arrives at it's destination. Constructing packets within a hypothetical telematics unit to deliver the individual application's data requirements would



(a) Signal with a high update frequency (b) Signal with low update frequency

almost invariably invoke a sample and hold approach to signals at worst making some types of application impossible and at the very least require that for time-sensitive application to include the age of the signal relative to the others contained within the packet.

One solution to this approach is to publish changes in the state of signals as they arrive in order to minimise the latency contributions that could arise from storing and holding them until a request is made. If this solution is adopted naively, this would end up as bonding all of the various networks within the vehicle together and forwarding it to cloud in real-time, which would definitely create a strain on the various cellular network providers. Assuming 2*1Mbps and 2*500Kbps CAN buses handling all of the network traffic in the vehicle with 'good' bus utilisation of 80% [78] this would translate into a generation rate of approximately 4Mbps deliverable with the average 3G connection in the UK in 2014 [79], not that much data for an individual vehicle but if an approach like this was widely adopted amongst OEMs there would like be network congestion which reintroduces latency and lost-data problems. This data rate would also translate into the significant sum of approximately £10 per hour using the costings presented earlier in this section which when scaled up with a more GDPR compliant approach, rather than requiring consumers to consent to provide this data, that might turn into a 1% retention this would be infeasible.

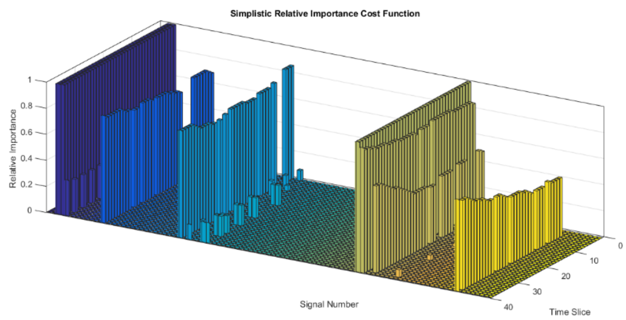


Figure 22: Relative Importance of the captured signal parameters from the reference dataset

I proposed a mechanism to prioritise signals based around the relative change in magnitude and signal frequency 22. This method, while not ideal due to it's bias toward high periodicity signals but this can be rectified by re-weighting the ratio between magnitude of the change and update frequency. This re-weighting could be used to improve the cost effectiveness of the transmission.

The solution presented here builds upon a distributed software development paradigm, often called Publish/Subscribe, a framework in which data publisher applications either exchange information with a broker which then forwards it to subscribing applications or each application maintains its own subscriber list that is managed through a separate mechanism. A broker mechanism was chosen for this research project due to it's relative simplicity in-comparison to a distributed subscription management interface and allows for an OEM to have direct overview over the operation of the broker rather than rely upon the subscription mechanism to remain secure and operational. In the situation where all information from a vehicle needs to be streamed into the cloud, a Publish/Subscribe mechanism is outperformed by raw translation and forwarding to a cloud-based destination due to the additional layer of abstraction.

Figure 23 is a visual representation of the system architecture developed to address the challenges of Vehicle to Cloud for generic applications.

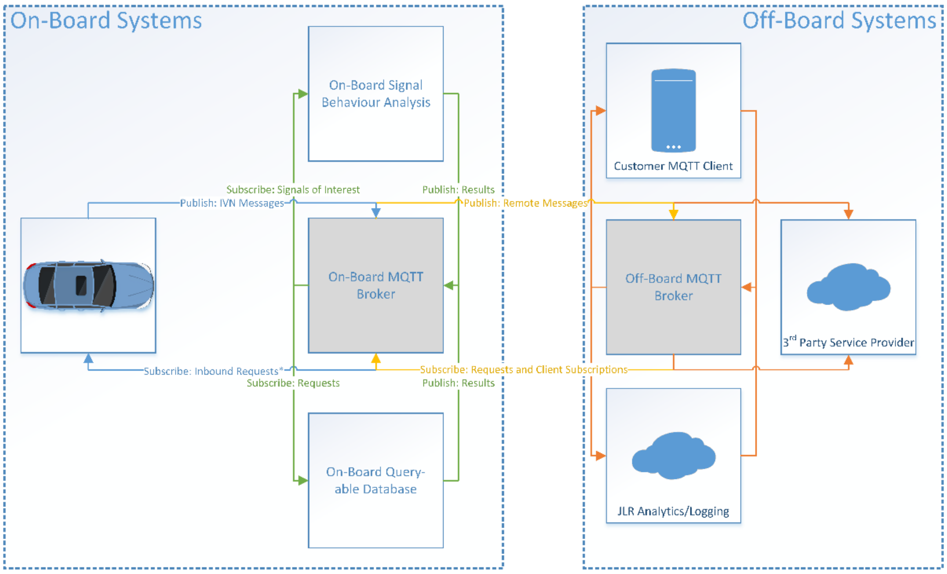


Figure 23: Proposed Vehicle Publish/Subscribe Architecture

In Figure 23 it is clear that the vehicle is being treated as a black box, where information is 'Published' to a broker that sits within the vehicle but separate from the existing network, this decision was taken, in part to enable this approach to be vehicle independent and in part to provide an additional barrier from attack. By limiting the number of possible ways this framework can interact with existing vehicle frames the smaller the attack surface.

With many commercial entrants into the Vehicle to Cloud space offering a Publish/Subscribe mechanism [80][81][82] the primary differentiator of the model above is the inclusion of multiple brokers. With a broker based On-Board the vehicle and a broker based Off-Board, in the cloud, it becomes possible to have applications running On-Board while the data connection between the two brokers is lost, restricted or limited in some manner. Similarly, having two brokers connected via the cellular connection provides a location for an optimisation function to be deployed allowing an OEM to utilise cost benefit analysis on either side of the link. Message topics, the label which subscribers inform the broker that they wish to receive, can be changed in real-time to heighten the importance if specific conditions are met, independently of the Off-Board systems. Similarly, if multiple applications communicating with the cloud broker are subscribed to specific signals you can prioritise signals in order to effectively gather data with a limited budget.

You can achieve somewhat similar behaviour with a single cloud-based broker but you run into challenges if the connection between the vehicle and the cloud is intermittent for any reason.

Another potential benefit of allowing applications to run both sides of the link is that On-Board storage could be utilised to facilitate requests for information that potentially occurred 'before' a subscription was issued from the cloud or determined to be important enough to be included in the broker synchronisation.

What was learnt? The automotive industry wishes to have access to vehicular data in order to deliver a variety of features/functions/applications that can be offered to consumers, provide improved information about product longevity, and for countless other reasons. The principal challenge is that without significant foresight into exactly what information is required in advance much of the value that could be generated is lost.

The proposed framework treats the vehicle as more than a dumb remote sensor looking to leverage the increasing computational resource of ECUs. To communicate the idea to the project sponsor, a simple demonstrator was built that presented a web-interface on the 'cloud-side' that enabled dynamic subscription management whilst an application designed to emulate a vehicle was able to recreate streams of vehicular signals run against an arbitrarily defined prioritisation function.

The question wasn't "what is the best way of prioritising information for transmission from the vehicle", however when considering the proposed approach there are several key limitations the first is that by only using the signal value and changes in frequency (which is metadata) the approach misses potentially relevant important signals for applications such as active threat monitoring. An expansion in this domain could be to look at the behaviours and relationships between signals, however just looking at relationships between arbitrary signal values is not necessarily indicative of an important event/state change. Deriving meaningful trigger/priority functions for specific applications would be a key area for future focus in this space.

Other limitations of the prioritisation approach is that there was no consideration to either the origin or destination of signals that were being received on the bus. This limitation emerges from the black-box approach to the underlying vehicle, by consuming data and only considering its value and meta-data, it becomes increasingly difficult to ascertain the impact of, and thus the importance of any changes to the overall system. An example of the importance of this is how a black-box telemetry system would behave under the reception of a significant spike in the accelerometer data. On its own, is this important? Is this change unexpected? Do we trust the authenticity of this signal? all these questions require additional understanding and considerations in the vehicular architecture.

With an exploration into how the current vehicular architectures can be exploited by telemetry system, principally the ability to sit passively on numerous buses and access information from across the vehicle, it becomes apparent that gathering signals as they arrive and constructing packets to be transmitted runs into the underlying fundamental properties of the signal packing process. At the opposite end of the spectrum, a solution that would translate all signals into packets and forward them onto a cloud application would be financially unfeasible, even if the networks could handle such an additional burden.

The middle ground, a Publish/Subscribe system with two brokers located at key points in the V2C application topology, presented above, aims to provide an OEM with the ability to deploy applications On-Board or Off-Board depending upon their current requirements with regard to resiliency against network coverage and limitations. With the brokers on either side of the cellular connection, On-Board and Off-Board, kept synchronised with a cost-optimisation function that can use contextual information only available on their respective sides such as: currently subscribed cloud-based applications and/or irregularities with an infrequently requested signal that could indicate a fault.

5 The Challenges Facing the Next Generation of In-Vehicle Network Architectures

Section 4 concludes with a paragraph titled 'What was learnt?', in which the challenges traditional In-Vehicle Networks have with providing the data to a Publish/Subscribe interface in a way that reflected the dynamic nature of Off-Board applications was described.

In this section, titled "The Challenges Facing the Next Generation of In-Vehicle Network Architectures" the project investigates these challenges with a detailed exploration of current in-vehicle network technologies in order to identify their origins and potential solutions.

With the technical details of Local Interconnect Networks (LIN), Controller Area Networks (CAN) and FlexRay having been presented in Section 3.4 it becomes apparent that with the peak bit rate of Flexray in a bonded dual channel mode being 20 Mbps (upto 10Mbps per channel) at the Physical Layer [43] and classic CAN frame formats supporting bitrates up to 1Mbps [40], Flexible Data Rate frames showing the bit rate as approximately 0.5Mbps during the Arbitration-Phase and 4Mbps in the data-phase [39] and LIN supporting a range between 1Kbps and 20Kbps [83]. It becomes apparent that for more recent automotive sensors, such as cameras (potentially operating at around 1Gbps (for RAW capture) [84] [85]) and LIDAR (a 16 channel automotive LIDAR unit from Velodyne operates at 8.6Mbps for single channel, 17.2Mbps for dual-return [86]), that the bandwidth of existing automotive applications is insufficient.

The simple solution to this problem is to not transmit the data collected by these sensors on the in-vehicle network, instead choosing to gather all of these sensors into a centralised location capable of performing all of the processing required.

The challenges and limitations of this approach include creating a single point of failure/access (SPOF/SPOF) and increasing the harness weight by having long shielded cable runs in the harness. ECU Consolidation is not necessarily a bad thing and it has numerous benefits including centralised system state monitoring, reduction in system weight, reduced part cost and improved upgradability [87] [88] [89].

Service Orientated Architectures (SOAs) are an increasingly common element in the development of automotive software with the introduction of AUTOSAR Adaptive [32] representing, at least in the author's mind, mainstream acceptance of the need for standardisation of Remote Process Calls through APIs and through services which are dynamically linked to clients during runtime which differs significantly from the traditional static compilation-time/configuration-time approach seen in AUTOSAR Classic [2].

Automotive Ethernet presents a direct solution to the challenges handling the high data-rate sensors that are appearing within the automotive ecosystem by providing PHY solutions with bitrates starting at 100Mbps [47], 1Gbps [48] and with current standards development exploring higher bit rates including 2.5Gbps, 5Gbps and 10Gbps [54], with these bitrates challenges associated with raw bitrate should be addressed for the automotive industry for a while to come.

The next challenge that Automotive Ethernet is looking to address is slower-speed connections with the standard 10BASE-T1S [52] which aims to introduce a multi-drop PHY to the range of automotive Ethernet solutions. Providing an option for ECUs that only require relatively low bitrate communication interfaces has numerous benefits including cost reduction for nodes and helps facilitate a holistic 'Full-Ethernet' vehicle with a unified data exchange format [90].

The remaining challenges to address for automotive Ethernet is that of managing frame latencies and Time Synchronisation across nodes on the network. The latter of these challenges is addressed, with the introduction of standards from the Time Sensitive Network working group, discussed above in Section 3.4, specifically focusing upon 802.1AS [57] and the time-sensitive standards within 802.1Q [56] which provides the mechanisms for time-synchronisation across time-domains, a constrained hardware facilitated variant/configuration of Precision Time Protocol (PTP), and requires that at least the switches be time-synchronised in order to behave as expected. Identifying approaches to manage frame latencies is the area of focus for the remainder of this document.

Why is Time Synchronisation important for TSN Standards? TSN Standards, in particular standards such as the Time Aware Scheduler (TAS) (802.1Qbv), provide tools for Ethernet switches to actively control traffic flow at specific times. For systems with a single switch, the internal clock of said switch is likely to be sufficient to enact the configured behaviours however with the introduction of multiple switches the challenge of 'how to ensure each switch is behaving as expected/configured at all times'. The solution to this challenge is Time Synchronisation which facilitates all switches inside a time domain to agree on their state (at startup) and to correct for their individual clock-drift to ensure that they stay in sync. For TSN this synchronisation is provided by 802.1AS.

What are the TSN parameters? For the purposes of this project, the relevant TSN standards exist within 802.1Q [56] with exceptions for Time Synchronisation, Frame Replication and Link Aggregation. Within 802.1Q the concept of tagged Ethernet frames is introduced with the ethertype, set to a value of 0x8100, of a frame being utilised to indicate to a compliant switch that the four bytes including containing the former ethertype, now called the TPID contain additional information. At the end of the four bytes the actual frame ethertype is handled as normal. The key parameters for TSN, within the Ethernet frames, exist within this tag: the Priority Code Point (PCP) which is 3 bits long; the Drop Eligible Indicator (DEI) which accounts for 1 bit and VLAN identifier (VID) which is 12 bits long.

At the most abstract level, the TSN parameters allow frames to be differentiated and therefore treated differently because of the values. As described in Section 3.4, queues of TSN compliant switches can be further broken down into 8 egress queues, one for each of the 8 possible PCP values. Each of these queues can have their own traffic shaper attached such as the Credit Based Shaper (CBS) in addition to having the Time Aware Scheduler (TAS) attached which only allows specific queues to be considered for transmission.

At the time of initial conception the Time Sensitive Networking Working Group responsible had recently published the Time Aware Scheduler. It was identified that there was some existing material available to the academic community about the performance of this new scheduler under synthetic behaviours but limited knowledge around the interactions of this new scheduler under other regimes.

In order to facilitate a range of novel scenarios and combinations of tools from the Time Sensitive Networking toolkit it was deemed that exploring and working with an older AVB standard, the Credit Based Shaper, and the most recent TSN scheduler, the Time Aware Scheduler, was taken forward.

What are, and how do Quality of Service requirements fit in? Without the Time Sensitive Networking standards, Ethernet-based networks operate purely on a FIFO basis with no ability to differentiate between frames. This inability to differentiate makes developing applications that have stringent timing requirements exceedingly difficult in a network with other applications.

These timing requirements are the Quality of Service requirements of the application. Applications such as Brake-by-Wire have stringent Quality of Service requirements with loop execution frequencies of around 50Hz [91] meaning a distributed closed loop would have at most 20 milliseconds to transmit information, perform the calculation and return a control signal before the next signal arrives.

The latency on a network depends upon the network technology in use, for example, CAN uses Carrier-Sense Multiple Access with Collision Detection and Arbitration on Message Priority (CSMA/CD+AMP) which introduces a source of latency from the arbitration process as described in Section 3.4. Ethernet that utilises a full-duplex link such as 100BASE-T1 or 1000BASE-T1 operates on a store-and-forward basis which introduces latencies at the switch due to a variety of factors as discussed in Section 3.4.

How best to utilise network technology specific solutions to deliver frames to their destinations on time to meet their Quality of Service requirements is the key task associated with developing the next generation of vehicular architecture.

When comparing the two queueing schemes, the Credit Based Shaper is designed to take bursty traffic and spread out, over a time period determined by a growth/fill rate and the size of the messages in the queue, the interval between frames. While this can be a desirable property, if this scheduling system gets applied to all traffic types it would spread out traffic of completely different application types that end up in the same traffic class rather than do this per-stream. The Time Aware Scheduler allows a network engineer to guarantee that a queue/traffic class is able to access the port at specific periodic intervals.

What was learnt? This section, acts as a springboard, connecting the findings from Section 4, which looked at exposing vehicular data to a generic cloud application framework, and the industry sponsor's interest in a new comer to the in-vehicle networking market, Automotive Ethernet. By exploring how In-Vehicle network technologies shape the vehicular architecture and looking to how automotive Ethernet could be utilised to address many of the limitations associated with existing solutions this section aimed to shape the questions that need be asked of an 'Full-Ethernet Vehicle'.

- How can Quality of Service (QoS) requirements be mapped to TSN parameters?
- Can this only be done at design-time or can it be constructed at run-time with known constraints?
- What is the trade off between these approaches?
- Can such a system meet start-up constraints set forth by Jaguar Land Rover's internal quality standards?
- Can other information within ECUs be utilised to optimise network performance?

In the next section, Section 6, this document explores the development of a Discrete Event Simulation (DES) framework to facilitate further exploration into automotive Ethernet and to provide a platform to address the questions raised here.

6 Simulating Time Sensitive Networks

In the end of the last section, Section 5, Automotive Ethernet was chosen as the focus for ongoing efforts in the research project due to in part, its relative novelty and industrial interest as a potential solution to the increasingly complex vehicular architectures that were becoming increasingly common in vehicles.

With numerous challenges facing the wider adoption of Ethernet for automotive applications it becomes apparent that having an experimental environment to propose and test solutions is important.

Simulation or Hardware Implementation With the need for an experimental environment clearly evident, the decision became choosing between an environment based around a specific hardware implementation or building atop models that behave as described within the various standards of interest for the research project in question. A hardware-based solution, a test-bench, has numerous strengths such as:

- commercial, validated implementations of the various standards.
- direct applicability and validity of results and measurements.
- easily disseminated and demonstrated to industry sponsors for increased impact (tangible).

The disadvantages of a hardware based solution include:

- a relatively immature development community.
- limited documentation.
- undiscovered hardware and firmware bugs.
- additional test equipment required.
- the logistical challenges associated with acquiring equipment as a non-OEM.

In comparison, a pure software simulated environment has strengths such as:

- scalability - the ability to test a wide variety of topologies with limited hardware dependence.
- parallelism - run multiple tests at the same time - Single Instruction Multiple Data (SIMD).
- Discrete Event Simulation for networks is a well established technique commercially and academically.

The weaknesses for a simulated environment are:

- limited real-world applicability of measurements without hardware validation.
- complexity of functionality required to be implemented.
- development and debugging time.
- wide-range of framework options.

The two most commonly utilised approaches are introduced above but other considerations were made in order to mitigate the weaknesses and build upon the strengths of the two approaches above. This hybrid approach looked to use another technology of interest to the automotive industry, virtualisation, and run multiple virtual ECUs with their own hardware interface connected via an existing TSN-compliant switch. The proposed architecture can be seen in Figure 24

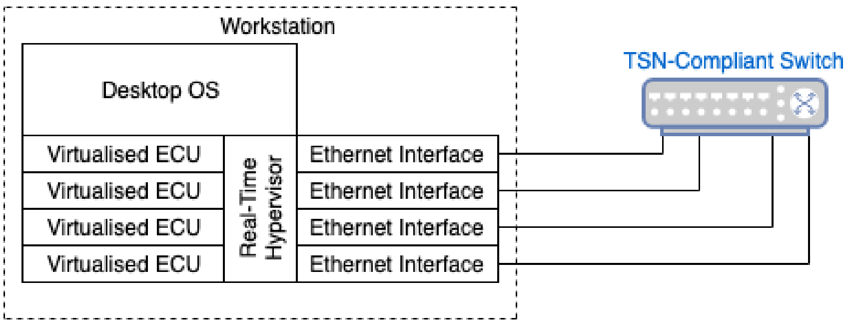


Figure 24: Proposed hybrid virtualised ECU network testing apparatus

The decision to work with a simulated environment was reached when the limitation of real-world applicability of measurements was somewhat mitigated by proposing a research pathway that focused upon working with problems whose potential solutions could not be realistically achieved with a hardware implementation such as high-dimensional configuration space searches and exploration of the impact of parameters that would not be configurable with any hardware option.

Simulation Capabilities Given that the decision was made to explore approaches that would be infeasible for a hardware testbench, the capabilities of a hypothetical simulated system are now of importance. From Section 5 there is a keen interest in exploring a 'Full-Ethernet' vehicle which means that we can, at least within the scope of this project, forego simulating CAN, FlexRay and LIN buses, however Ethernet comes with a wide range of PHYs including 100BASE-T1, 1000BASE-T1 which are both standardised and a variety of other PHYs that could be of interest to the automotive industry including a multi-drop PHY (10BASE-T1S) and a multi-gigabit PHY such as those proposed in the MultiGig Automotive IEEE working group [54]. The desire to provide a wide range of options when developing a simulated network means that an important capability of the simulated system will be flexibility and modularity allowing for new PHY modules to be created without impacting the behaviours of other modules directly. The project does not care expressly about the EMC implication of PHYs and as such does not need to provide EMC simulations for all of non-standard PHYs and their terminations which means that aside from the length of the wire the simulation need not require this information.

Time Sensitive Networks require time synchronisation and the method is covered in Section 3.4, however as this is a prerequisite of the standards operating correctly, the simulation needs only provide the ability to model clock drift and resync if explicitly required by an external stakeholder. 802.1Qav and 802.1Qbv are two key standards with regard to regulating traffic flow through a switch therefore any simulation must be able to effectively model their functionality. This means that for a simulated switch the ability to classify frames by their 802.1Q tags, assign frames to the correct egress queues and shape the traffic using the credit based shaper whilst simultaneously opening/closing the gates attached to each queue as specified within the Gate Control List, an important user configurable parameter.

Most importantly, the simulation environment needs to be able to precisely capture the times associated with important events in an Ethernet frame's lifecycle by determining the creation time, arrival time and the component of the latency associated with the queues is an important element when exploring system-wide performance.

With these system capabilities the simulation environment OMNeT++ was chosen over other competing discrete event simulation frameworks such as NS3 or NetSim. NetSim was not selected over the competing solutions because the academic version lacked the export of data in flexible data formats (.csv) and the ability for users to debug their own code [92]. NS3 and OMNeT++ are highly similar in out-of-the-box capabilities, both programmable and extensible in C++ and both can export data in a wide range of different formats depending upon the user's needs [93][94]. The primary differentiating factor between NS3 and OMNeT++ in the authors opinion is the INET framework [95] which provided a comprehensive implementation of existing modules following an OSI-like layered approach with clear documentation and class inheritance between modules allowing for easy integration of new standards and new PHYs.

Validating the Simulated Environment Given the various challenges associated with the development of either a hardware testbench or the hybrid solutions discussed above, validating the measurements for the simulated environment was not possible within the time frame of the research project.

However due to the importance of validating the accuracy of measurements captured by the simulation environment, an approach to validation has been presented.

In order to validate a simulation environment it is important to start with a simplistic test case that can be replicated within both hardware and simulation environments. It is important to note that a simple test-case is not necessarily the simplest possible setup but instead that there is the possibility of exploring a non-trivial scenarios. In the scenario presented in Figure 25, the asymmetry between the nodes allows for a validation experiment to explore not just the inter-frame gap but also the scenario where frames are dropped.

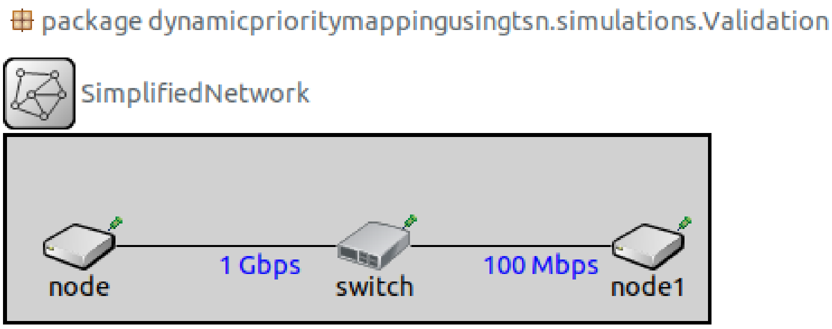


Figure 25: Example of a simplified network topology for Validation purposes

It is important to recognise that for validation to work the measurements must be equivalent between the setups which is non-trivial due to the latency contributions from the various stages between the leading edge of Ethernet frame arriving to the time that information is available within memory to have a timestamp applied. The mitigation to this approach is to take the simplest scenario and try to identify as many sources of constant latency contribution as possible before conducting more complex scenarios, this doesn't completely solve the problem as utilisation could be a latency contributor itself but it can either be used as an offset for results coming out of the simulated environment or as a quantifiable source of error in the results.

Strengths and Limitations of the Implementation The biggest limitations of the simulation library involve the lack of clock-synchronisation and lack of hardware validation. Clock synchronisation is a corner stone of the various Time Sensitive Networking standards and the explicit synchronisation mechanism is not covered in the simulated environment the original decision was taken as adequate clock-sync is a prerequisite to the correct function of the standards of interest however with further research it became increasingly obvious that an important area of research would have to include the scenario with bad clock synchronisation and the synchronisation process itself because it isn't instantaneous.

Hardware validation of the simulation would increase the confidence in the measurements being representative of a real system which is an important element but is in fact not the most important reason why hardware validation is a limitation. Hardware Validation in this particular scope allows for the behaviours of the various components to be compared against another source. The ideal scenario would be having multiple different vendor implementations to validate that the simulation is behaving as expected when compared to competing solutions.

The ability to perform a more hybrid validation of system behaviour and perhaps even facilitate more complex network configuration development strategies lies in both the cRealTimeScheduler and the ExtLowerEthernetInterface. The cRealTimeScheduler ties the simulation process to an external clock reference such that delays are achieved through sleeping the process at appropriate times, this would enable simulated, albeit basic in comparison to a more traditional ECU Hardware in the Loop (HiL) setup, ECUs to be modelled. The ExtLowerEthernetInterface class allows the operator of the simulation to specify specific hardware Ethernet interfaces to replace the simulated ones. For a simulation machine with multiple Ethernet interfaces, and the computational capability to schedule the traffic, this would enable the simulation to act as a virtual network for physical and modelled ECUs simultaneously.

In order to leverage these capabilities, an alternative between software-only and full hardware verification was proposed in the form of a hybrid validation approach. Given that verifying the behaviour of the simulated TSN switches under the configuration being proposed in this research would be a fundamental step before taking any practical results forward, it was proposed that a simple test case with two known-traffic generating ECUs were to leverage the ExtLowerEthernetInterface and pass this information into a commercial TSN switch. Comparison between the pure simulation and that of the switch would increase the confidence of other findings.

The fact the hybrid validation methodology exists as an option, even if it wasn't able to be demonstrated within the time frame, was as a response to industry trends that are continuing to leverage both Hardware and Software-in-the-Loop (SiL & HiL) testing to reduce development time.

What was learnt? What were the outcomes? Working with Time Sensitive Networking and Automotive Ethernet is still extremely difficult in comparison to working with other automotive network technologies such as CAN and LIN. This is to be expected given both the relative novelty of the various standards associated with TSN, further enhanced given the slow adoption of their precursory standards from AVB, and the significant leap in complexity/generalisability that Ethernet represents over industry-specific solutions.

In order to implement these standards various new modules were required to be created, many inheriting from and expanding upon existing modules/classes from the INET library, others did not have an existing analogue and as such were created from scratch. OMNeT++ can be considered a modular and inheritance based design tool with the Graphic User Interface allowing for interconnections between interfaces that inherit the same class in their code. Each module has a definition file (.ned) and more traditional C++ source code, in the case of this project if a new module needed to be written from scratch a new class was created to provide the functionality and if an existing parent class existed it was inherited and the appropriate methods were modified for the desired behaviour.

The TSN Classifier Module shown in Figure 26 is an example module that was created from scratch. It is the first component of the more abstract EgressQueue class that each port on the switch is connected to. It receives Ethernet frames, attempts to cast the base frame into that of a Q-tagged frame in order to read which queue/traffic class it should be routed to, or if the dynamic_cast fails, assigns it to the untagged queue.

Figure 26: TSN Classifier module

A limitation of the approach is that the tsnClassifier has no configurable properties that would likely exist in a more complete implementation, such as binding multiple priorities to certain Traffic Classes, in this implementation it is a one-to-one mapping.

For the queues themselves, there is little that needed to be changed from the base implementation of the FIFO queue module provided by INET. The deiQueue module inherits its interface definitions and class from that of the FIFO queue but modifies the insertion method to either prune existing frames from the queue that have the DEI boolean flag set to True or to drop an inbound frame iff the queue is full and it has the flag set to True.

Other base modules that were implemented from scratch include creditBasedFairQueueing, the timeAwareScheduler and the hardwareClock module. The creditBasedFairQueuing behaves as a leaky bucket exposing a runtime configurable parameter (idleSlope) that increments over time, by interacting with the hardwareClock module, it doesn't issue a frame until the amount of credit (in bits) is greater than the size of the first frame in the preceding queue module.

The timeAwareScheduler also interacts with the hardwareClock and opens and closes the queue based upon what slot in the switch configuration is open at that moment, it reads the internal switch configuration for the Gate Control List and opens and closes based upon Traffic Class and slot. This module does not support frame pre-emption and partial frame transmission which significantly impacts the theoretical performance and the applicability to the real-world as most implementations would include these options.

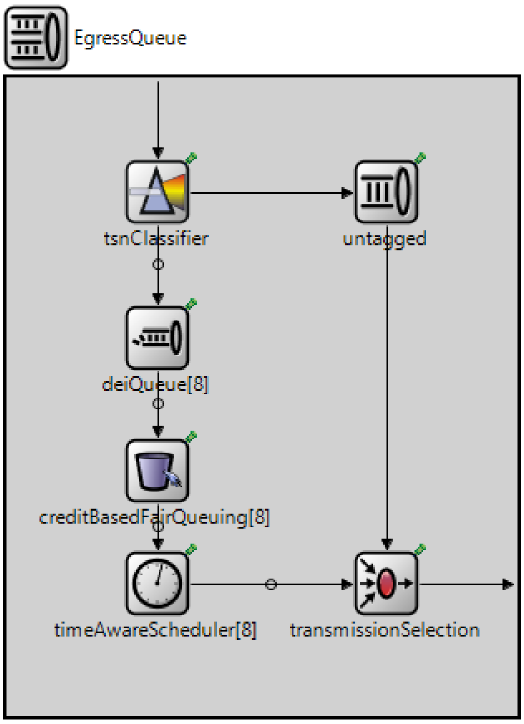
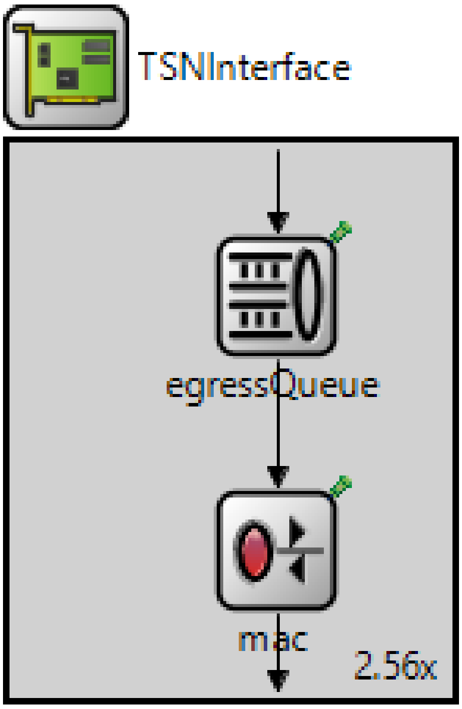


Figure 27: Example of the 8 Egress Queues, Credit Based Shaper and the Time Aware Scheduler in an Queue module

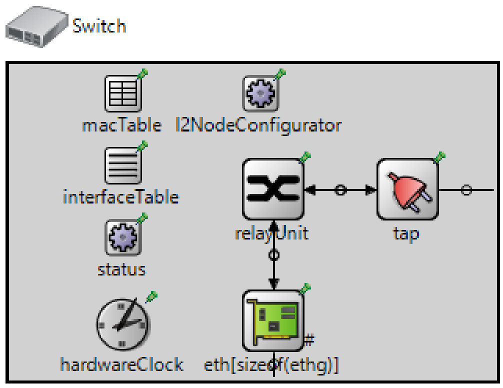
The hardwareClock module would ideally be an implementation of the Precision Time Protocol Annex F to accurately model the interactions of these components under high drift and the re-synchronisation processes. However, this module is lazy and just exposes a PTP timestamp structure that is calculated from a per-switch initial time and the simulation global time stamp. While this does mean that the re-synchronisation behaviours cannot be accurately reflected scenarios with high drift can be studied by artificially increasing the initial timestamp for individual switches.

The composite module shown in Figure 27 is appended to the outbound path for all ports into a composite module, or a module made up of other modules, called TSNInterface shown in Figure 28a. An array of TSNInterface replaces the array of more traditional EthInterface modules (that the new module inherits most properties from) that are found in the INET Ethernet switch.

The modules described above are but one outcome of this work. By exposing a variety of configuration parameters in the traffic generators and switch configurations in the simulation configuration the next step would be to model the functionality of a Time Sensitive Networking compliant switch with applications capable of transmitting and receiving 802.1Q tagged frames under a variety of different conditions with important parameters such as idleSlope, the Gate Control List and the Shared Medium TDMA configuration being exposed



(a) TSN Interface composite module



(b) Modified switch composite module

to allow for complex parameter sweeps. Another, less direct outcome, is a set of methods for testing the validity of the results and, when using more of the advanced scheduling and hardware interfacing capabilities of the OMNeT++ framework, provided a potential avenue for validating other 3rd party devices both software or hardware.

7 Modelling and Predicting Time Sensitive Network Behaviours for Dynamic Priority Allocation

The output of Section 6 was a Discrete Event Simulation (DES) environment capable of exploring the behaviours of an Ethernet-based network with Time Sensitive Networking Working Group standards applied to it.

As previously discussed the advantages of a simulated environment over a hardware test-bench include the ability to sample the configuration space, of the network configuration, in parallel rather than incrementally.

The research questions raised in Section 5, are:

- How can Quality of Service (QoS) requirements be mapped to TSN parameters?
- Can this only be done at design-time or can it be constructed at run-time with known constraints?
- What is the trade-off between these approaches?
- Can such a system meet start-up constraints set forth by Jaguar Land Rover's internal quality standards?
- Can other information within ECUs be utilised to optimise network performance?

This section, "Modelling and Predicting Time Sensitive Network Behaviours for Dynamic Priority Allocation", attempts to provide solutions to some of these questions.

Methodology The first question from Section 5 is "How can QoS requirements be mapped to TSN parameters?". Several approaches have been presented in Section 3.5 which look to take a known set of applications, whose quality of service requirements are known and wish to synthesise a switch configuration that is capable of meeting the deadlines that are set by the requirements. In order to identify the differences between these approaches and the approach presented here it is important to break down the types of assumption that can be made when looking at this question.

Quality of Service requirements do not exist in a vacuum and a key element that makes them meaningful is information about the applications themselves. At the simplest level, a quality of service requirement is useless without knowledge of where the source and destination of message that makes the application work are. The State of the Art approaches address the challenge of determining which applications need to use which TSN parameters and use the Quality of Service requirements as a 'fitness function'.

With the scope of this project exploring how to address the challenges of Vehicle to Cloud (V2C) applications that have best-case latencies dependent upon On-Board optimisation strategies (rather than aligned with their requirements) and not all potentially valuable data available for consumption, a Full-Ethernet vehicle with a Service Orientated Architecture in which applications are loaded and unloaded as needed during run-time is the working basis for the approach presented in this project.

Having a dynamic Service Orientated Architecture with dynamic allocation of applications makes traditional automotive assumptions such as the destination of a specific application's data significantly more complex. Fortunately even within a vehicle, there are still physical constraints around where raw data sources can originate and where some application outputs can sink their data these constraints are the fixed parts of a complex system of remote process calls and data streams that are expected to change during run time.

When restating the question, "How can QoS requirements be mapped to TSN parameters?" it becomes increasingly apparent that from the top-down, application-first perspective that the second question "Can this only be done at design-time or can it be constructed at run-time with known constraints?" are fundamentally linked together.

The approach presented here is to utilise the simulation environment and a model of a hypothetical Full-Ethernet vehicle to construct a model-utilising run-time parameters to make predictions around the latency of a frame given knowledge of the system state. In a perfect world, a perfectly accurate prediction would enable a node on the network to determine the most appropriate TSN parameters to assign to a particular frame in order to meet it's Quality of Service requirements. In reality however, it is impossible to construct a perfect prediction in principally because it does not have complete system knowledge but another factor is that it does not know the behaviour of every other node on the network that might be facing a similar decision.

This leads to the answer to the third question "What is the trade-off between these approaches?" to which the answer is determinism in the transmission of information between components of an application that might be distributed across the network. The approaches presented in Section 3.5 provide the guarantee that the upper bound of the latency is always less than the applications Quality of Service requirements. In our proposal no such statement can be made instead the probability of a frame missing its deadline can be computed.

The first stage of developing the latency prediction model was to construct a network architecture that, with support from Jaguar Land Rover network specialists contained all of the features they might expect from within a full-Ethernet vehicular architecture in future.

From Figure 29, which is the network architecture being utilised as the full-Ethernet vehicle, there are several key takeaways which are:

- 10Mbps Shared-Medium/Multi-Drop Ethernet for low cost ECUs will

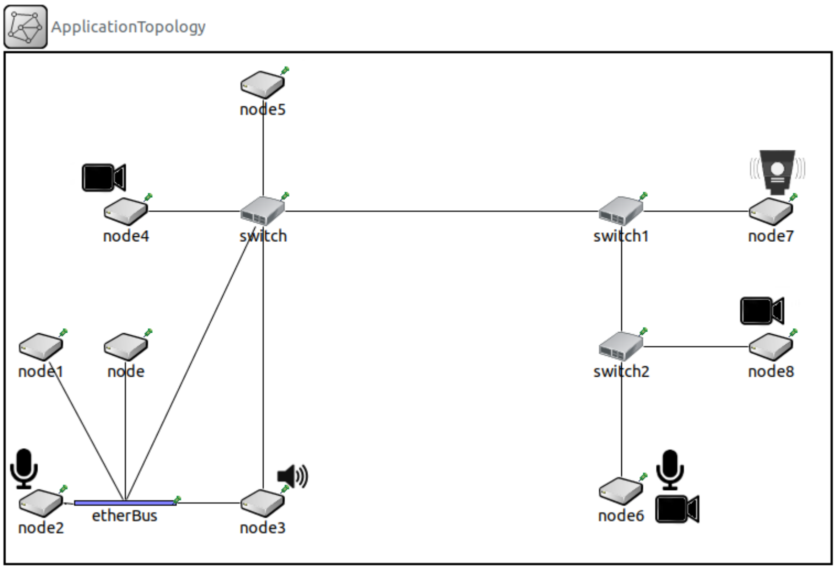


Figure 29: Simulated Network Architecture

help the transition away from other network technologies such as CAN and FlexRay.

- Connections that carry camera sensor data are at least 1Gbps.
- The Node that has a LIDAR sensor only requires a 100Mbps connection
- Microphones have a variety of uses from hands-free (node2) to Active Noise Cancellation (node6) with extremely variable QoS requirements.
- Nodes can and will have redundant links (node3).
- Switches will have a variety of different Medium Dependent Interfaces (MDIs) to allow all compatible segments to communicate directly.

The first stage of the proposed approach is to build a model of the best-case latency of the network to build all other predictions off because the End-to-End latency of an Ethernet frame has an absolute minimum value derived in part from the fundamental propagation properties of an electrical signal wire of length, L, which is the propagation delay and in part due to the speed in which a message of size, S, can get encoded onto the bus at bitrate, R, known as the transmission delay. Within the simulation environment those are the two calculable components of $T_{BestCase}$ in a test bench another source of latency Processing delay which relates to the time it takes for the various state machines within the switch to parse and correctly handle the frame before it is put into a queue.

$$T_{\text{Propagation Delay}} = \frac{L}{c} \tag{1}$$

$$T_{\text{Transmission Delay}} = \frac{S}{R} \tag{2}$$

With the best case latency contribution in its simplest form being:

$$T_{\text{Best Case}} = \sum_{i=0}^{N} T_{\text{Transmission Delay}i} + T_{\text{Propagation Delay}i} \tag{3}$$

The Best Case latency under the network presented should be symmetric and depend upon the path in question, N, which each segment having it's own unique Transmission and Propagation contribution. This value is eminently computable with it, when as described so far, being completely independent of the TSN standards.

In order to compute the latencies the approach adopted was to define the network topology as a undirected graph of each unique MAC address in the Local Area Network as seen in Figure 30.

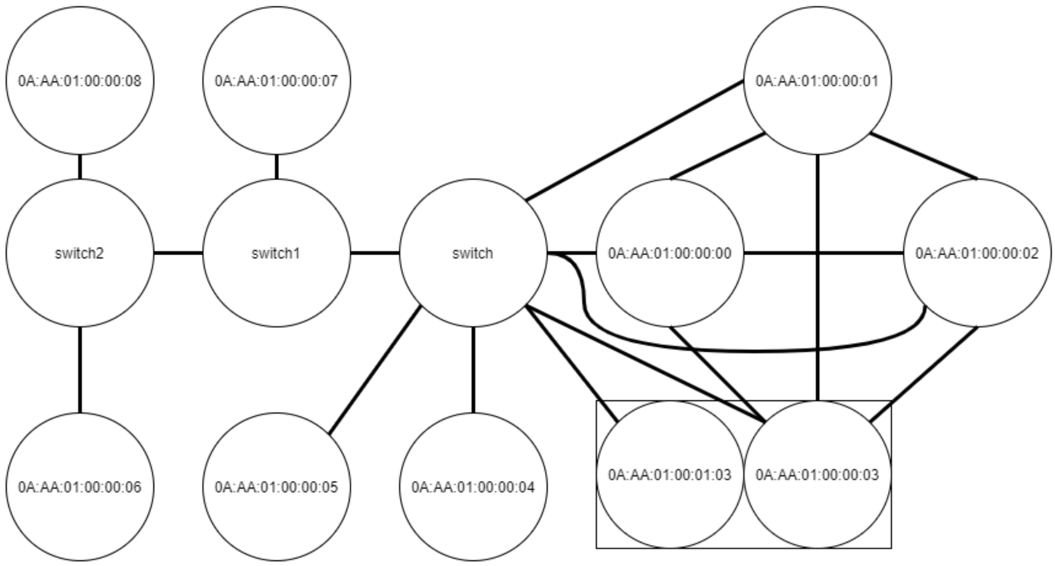


Figure 30: Unidirected Graph of the Network Architecture

When providing fixed link lengths and bitrates the best case latency from each MAC address to each other was calculated. The inclusion of a constant parameter, c, covers the unknown latency contribution that would be added when a message is moved from receiving buffer to the queuing buffer which would be determined by the specific hardware configuration in use by the switch.

Assuming that each route is not necessarily symmetric and that each route between devices is unique, both of which hold for a general network, the minimum latencies could be presented as a matrix.

When considering results the best case latency experienced by the devices on the shared medium will vary only by the propagation speed of the electrical signal (where the device is connected on the bus) and the message length which would be approximately 1.2 milliseconds with the aforementioned variations due to physical position being of the order of 0.1 microseconds (essentially negligible). For the routes that are point to point (only) the variation between best case latencies is due to the link speed (the primary component) and harness length. For the highest speed (multi-gig) links on the network, the best case latency for a 1522 byte frame is approximately 30 microseconds, with slower routes being closer to 400 microseconds. For routes that utilise both point-to-point and shared media links the best case performance is the sum of the two results, with the primary contributor to the minimum latency being the behaviour of the shared medium, in this case due to the significantly lower bitrate (10Mbps).

When taking these analytically calculated latencies you can then combine it with the Interaction Latency to get the Total End to End latency as shown below:

$$T_{E2E} = T_{\text{Best Case}} + T_{\text{Interaction}} \tag{4}$$

However utilising the simulation framework to explore the configuration space of a linear sample of frame sizes, alongside each unique permutation of Source and Destination was chosen for speed and scalability.

Now with a model for the latency of a frame, albeit one that doesn't account for interactions with any other traffic, the next stage is to use it to measure the difference between these predictions and an application topology whose behaviour is designed to reflect the type of traffic that might be found within a dynamic service orientated architecture in future.

The working assumptions of such an application topology is that there are some applications/modules that take sensor data and report it periodically to another location on the network. Other applications wait for the arrival of some information and then after some delay respond either to the source of the message or to another node on the network. Some applications such as camera sensors might stream information to a relatively fixed, within a drive cycle at least, location on the network perhaps to a graphically accelerated node capable of performing computer vision tasks.

In order to use the predictions three decision modes were developed. If you are to treat the assignment component of the network as an agent with the latency being a reward/fitness function, the following analogies arise. If the agent is acting randomly, it makes no use of the prediction for the latency of the Ethernet frame. This behaviour was chosen to provide a baseline for the performance of the other models, the ideal approach to bench-marking the performance of the proposed system would have been to have existing assignment strategies for comparison, but as noted earlier there was no relevant work on dynamic assignment in the literature.

For the other decision modes the actors were given two opposing behaviours. The first, the Greedy decision mode, would choose the frame parameters that would get a frame to its destination in the least amount of time, independent of the frame's Quality of Service requirements. In exact opposition to this was the Least-impact decision mode would aim to minimise the over provision of network resources by aiming to arrive 'Just in Time'.

Greedy Optimisation, which is Decision Mode 2, can be seen below:

$$(PCP, DEI) = \min_{T_{E2E}} T_{E2E}(Src, Dest, Size, PCP, DEI) \tag{5}$$

Decision Mode 3, Least-impact, can be seen below:

$$(PCP, DEI) = \min T_{QoS} - T_{E2E}(Src, Dest, Size, PCP, DEI) >= 0 \tag{6}$$

An alternative approach would have been to integrate this entire system into that of a reinforcement learning algorithm with the model trained using the complete dataset and then fine-tuning assignment behaviours based around the topology (electrical and application). Upon reflection this approach would have led to significantly more results within the projects time frame.

With these three decision modes introduced, the first stage is to start utilising the best-case latency predictions under these three different modes of decision making when running on the network with the, as defined above, application topology. In order to determine if the network is operating correctly, there are variety of different metrics that could be used. By looking at the number of dropped Frames and the number of missed deadlines which when both values are 0 would represent a correctly configured network, at least under the conditions defined.

Decision Mode 1 or Random Allocation was shown to have an approximate drop rate of 3% with no missed deadlines. In comparison both the Greedy and Least impact solutions had missed deadline of 47% and 56% respectively which includes dropped frames, which was expected behaviour given that without more state information the nodes would have continued with the same behaviour.

After collecting the results from this batch of simulation runs alongside their predicted Quality of Service parameters the next stage was to construct an improved model that utilised information about the state of the system, in the specific case of the network topology utilised in this research, a vector containing 81 elements. Within the network there are 81 queues whose state is reported every 0.1s to the broadcast address (FF:FF:FF:FF:FF:FF) which is captured and recorded alongside the frames arriving and being dropped throughout the network.

In order to increase the range of samples being fed into the model, results from Decision Mode 1 are combined with those of Decision Mode 2 and Decision Mode 3 in order to construct the model. This was done because otherwise the model would only be able to make accurate predictions of frame latencies that have been observed which in the case of Decision Mode 2 and 3 would be a single priority code point (PCP) and drop eligibility indicator (DEI).

Path	Minimum Observed Latency (ns)
Shared Medium Only	63466
Point to Point Only	1798
Combined	63472

Table 1: Minimum Observed Results from Stage 2 Simulations for across 3 classes of network path

The number of messages in the total dataset captured from sampling across each Priority Code can be seen in Figure 31.

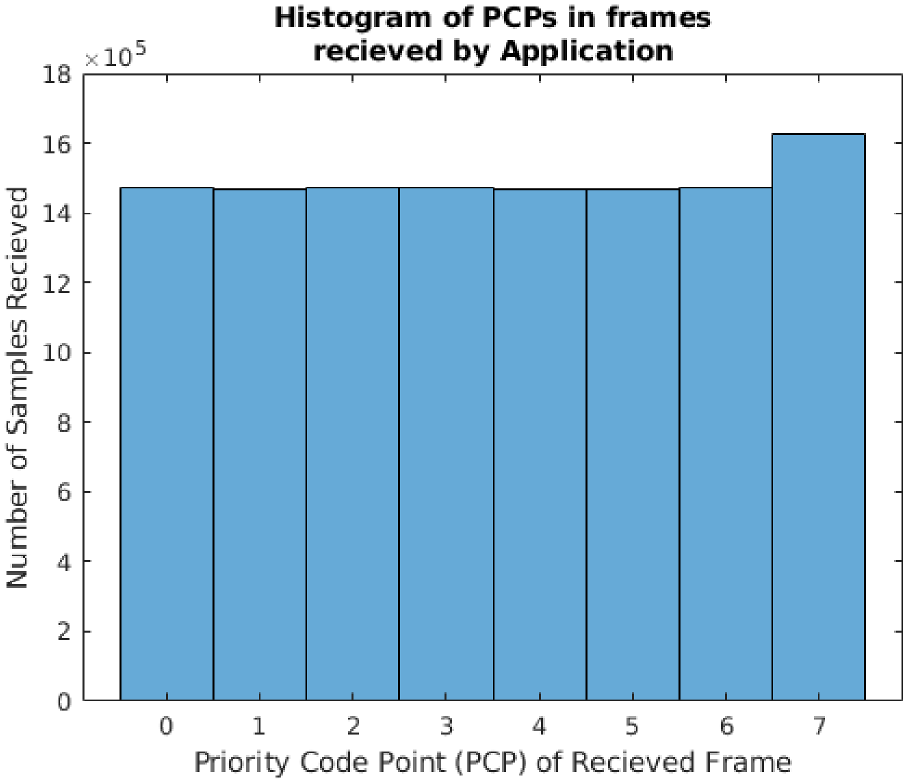


Figure 31: Graph showing the distribution of PCP codes for messages across the dataset captured

When exploring the minimum observed latency values there are three apparent classes of route: Shared Medium only, Point to Point and a combination. For the fastest point-to-point observation, it was generated (as expected) on a path with the highest average link speed on both sides of the switch.

When plotting the raw data from a shared medium node, it becomes apparent that the expected periodicity is being captured in the generated data.

For the remainder of the document, the graphs shown are representative of the three different classes, with tabluar data being the mean across each of

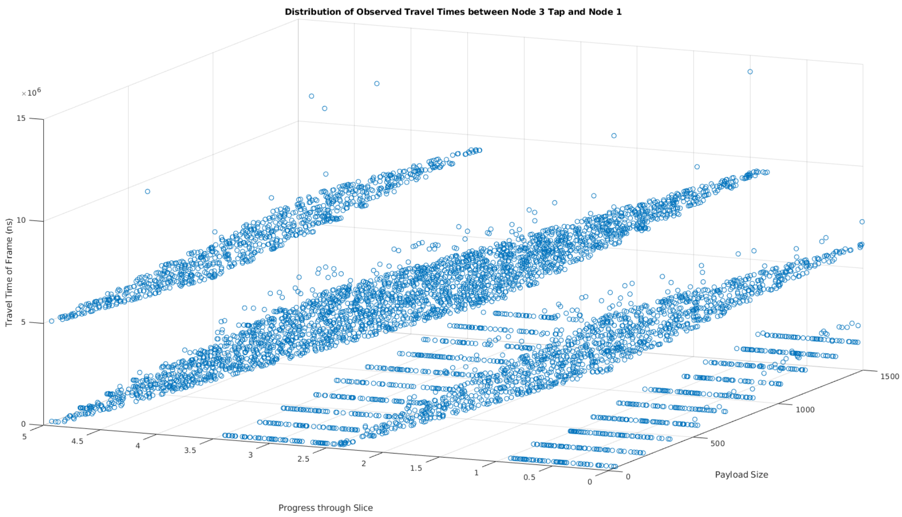


Figure 32: Graph showing the travel time of messages varied by time of arrival (in slice number) and payload size.

the results in that class.

Taking these best case samples and constructing a linear model using the basic parameters: payload size, time of transmission and priority code in order to make predictions about new messages.

Of the three classes of traffic, the linear model for best-case traffic fitted the best on the Point to Point as seen in Figure 33.

The fact that it wasn't able to fully predict the behaviour when the equations for calculating the value were linear suggests that the sampling rate was too high causing unexpected self-interactions for the messages during this phase of the testing.

For both the Shared Medium and Hybrid network samples, the residuals are relatively asymmetric. Again, this implies that the scenario constructed wasn't able to provide sufficient isolation between each transmission causing unexpected interactions to occur.

Path Type	Representative Sample Model Parameters	
	RMSE (ns)	R^2
Shared Medium Only	$2.6268 * 10^6$	0.0346
Point to Point Only	$1.9983 * 10^5$	0.4783
Combined	$2.6346 * 10^6$	0.0031

Table 2: Linear Model Fit Accuracy/Error Parameters for Representative Samples

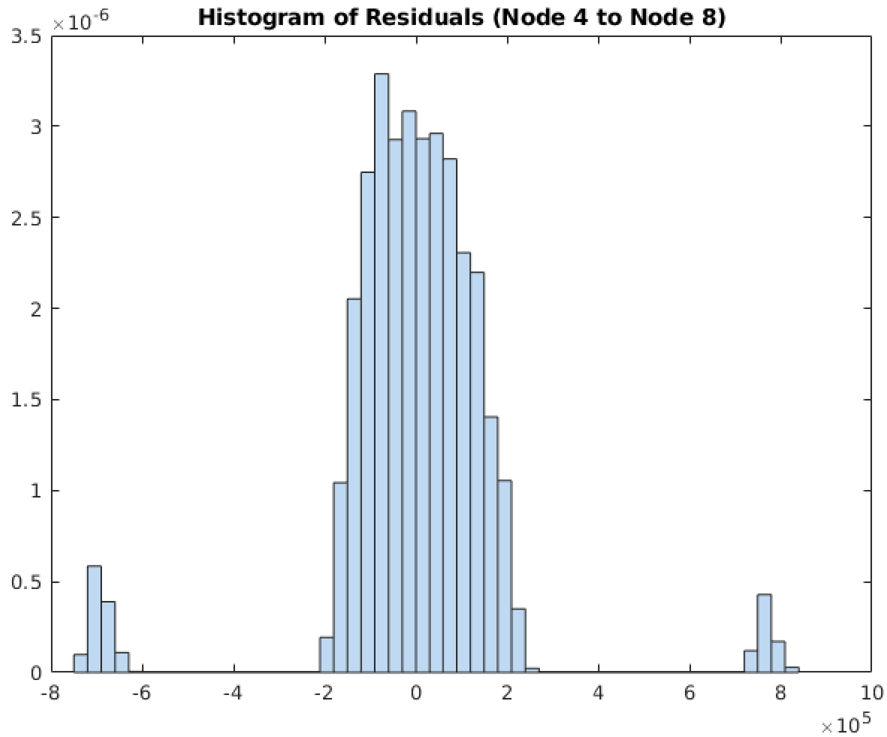
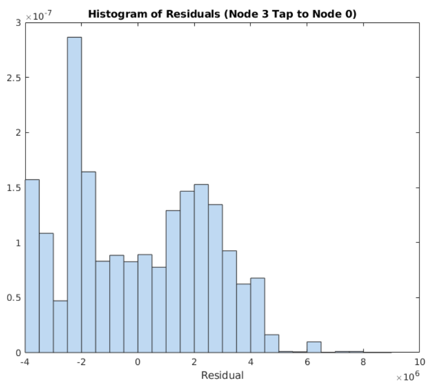


Figure 33: Residuals of Linear Model when fitted to Best Case traffic data for a Point-to-Point link

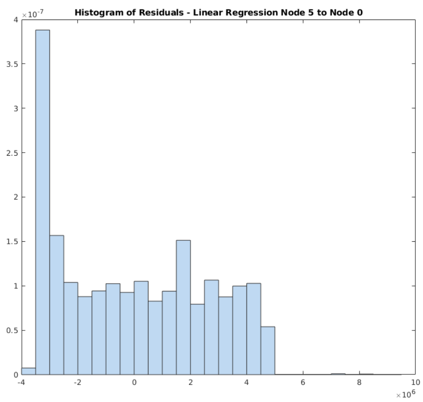
Unfortunately within the time frame allotted and given the number of samples required under a variety of different network states the latter phases of the process which include the introductory modelling of queue behaviour under specific application topologies wasn't achievable within the time period. Nor was it possible to quantify the exact performance this system would achieve under the conditions described which would in turn would have increased the value of this contribution by providing a reference against state of the art.

What was learnt? This Section addressed the questions raised within Section 5 and explore a methodology, using the simulation environment from Section 6 that enabled modelling a future automotive architecture that employs Service Orientation and dynamic loading and unloading of modules built atop a Full-Ethernet network, that addressed the questions raised.

The solutions within this section propose that mapping from a set of applications and their respective Quality of Service requirements can be handled with a minimisation function using a latency prediction for the application specific frames with predictions updated using system state vectors that can only be collected at run-time the system in exchange for determinism within the data-link and network layer to correspond with the non-deterministic na-



(a) Shared Medium Only Residuals



(b) Hybrid network path residuals

ture of the proposed future software architecture. Delivery of the proposed solution within the startup time specified in the Jaguar Land Rover is not possible, in part because the system needs to be constantly fed the system state and in part because the question assumes that there is a single optimal state to be reached which for this type of architecture is constantly varying due to changes in application state.

What the results represent and what do they mean? As discussed above, exploring the latent space of switch states for each path wasn't possible within the time frame. This latent space is essentially constructed from measurements of E2E latency of a frame through the network and then sampling the state of the switches (at the time it traversed them) to construct the space. The fundamental hypothesis is that for a specific network and application topology this latent space can be described with a model with low error terms. A fundamental question presented to conference was if you were using this type of predictor is there an error term that represents acceptable risk?

Figures 34a, 34b and 33 represent the residuals, difference between the measured sample and the prediction from a linear model based upon the limited sampling that achieved during the study. Importantly the linear model was chosen because it showed the best fit to the data being fed during construction. A more scientifically robust approach would be to use network calculus to model the individual components of latency, this is beginning to appear in the scientific literature and for the measurements that are not included in the calculus it an appropriate approach would be to treat these factors as Gaussian processes or kernels and then train a model with the initial assumption that these factors are correlated to all the factors from the network calculus.

The distribution of the residuals in Figure 33 is relatively symmetric, with variation in predictions around 3.5 microseconds nanoseconds, if we assume that the sample was representative of the entire latent space this would be an amazing result, it is however constrained by the lack of time available to search a much broader space.

Assuming that 3.5 microseconds was the maximum residual that the model had for Point to Point connections on the network this would mean that the Least Impact prediction could now be presented as:

$$(PCP, DEI) = \min T_{QoS} - (T_{E2E}(Src, Dest, Size, PCP, DEI) + e) >= 0 \quad (7)$$

With the maximum error term, e, being taken of 3.5 microseconds you can begin to rule out applications with a Quality of Service requirement greater than that. It is also important to consider that by grouping paths together by link speed, the behaviours of traffic on paths solely going through high link speed applications are being merged with those going through slower link speeds and that the error term for higher link speeds could be significantly reduced.

A perfect model would have no residual across the entire sample range as each latent space dimension would have a well defined impact upon the observed latency, the reality is that in a real system the perfect information approach that is being used to construct the latent space here is invalid.

Given that in this proposed approach the model has perfect information about the state of the other simulation, the next phase of this research, once there had been a more robust model constructed and more of the latent space explored with more traffic generation models, would be to quantify the change in the error term in the model with the introduction of information asymmetry. This would be achieved by introducing state update messages between each network component (ECU and Switch) about the behaviours they observed during the last period and then changing the model to leverage these assumptions in a store and forward manner, updating the parameter value if/when a new update arrives.

This is certain to reduce the efficacy of the model but quantification of the changes to the efficacy of the model under information asymmetry would be novel and help to further refine the classes of traffic this type of approach would be suited for inside of the automotive industry and more generally.

8 Discussion

Vehicles are becoming increasingly complex and with that complexity comes increased cost. The increase in cost comes from a variety of different factors ranging from obvious, new more expensive sensor technologies to the more subtle additional development and validation procedures associated with new technologies or methods.

With the telematics system proposed in Section 4 the broad concept of "any-data, anywhere" was a key motivator. The automotive industry has typically seen telematics systems as a tool to facilitate the collation of information from a fleet of vehicles so that they can glean never-before seen insight into their products. In order to stand apart from the more commercialised solutions that provide that functionality, the dual-broker system aims to provide enough abstraction such that the origin of the information, On-Board or Off-Board, does not matter.

By beginning with the vehicle as a black box, the required properties that the proposed location-invariant/hardware-agnostic system were more apparent. These properties typically boil down to Quality of Service requirements of the application: low latency or reliable sample capture or dependent upon stream or secure etc.

Many of these requirements have elements that can be addressed with network technologies that are off the vehicle, an example can be the reduced latency and increased bandwidth targets of 5G cellular networks which should assist with delivering these applications. However, as discussed in Section 5 a key contributing factor to some of these properties is the In-Vehicle Network itself. Vehicle networks have traditionally been designed to facilitate the exchange of information between ECUs where applications or features cannot operate within the same processor, either for compute limitations or physical proximity constraints. This communication 'as-needed' strategy leaves network engineers with the task of developing a configuration that meets only the requirements of the features/applications that need exchanging. Whilst this is by no means a limitation for current vehicular platform design, including having numerous benefits including an verifiable validation procedure, the ability to develop and deploy software whose features can be enabled/disabled at run time is a key aspect of the next generation of automotive architectures.

With software that is actively changing its behaviour, it becomes non-trivial to compute an ideal network configuration for all possible scenarios. One approach would be to utilise multiple configurations that can be loaded and unloaded at the same time as the system behaviour changes, this approach would lead to a vehicle's network behaviour changing as applications are started up and shutdown as needed but would require an engineer to have computed all possible configuration states in advance which risks becoming a complex task in of itself.

The approach presented in Section 7 looks to utilise a model of a network, under a single static switch configuration but with the variable application behaviours that define the type of software platform expected in future. Services that interface directly with specialised hardware to take measurements or accelerate computation can act as constraints for some networked applications, with the remainder providing a wide range of additional traffic. Through the use of a priority assignment scheme that utilises network latency predictions, the network behaviour can shift dynamically as the state of the switches, queue utilisation, changes.

A significant advantage of the proposed approach is the unspecified nature of the switch configuration. There needs to be one, but the approach looks to best utilise whatever configuration it is presented with, which means that it can work alongside the existing approaches presented in literature. By offloading safety-critical, time-sensitive streams to a more traditional allocation framework but utilising my approach for all other time-sensitive application streams would mitigate against the principle disadvantage of dynamic allocation. Given that a Dynamic Priority allocation scheme exploits the latency differentiation that arises from different TSN parameters, the question, "what metric could be used to assist in the generation of a switch configuration that is most suitable for a Dynamic Priority allocation scheme?" becomes increasingly important. Is there a way to synthesise a network switch configuration alongside the per-node network models that work together to provide a robust and stable to changes/perturbations in a manner that complements both approaches?

Dynamically allocating priorities based upon the network state has one key disadvantage over the more traditional switch-based configuration approaches in that the non-deterministic nature of the emergent behaviour can only specify the probability of failure rather than being able to provide a guarantee of no failure. This can be mitigated somewhat, as discussed in the previous paragraph, by utilising the approach for non safety-critical applications.

The simulation toolkit represents a set of tools for Jaguar Land Rover and other Automotive OEMs to potentially exploit in on-going research efforts around automotive Ethernet. While the area of research explored in this project has been highly varied and principally targeted at full-Ethernet vehicles, rather than an as the highly iterative process that defines the automotive product development cycle, there are a variety of potential use cases for an automotive Ethernet simulation framework at the current stage of development. There are numerous areas left unaddressed with this work that with some minor additions to the simulation framework, could facilitate the simulation of other automotive interfaces which in turn could be utilised to answer questions like, what exactly is the best strategy for taking the 8 byte CAN frame payload and 11 bit ID and converting it to an Ethernet frame? With the existing capabilities of the OMNeT++ framework, additional exploration of the proposed telematics system could be explored introducing more complex inter-vehicle communication frameworks in addition to existing cellular technologies.

There are several key strands of follow-up work, building up from the contributions of this work, that can be used to provide increased confidence in the results and methods proposed. The first of these strands of ongoing work would be hardware validation of the simulation framework. The second strand would explore the integration of other vehicular network technologies into the model, such as CAN and FlexRay, in order to improve the short/medium-term applicability to current automotive questions with regard to the gradual adoption/development of automotive Ethernet in their products. With many complex driving factors behind the decisions determining the component selections of vehicles, it is highly likely that traditional network technologies will continue to exist, which in turn highlights emphasises the need for a focus on integration and hybrid network architectures.

9 Conclusion

To conclude, this document provides an overview of the various innovative contributions from one potential pathway of an Engineering Doctoral research project titled "On-Board and Off-Board Data Platforms".

With the broad nature of the project title, the first portfolio contribution explored Vehicle to Cloud (V2C) applications and proposed a generic Publish/Subscribe dual-broker framework that enables a vehicle (On-Board) to identify information to be reported to the cloud and s a mechanism to take the signal requirements of all applications in the cloud (Off-Board) to be collected then distributed to their respective subscribers. By recognising that exchanging all information that could potentially be consumed by the vehicle would introduce significant costs to the manufacturer the proposal introduced a cost optimisation function that, by default, kept On-Board and Off-Board applications in a desynchronised state only exchanging information deemed of interest to On-Board systems and when sufficient clients required the signal from Off-Board. This cost optimisation function provides a tool to manage the operational challenges, in particular the cost of utilising an external cellular network, to the OEM.

There is little novelty in the treatment of a vehicle as a dumb endpoint in a fleet, however by introducing the concept that a vehicle may push relevant information you begin to build the framework for applications more complex than telemetry. Some examples where this two-way relationship could be of significant interest is in digital twinning, which is considered briefly in the portfolio around preempting increases in warranty claims, but fundamentally a digital twin is an enabling tool toward more data-driven design methodologies. Another application, that runs into the same challenges that led to the exploration of Automotive Ethernet and Service Orientated Architectures, is part of the Active Threat Monitoring that recent UNECE regulation has introduced for OEMS. A vehicle is more capable of identifying that it is under some form of novel local cyber-attack and should be able to dynamically push data that can be used to classify and construct an appropriate mitigation in future to the cloud (assuming that an attacker hasn't also compromised the network connection).

A telematics system is only as good as the information it is able to collect, with good referring to the relevancy of the information for the tasks requiring telematics. With the proposed solution relying upon existing In-vehicle network technologies to sample signal data, the information was subject to the constraints of the messages that contained it when exchanged on the network. While some applications have little by way of Quality of Service requirement, other applications that require low-latency or information that is not required to be distributed on the network cannot be implemented in this way. With these challenges in mind, the second portfolio submission explored "The Challenges Facing the Next Generation of In-Vehicle Network Architectures" looking to establish if a more recent entrant into the automotive network space, Ethernet, could potentially address these issues.

Ethernet-based vehicular networks look to present a networking solution that could enable more abstract developing paradigms such as Service Orientation, Remote Process Calls and Publish/Subscribe, which are already common outside of the automotive industry, by providing a range of higher-bandwidth, lower latency PHYs and more generic data structures that can integrate more directly with the Open Systems Interconnection (OSI) model.

Exploring the domain of automotive Ethernet is difficult, in part due to the early stage of development that most standards compliant solutions are in. With much of the focus from Silicon and IP manufacturers is upon the co-development of demonstrators with OEMs and their suppliers rather than production of commercially available solutions, this situation is unlikely to change until widespread adoption of the technology is achieved.

The third contribution looks at Simulating Time Sensitive Networks to explore novel solutions to problems that arise when working with the dynamic Service Orientated application architectures that would enable telematics systems of the future to access a wider range of applications while reducing the impact of the In-vehicle Network configuration of the latency of the signals themselves.

While the software modules and packages used to create the scenarios are highly constrained to the specific scenarios of interest for later work, there is some value in the creation of software tooling that would allow for combinations and permutations of these new Standards and Technologies without waiting for the hardware to catch-up and value persists in considering the interaction of different standards together. The Time Sensitive Network standards provide only a set of tools, and while relevant recent activity has focused on the creation of guidance and support for automotive applications exploring how these tools could interact and work together is itself a valuable question.

Much of the effort in the space of TSN is looking at the creation of static network configurations/topologies which will likely be the avenue taken by the automotive industry. This is almost entirely due to the safety challenges that non-deterministic systems introduce, however Submission 4 at its most fundamental was a proposal looking to solve the problems of having the network and applications running atop it to optimise themselves, at run time.

Considering how one might go about achieving run time reconfiguration with mixed criticality using the new standards is novel. This required pulling in themes from a variety of different automotive trends and applications beyond just looking at Vehicle to Cloud applications. The proposed approach utilises a model for the network latency, constructed from measurements collected from a simulated environment, which can be used to make predictions of the latency an individual frame will experience. Nodes utilise these predictions by tailoring their frame parameters to achieve their Quality of Service requirements. By continually adding additional samples to the model, across a diverse range of system states, the model is designed to converge upon a set of behaviours that enable the network to achieve the application quality of service requirements with minimal dropped frames or missed deadlines.

There are numerous ongoing challenges facing the wider adoption of integrated telematics, Service Orientation, Publish/Subscribe data models and automotive Ethernet. The proposals in this project were not aimed at finding the best or optimal solution but instead to present the problem from a new software-first perspective that the automotive industry is slowly adopting.

When considering the wider automotive industry, as it spawns entirely new markets out of the continued exploration of the capabilities of Connected and Autonomous Vehicles, it becomes apparent that new ways of addressing the challenge of software complexity required for these features and the ever-increasing appetite for vehicular data, even if they are not adopted, are needed. With a wide range of options, manufacturers, suppliers, and new entrants to the market will be able to more clearly identify their ongoing technology strategy.

With the pace ever increasing toward a vehicle as a software and data platform, contributions within this document: a Publish/Subscribe Dual Broker Telematics system, a Time Sensitive Network Ethernet simulation environment, and an approach for top-down/application-centric network configuration/optimisation can be utilised as the basis for the numerous technical challenges that lie ahead or as the seed for a wholly new approach.

On December 30th 2020, the methodology proposed as part of Submission 4, that looked to optimise traffic flows based upon Quality of Service requirements and a network model was granted by the UK Patent Office [96]. To be granted patent on an invention the must be something that can be made or used, new and have an inventive step. This, in conjunction with the financial costs associated with the filing and grant process indicates that some value for the ideas and concepts that derived the method.

10 Further Research - In-Vehicle Network Security

With the introduction of the General Data Protection Regulation (GDPR), numerous large scale data breaches/thefts/exfiltrations consumers are more aware of the need for the products and services that they use to have robust defences, with consumer research showing a decrease in corporate trust between 2017-2016 of 5% (down to 12%) [97].

Data privacy is one of the pillars of the cyber-security industry, when applied to an automotive industry that is increasingly providing intelligent route planning, concierge services and integration with other consumer electronics devices, the need for manufacturers to adopt a proactive approach to their products is crucial.

To that end, the United Nations Economic Commission for Europe (UN-ECE) World Forum for Harmonization of Vehicle Regulations (WP.29) has been working to develop regulations that enforce the use of Cyber Security Management System (CSMS) [98].

As part of this, at present draft, regulation requires an automotive manufacturer to demonstrate that they have rigorously identified, assessed and implemented appropriate mitigation's for risks during the: development phase; production phase and post-production phase [98].

Important to note here is that this regulation touches upon elements throughout the entire new product development cycle, requiring extensive examinations of supply chain partners and will eventually require the manufacturer to have in place appropriate monitoring, reporting and incident response for cyber threats for the entire life cycle of a vehicle, not just for the production period.

As part of the post-production phase, the vehicle manufacturer is expected to implement measures to [98]:

- Use a systematic risk-based approach defining organisational processes, responsibilities and governance to treat risk associated with cyber threats to vehicles and protect them from cyber-attacks (CSMS)
- Detect and prevent cyber-attacks against their vehicles
- Support the monitoring capability of the manufacture with regards to detecting threats, vulnerabilities and cyber attacks
- Provide forensic capability to assist with the analysis of cyber-attacks

In a corporate setting, many of these requirements are part of a mature approach to cyber-security and risk management. A Cyber Security Management System (CSMS) is a risk-based approach for defining process, risks, responsibilities and the governance in order to track and respond to threats, this is somewhat analogous to the Information Security Management System (ISMS) that many organisations leverage to assist with information security of their organisations. This comparison likens an ISMS, which is for an organisation, to a CSMS which is for a product, in this case a vehicle.

For example, the detection and prevention of cyber-attacks against assets aligns very much with commercial Managed Detection and Response (MDR) services that attempt to monitor endpoints throughout an organisation for signals of compromise.

This raises the question: What does an automotive cyber-attack look like?

The process of developing an understanding of the risk landscape, threat actors and security threats is called Threat Modelling. The STRIDE Model, developed by Microsoft, for identifying computer security threats, views threats as a violation of desirable system properties [99]:

- Spoofing is a violation of Authenticity
- Tampering is a violation of Integrity
- Repudiation is a violation of Non-repudiability
- Information Disclosure is a violation of Confidentiality
- Denial of Service is a violation of Availability
- Elevation of Privilege is a violation of Authorisation

Authenticity is the property of a system to be able to verify that the information is genuine an example of this in an automotive environment would be the ability for an actor to transmit messages onto the bus with any message ID. The traditional automotive environment has no mechanism for distinguishing between a message from an ECU that is supposed to use that message ID and another source - another way of putting this is that messages on the network are trusted.

Integrity is the property that information remains intact and unaltered when communicated. CAN, FlexRay and LIN have checksums as a partial measure of the integrity of a message, often protecting against occasional bit flips, with LIN (1.3) using an 8 bit checksum that is a sum of the data bytes, CAN using a 15 bit Cyclic Redundancy Check (CRC) polynomial and FlexRay, which has multiple different CRCs one in the header (11bit) and one for the data (24bit). The fact that a CRC or Frame Check Sequence (FCS) only provides a partial measure of integrity is that due to the nature of the methods collisions do exist within the mapping space and for some non-cryptographic measures it is theoretically possible, computationally expensive, to calculate collisions allowing for data to be modified without recalculating the checksum.

Non-repudiability is the property in a multi-actor system in which no device can deny that they sent a message, altered a state or verified something else when they have done so. Within a traditional automotive bus based network, given that there is no mechanism to guarantee the authenticity of a message received or its origin, other than it came from the bus this property is not incorporated into the vehicular architecture.

Confidentiality is the property that requires only the authorised parties are able to view the information. By design, the devices/parties that are authorised to receive information are those connected on the bus, while there is a bitmask filter incorporated into many bus transceivers (in order to only present relevant messages to the ECU), there is no guarantee that all devices connected, particularly in an adversarial environment, are those that the authorised and intended recipients of the data/information being transmitted.

The property of Availability is that the system is immune to denial or downgraded service. As discussed previously, there are numerous conditions in which an incorrectly configured device on a bus-based network could downgrade the communication service experienced by other devices on the network. In an adversarial environment, intentionally ignoring the medium access control of a CAN/FlexRay or LIN network or intentionally over competing for resource or just sending error frames could seriously impact the availability of the medium for other devices.

Authorisation is the ability of a system to restrict access to capabilities and deny privilege escalation. Many manufacturers have separate specialist networks that are grouped together by the commonality of the data that they need to share. Given the significant increase in inter-domain communication that Advanced Driver Assistance Systems (ADAS) and Smart Connected Automated Vehicles (SCAVs) require this had led, in many cases, to a gateway module that sits on all the buses acting as a relay between the different networks. While Privilege escalation is typically seen as gaining new permissions (vertical) moving between various networks is an example of horizontal privilege escalation. While in most cases, this is intended functionality, in situations where a maliciously constructed frame is able to exploit the gateway's configuration to talk to a device on a different network this can be considered privilege escalation.

As with all risk management exercises, especially those within the cyber security domain, the acknowledgement that there isn't a perfect solution, there is no one solution that makes your product, in this case a car, secure.

In the automotive environment, especially a Next Generation Architecture that utilises Service Discovery, Remote Procedure Calls (RPCs) and a continuation of the increases in the inter-domain communication that has driven automotive network design then it is inevitable that eventually the security of a system will be relevant to safety functions. This security and safety overlap already exists in the automotive industry that has limited attack surface due to the lack of connectivity, i.e. compromising one car is not going to compromise hundreds or thousands of others, this risk has been accepted implicitly. The biggest change is the requirement of manufacturers to explicitly justify the risks and mitigation that they have in-place to a certifying agency.

Zero Trust, a recent trend in the world of cloud and business networking, is another way of modelling and eventually building secure systems [100][101]. The underlying concept is that in every system, there are layers where information is passed between one component to another, this movement between components is a trust boundary. The argument follows that while the information remains within a trust boundary, the risks/threats it faces are well defined (importantly not non existent just because information remains inside a trust boundary), but outside of that context the information is open to the plethora of cyber-security attacks, some of which were described as part of STRIDE above.

The National Cyber Security Center in the United Kingdom specifies 8 principles that make up a zero trust environment [101]:

- Know your architecture including users, devices and services
- Know your user, service and device identities
- Know the health of your users, devices and services
- Use policies to authorise requests
- Authenticate everywhere
- Focus monitoring on devices and services
- Don't trust any network, including your own
- Choose services designed for zero trust

Many of the desirable properties align strongly with the themes in STRIDE and the requirements from the UNECE. Authentication which strongly correlates to Authorisation, Authenticity, Non-repudiation and Confidentiality is currently non-trivial to achieve on a traditional automotive networks. There is an implicit level of trust applied in traditional automotive network architectures as you expect them to behave correctly but there is no mechanism to guarantee this.

The impact of the implict trust that exists within traditional automotive networks can be seen, even within the context of this research. Submission 1 can be described as a 3rd party interfacing with a vehicle's data buses and extracting information for other purposes. This is entirely due to the nature of bus-based networks, with the inverse challenge of determining if a signal value was sent from a trusted source being more important due to the potential change propagating within the network, the only defence is that of obscurity of the mapping from payload bytes to signals.

Given the numerous types of threat, the new regulations, the zero trust design principles, how can Ethernet/IP be leveraged to reduce the attack surface?

The underlying motivations behind adopting Ethernet (and IP) as an automotive networking standard are: the cost reduction associated with a single networking standard; the ability to increase the reuse ability of modules by abstracting them above the hardware and network; the ability to leverage a mix of PHY speeds without a translation layer (gateway module); and the ability to leverage existing compatible standards.

It is the latter of these that is of most interest to the current conversation. The development of new EMC compliant PHYs, within the 802.3 working group, includes compatibility with the rest of the 802.1 standards.

The long term ambitions of a next generation automotive architecture, many of which have been discussed throughout this document, include: runtime service discovery; support for plug and play to reduce manufacturing complexity and begin to pave the wave for life time upgradability and pushing for hardware decoupling wherever possible. When mixing in the requirements of the imminent UNECE regulations, the categories of cyber threat considered under STRIDE and the philosophical approach to secure systems that is Zero Trust, the applications, many of which, will have safety critical functionality will need to be able to have a measure of trust in the underlying system components.

If a safety application is unable verify the Authenticity, Integrity, Nonrepudiability, Confidentiality, Availability and Authorisation of the other system components then it should fall back to an redundant system (if one is available) and if redundant options do not exist then it shouldn't continue to operate.

In an next generation automotive Ethernet architecture, all the hardware should have a unique identity that is globally unique, unable to be modified once configured and be stored in such a way that resists exfiltration (confidentiality) and modification (integrity). Within the automotive supply chain there already exists the hardware to provide this functionality in the form of a Hardware Security Module (HSM), in the consumer electronics domain a Trusted Platform Module (TPM) provides this functionality.

The first element of developing a secure system is ensuring that the software running is correct and hasn't been modified during transit, this property is especially important when considering Over-the-Air (OTA) update functionality. Secure Boot, is a mechanism that verifies that the firmware being loaded has been signed by the correct private key.

While this functionality is extremely useful given the complex set of regulations, it is likely that further extensions to SecureBoot such as Hardware Attestation [102][103] [104]. Hardware Attestation would enable manufacturers to identify if hardware components have been modified, rightly or wrongly, and in the case where sufficient trust cannot be established disable certain features until the records for what ECUs make up a vehicle are updated.

With each ECU within the network having a unique identity, with corresponding certificates, it becomes possible for an Ethernet switch to apply

Port-based Access Control. In 802.1X, Port-based Network Access Control is defined within 802.1X which defines a specific method of using Extensible Authentication Protocol (EAP) over LAN dubbed EAPOL [105] when combined with 802.1AE MAC Security (MACsec) this enables devices to associate with the switch, have their identity validated, and to receive a symmetric Secure Association Key (SAK) which is used to encrypt further transmission between those two points on the network. EAPOL and MACsec provide the integrity, through the use of an Integrity Check Value (ICV), and authentication by using the shared key.

At this stage, ECUs in this architecture would be able to establish new symmetric keys every power cycle, if necessary, which should make the attack surface the Hardware Secure Module, which should by virtue of it's design be subject to significantly more testing and validation because of it's importance to establishing trust.

With the ability to authenticate devices physically connected and and validate the origin of messages being sent, this becomes the first location for meeting the monitoring capability around cyber threats and being able to provide forensic capabilities. Having switch hardware capable of reporting incidents of failed authentication attempts and invalid packets will be one of many source of data for an Intrusion Detection System.

MACsec is a Layer 2 protocol and as discussed provides useful mechanisms to protect against malicious devices being connected to the network. While it would likely be possible to connect every device in a next generation vehicular architecture to the same Local Area Network, it is likely due to the desirable properties of TSN that 802.1Q tagged Ethernet frames will be used. This provides the ability for automotive manufacturers to separate functionality onto VLANs. Combine this with IPsec, which provides similar functionality to that of MACsec but at the network layer and two-way TLS would mitigate the risks associated with compromise in the application layer, by restricting access to other internal and external routes, within a hyper-visor or host operating system by restricting what MAC and IP ranges it could purport to be.

The automotive industry utilises a complex supply chain in order to deliver products and functionality to the customer. In a Next Generation Architecture, there is an increased need for each layer of complexity to be built atop a trusted platform. Obviously if a threat actor is able to compromise a Hardware Security Module then they would be able to impersonate that ECU on the network and as such be able to interact with any of the services that it is authorised to do.

The challenges of such a robust hardware security model can be seen in the Right to Repair movement in which certain manufacturers require each modular component of their system to be correctly signed before enabling the functionality. While this is currently being fought within the consumer electronics space, it is very likely that the automotive repair industry and supply chain will strongly oppose full hardware attestation without having a well documented process for adding hardware to the vehicle's approved list. This process itself will need to be developed with extreme care as a threat actor could subvert any functionality to just add their own hardware to the approved list.

11 Bibliography

[1] IEEE-SA, *IEEE 802.1Q-2018 - IEEE Standard for Local and Metropolitan Area Networks - Bridges and Bridged Networks*. IEEE Computer Society, 2018, vol. 2018, ISBN: 9781504449298. DOI: 10.1109/IEEESTD.2018.8403927. [Online]. Available: https://standards.ieee.org/standard/802_1Q-2018.html.

[2] AUTOSAR, "AUTOSAR Project Objectives," AUTOSAR, Tech. Rep. Foundation, 2016. [Online]. Available: https://www.autosar.org/fileadmin/user_upload/standards/foundation/1-0/AUTOSAR_RS_ProjectObjectives.pdf.

[3] S. Martínez-Fernández, C. P. Ayala, X. Franch, and E. Y. Nakagawa, "A survey on the benefits and drawbacks of AUTOSAR," *WASA 2015 - Proceedings of the 2015 ACM Workshop on Automotive Software Architecture, Part of CompArch 2015*, no. May, pp. 19–26, 2015. DOI: 10.1145/2752489.2752493.

[4] Mintel, "Executive Summary: Car Purchasing Process," Mintel Group, Tech. Rep. March 2019, 2019. [Online]. Available: https://academic.mintel.com/display/920578/?highlight.

[5] SAS Institute Inc., "The connected vehicle: big data, big opportunities," *SAS White paper*, p. 10, 2016. [Online]. Available: http://www.sas.com/content/dam/SAS/en_us/doc/whitepaper1/connected-vehicle-107832.pdf.

[6] European Commission, "The Race for Automotive Data," no. January, 2017. [Online]. Available: https://ec.europa.eu/growth/tools-databases/dem/monitor/sites/default/files/DTM_The%20race%20for%20automotive%20data%20v1.pdf.

[7] Deloitte LLP, "Big data and analytics in the automotive industry Automotive analytics thought piece Contents," p. 16, 2015. [Online]. Available: https://www2.deloitte.com/content/dam/Deloitte/uk/Documents/manufacturing/deloitte-uk-automotive-analytics.pdf.

[8] T. S. Catapult, "Market Forecast For Connected and Autonomous Vehicles," Centre for Connected & Autonomous Vehicles, Tech. Rep. July, 2017.

[9] T. S. O. M. M. (SMMT) and T. LIMITED, "CONNECTED AND AUTONOMOUS VEHICLES - 2019 REPORT: WINNING THE GLOBAL RACE TO MARKET," Tech. Rep., 2019, p. 13.

[10] 5G Automotive Association, *Explore the technology – 5G Automotive Association*. [Online]. Available: `https://5gaa.org/5g-technology/c-v2x/`.

[11] L. Miao, J. J. Virtusio, and K. L. Hua, "Pc5-based cellular-v2x evolution and deployment," *Sensors (Switzerland)*, vol. 21, no. 3, pp. 1–14, 2021, ISSN: 14248220. DOI: `10.3390/s21030843`.

[12] Deloitte LLP, "Global Mobile Consumer Survey (UK) - The Race to 5G," Tech. Rep., 2018. [Online]. Available: `http://www.deloitte.co.uk/mobileuk/assets/img/download/Global-Mobile-Consumer-Survey-2018-UK-Cut-The-race-to-5G.pdf`.

[13] T. GSMA, "Study on Socio-Economic Benefits of 5G Services Provided in mmWave Bands," no. December, 2018.

[14] P. Kremonas and J. Pâris, "eCall Factsheet," European Emergency Number Association (EENA), Tech. Rep., 2015. [Online]. Available: `https://eena.org/wp-content/uploads/2018/11/eCall-factsheet.pdf`.

[15] L. Bergonzi, "eCall Key Performance Indicators," *EENA Operations document*, pp. 1–19, 2016. [Online]. Available: `https://eena.org/wp-content/uploads/2018/11/eCall-Key-Performance-Indicators.pdf`.

[16] Ofcom, "Mobile not-spots," no. November, 2010. [Online]. Available: `https://www.ofcom.org.uk/__data/assets/pdf_file/0024/46158/not-spots.pdf`.

[17] Y. Agarwal, K. Jain, and O. Karabasoglu, "Smart vehicle monitoring and assistance using cloud computing in vehicular Ad Hoc networks," *International Journal of Transportation Science and Technology*, vol. 7, no. 1, pp. 60–73, 2018, ISSN: 20460430. DOI: `10.1016/j.ijtst.2017.`

12.001. [Online]. Available: https://doi.org/10.1016/j.ijtst.2017.12.001.

[18] W. He, G. Yan, and L. D. Xu, "Developing vehicular data cloud services in the IoT environment," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1587–1595, 2014, ISSN: 15513203. DOI: 10.1109/TII.2014.2299233.

[19] P. P. Ray, "A survey on Internet of Things architectures," *Journal of King Saud University - Computer and Information Sciences*, vol. 30, no. 3, pp. 291–319, 2018, ISSN: 22131248. DOI: 10.1016/j.jksuci.2016.10.003. [Online]. Available: https://doi.org/10.1016/j.jksuci.2016.10.003.

[20] ERICSSON IOT, *Connected Vehicle Cloud telematics service - Ericsson*. [Online]. Available: https://www.ericsson.com/en/internet-of-things/automotive/connected-vehicle-cloud/telematics-service.

[21] ——, *Automotive partnership built on innovation - Ericsson*. [Online]. Available: https://www.ericsson.com/en/cases/2017/partnerships-built-on-innovation.

[22] Tesla, *Introducing Software Version 10.0 — Tesla UK*. [Online]. Available: https://www.tesla.com/en_GB/blog/introducing-software-version-10-0.

[23] JAGUAR LAND ROVER LIMITED, *InControl — Infotainment Overview — Jaguar*. [Online]. Available: https://www.jaguar.com/incontrol-global/incontrol/infotainment.html.

[24] BMW, *Using BMW ConnectedDrive — BMW ConnectedDrive — BMW UK*. [Online]. Available: https://www.bmw.co.uk/en/topics/owners/bmw-connecteddrive/using-bmw-connecteddrive.html.

[25] Nissan, *Nissan in-vehicle infotainment system (IVI) — Nissan*. [Online]. Available: https://www.nissan.co.uk/ownership/nissan-infotainment-system.html.

[26] Ford Smart Mobility, *City Data Solutions - http://citydatareport.fordmedia.eu/*. [Online]. Available: http://citydatareport.fordmedia.eu/.

[27] Autonomic LLC, *Autonomic*. [Online]. Available: https://autonomic.ai/.

[28] MISRA, "Guidelines for the Use of C Language in Vehicle Based Software," no. April 1998, 1998.

[29] ——, *MISRA-C:2004 Guidelines for the use of the C Language in Critical Systems*, October 2004. 2004, pp. 1–116, ISBN: 978-0-9524156-4-0.

[30] ——, *MISRA Compliance:2016 Achieving compliance with MISRA Coding Guidelines*, April. 2016, ISBN: 9781906400132.

[31] R. Bagnara, A. Bagnara, and P. M. Hill, "The MISRA C Coding Standard and its Role in the Development and Analysis of Safety- and Security-Critical Embedded Software," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11002 LNCS, pp. 5–23, 2018, ISSN: 16113349. DOI: 10.1007/978-3-319-99725-4{_}2.

[32] AUTOSAR, "AUTOSAR Adaptive Platform Release Overview - 19-03," AUTOSAR, Tech. Rep., 2019, pp. 1–20.

[33] I. Cunningham, *Enabling Connectivity with Service Oriented Architectures Different Perspectives on Connectivity New Vehicle Functions and Services with Connectivity*, 2018. [Online]. Available: https://www.cenex-lcv.co.uk/storage/seminar-programme/sessions/presentations/iain_cunningham_enablin_connectivity_with_service_oriented_architectures_1537363468.pdf.

[34] LIN Consortium, "LIN Specification Package Revision 2.2A," Tech. Rep., 2010.

[35] J. Takahashi, Y. Aragane, T. Miyazawa, H. Fuji, H. Yamashita, K. Hayakawa, S. Ukai, and H. Hayakawa, "Automotive attacks and countermeasures on LIN-Bus," *Journal of Information Processing*, vol. 25, no. February, pp. 220–228, 2017, ISSN: 18826652. DOI: 10.2197/ipsjjip.25.220.

[36] R. hvanth, D. Valli, and K. Ganesan, "Design of an In-Vehicle Network (Using LIN, CAN and FlexRay), Gateway and its Diagnostics Using Vector CANoe," *American Journal of Signal Processing*, vol. 1, no. 2, pp. 40–45, 2012, ISSN: 2165-9354. DOI: 10.5923/j.ajsp.20110102.07.

[37] ISO, “Road Vehicles – Controller area network (CAN) conformance test plan Part 2 : High-speed medium access unit with selective wake-up functionality,” 2014.

[38] ——, “Road vehicles — Controller area network (CAN) — Part 3: Low-speed, fault-tolerant, medium-dependent interface,” no. 1, 2006.

[39] F. Hartwich and Robert Bosch GmbH, “CAN FD CAN with Flexible Data-rate Automotive Electronics,” Robert Bosch GmbH, Tech. Rep., 2012. [Online]. Available: https://www.can-cia.org/fileadmin/resources/documents/proceedings/2012_hartwich.pdf.

[40] ISO, “Road vehicles – Controller area network (CAN) – Part 1: Data link layer and physical signalling,” 2003.

[41] Bosch, “CAN with Flexible Data-Rate Specification Version 1.0,” vol. 0, p. 32, 2012.

[42] G. Marcon Zago and E. Pignaton De Freitas, “A Quantitative Performance Study on CAN and CAN FD Vehicular Networks,” *IEEE Transactions on Industrial Electronics*, vol. 65, no. 5, pp. 4413–4422, 2018, ISSN: 02780046. DOI: 10.1109/TIE.2017.2762638.

[43] ISO, “Road vehicles — FlexRay communications system — Part 1: General information and use case definition,” no. 17458-1:2013, 2013.

[44] ——, “FlexRay communications system Part 2 : Data link layer specification,” 2013.

[45] AUTOSAR, “Specification of Time Synchronization over FlexRay,” AUTOSAR, Tech. Rep., 2015.

[46] OPEN Alliance SIG, “IEEE 100BASE-T1 System Implementation Specification,” OPEN Alliance SIG, Tech. Rep., 2017, pp. 1–28.

[47] IEEE 802.3 Ethernet Working Group, “IEEE Standard for Ethernet Amendment 1 : Physical Layer Specifications and Management Parameters for 100 Mb / s Operation over a Single Balanced Twisted Pair Cable IEEE Standard for Ethernet Amendment 1 : Physical Layer Specifications and Management Param,” *IEEE Std 802.3bw-2015 (Amendment to IEEE Std 802.3-2015*, vol. 2015, 2016. DOI: 10.1109/IEEESTD.2016.7433918.

[48] "IEEE Standard for Ethernet Amendment 4: Physical Layer Specifications and Management Parameters for 1 Gb/s Operation over a Single Twisted-Pair Copper Cable," *IEEE Std 802.3bp-2016 (Amendment to IEEE Std 802.3-2015 as amended by IEEE Std 802.3bw-2015, IEEE Std 802.3by-2016, and IEEE Std 802.3bq-2016)*, pp. 1–211, 2016. DOI: 10.1109/IEEESTD.2016.7564011.

[49] ——, *IEEE Standard for Ethernet Amendment 9: Physical Layer Specifications and Management Parameters for 1000 Mb/s Operation Over Plastic Optical Fiber*, 2017. DOI: 10.1109/IEEESTD.2017.7875375.

[50] Broadcom, "Transceiver Specification For Automotive Applications," 2014.

[51] IEEE Computer Society, *IEEE Standard for Ethernet 802.3-2018*. 2018, vol. 2018, pp. 1–5600, ISBN: 973-07381-7312-2. DOI: 10.1109/IEEESTD.2018.6419735.

[52] 802.3 10Mb/s Single Pair Ethernet Task Force, "IEEE 802.3 10Mb/s Single Pair Ethernet Task Force - Objectives," no. March 2018, pp. 10–12, 2018.

[53] P. Beruto and A. Orzelli, *Canova Tech - PLCA*, 2018. [Online]. Available: http://www.ieee802.org/3/cg/public/July2018/PLCA%20overview.pdf.

[54] IEEE 802.3 Multi-Gig Automotive Group Ethernet Phy Study Group, *IEEE 802.3 Multi-Gig Automotive Ethernet PHY Task Force - Approved Objectives*, 2017. [Online]. Available: http://www.ieee802.org/3/ch/0317_approved_objectives_3NGAUTO.pdf.

[55] NAV Alliance, *Navigating the Future of Connectivity*, 2018. [Online]. Available: https://nav-alliance.org/wp-content/uploads/2018/10/NAV-Alliance-deck-October-2018.pdf.

[56] M. T. G. o. I. 802.1, "Approved Draft - IEEE P802.1Q-REV/D2.2," Institute of Electrical and Electronics Engineers, Tech. Rep., 2018, p. 2000.

[57] "IEEE Standard for Local and metropolitan area networks — Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks — Corrigendum1 : Technical and Editorial Corrections," Tech. Rep. March, 2015.

[58] IEEE Computer Society. LAN/MAN Standards Committee., *IEEE Standard for Local and metropolitan area networks–Frame Replication and Elimination for Reliability*. 2017, pp. 1–102, ISBN: 9781504442978. DOI: `10.1109/IEEESTD.2017.8091139`. [Online]. Available: `https://release1-www.engineeringvillage.com/share/document.url?mid=cpx_2abac0741609323de53M5fe910178163176&database=cpx`.

[59] L. Zhao, P. Pop, and S. S. Craciunas, "Worst-Case Latency Analysis for IEEE 802.1Qbv Time Sensitive Networks Using Network Calculus," *IEEE Access*, vol. 6, pp. 41 803–41 815, 2018, ISSN: 21693536. DOI: `10.1109/ACCESS.2018.2858767`.

[60] L. Zhao, F. He, E. Li, and H. Xiong, "Improving worst-case delay analysis for traffic of additional stream reservation class in ethernet-AVB network," *Sensors (Switzerland)*, vol. 18, no. 11, 2018, ISSN: 14248220. DOI: `10.3390/s18113849`.

[61] M. Gutiérrez, A. Ademaj, and W. Steiner, "Self-Configuration of IEEE 802 . 1 TSN Networks,"

[62] J. Pfrommer, A. Ebner, S. Ravikumar, and B. Karunakaran, "Open Source OPC UA PubSub over TSN for Realtime Industrial Communication," *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, vol. 2018-Septe, pp. 1087–1090, 2018, ISSN: 19460759. DOI: `10.1109/ETFA.2018.8502479`.

[63] C. R. Spitzer, *The avionics handbook*. CRC Press, 2001, ISBN: 9780849383489.

[64] S. I. Kazi, "Architecting of Avionics Full Duplex Ethernet (AFDX) Aerospace Communication Network," ISSN: 2230-9543.

[65] Emerson Network Power, "The Promises, Challenges and Imperatives of Data Center Infrastructure Management," Emerson Network Power, Tech. Rep., 2010. [Online]. Available: `https://www.vertiv.com/globalassets/products/monitoring-control-and-management/software/data-center-infrastructure-management-promises-`

challenges-and-imperatives-of-data-center-infrastructure-management.pdf.

[66] M. Strohmeier, M. Schäfer, R. Pinheiro, V. Lenders, and I. Martinovic, "On Perception and Reality in Wireless Air Traffic Communication Security," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 6, pp. 1338–1357, 2017, ISSN: 15249050. DOI: 10.1109/TITS.2016.2612584.

[67] J. B. Maggiore, "Remote Management of Real-Time Airplane Data," *Boeing AERO*, no. 3, pp. 22–27, 2007. [Online]. Available: http://www.boeing.com/commercial/aeromagazine/articles/qtr_3_07/AERO_Q307.pdf.

[68] Deloitte, "Fasten your seatbelts: in-flight connectivity takes off," no. October 2017, 2017. [Online]. Available: https://www.deloitte.co.uk/tmtpredictions/assets/img/downloads/FY18-Predictions-In-Flight-Connectivity-Single-Prediction.pdf.

[69] B. Venkata, S. #1, and C. Sree Vardhan, "Telematics and its Applications in Automobile Industry," *International Journal of Engineering Trends and Technology (IJETT)*, vol. 4, no. April, pp. 554–557, 2013, ISSN: 2231-5381. DOI: 10.2475/06.2006.02. [Online]. Available: http://www.ijettjournal.org.

[70] Bearing Point., Management & Technology Consultants, "Global Automotive Warranty Survey Report," pp. 1–30, 2015. [Online]. Available: http://www.bearingpoint.com/ecomaXL/files/AutoWarrantyReport_final_web.pdf.

[71] B. Choi and J. Ishii, "Consumer perception of warranty as signal of quality: An empirical study of powertrain warranties," *Journal of Chemical Information and Modeling*, pp. 1–22, 2010, ISSN: 1098-6596. DOI: 10.1017/CBO9781107415324.004.

[72] D. Howdle and Cable.co.uk, "The cost of 1GB of mobile data in 230 countries," Cable.co.uk, Tech. Rep. November 2018, 2019, pp. 1–4. [Online]. Available: https://s3-eu-west-1.amazonaws.com/assets.cable.co.uk/mobile-data-cost/cost-of-1gb-mobile-data-in-230-countries-release.pdf.

[73] Department for Transport, "Vehicle Licensing Statistics : Annual 2018," Department for Transport, Tech. Rep. April, 2019. [Online]. Available: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/800502/vehicle-licensing-statistics-2018.pdf.

[74] H. Griffiths, "STRUCTURE OF THE UK AUTOMOTIVE TELEMATICS MARKET With a focus on the use of telematics in road safety solutions STRUCTURE OF THE UK AUTOMOTIVE TELEMATICS MARKET With a focus on the use of telematics in road safety solutions FUTURE CITIES CATAPULT STRUCTUR," Tech. Rep., 2017. [Online]. Available: https://iotuk.org.uk/wp-content/uploads/2017/04/Structure-of-the-UK-Automotive-telematics-Market.pdf.

[75] J. Wan, J. Liu, Z. Shao, A. V. Vasilakos, M. Imran, and K. Zhou, "Mobile crowd sensing for traffic prediction in internet of vehicles," *Sensors (Switzerland)*, vol. 16, no. 1, pp. 1–15, 2016, ISSN: 14248220. DOI: 10.3390/s16010088.

[76] G. Huston, *Evaluating IPv4 and IPv6 Packet Fragmentation — RIPE Labs*, 2016. [Online]. Available: https://labs.ripe.net/Members/gih/evaluating-ipv4-and-ipv6-packet-fragmentation.

[77] G. Pasaoglu, D. Fiorello, A. Martino, G. Scarcella, A. Alemanno, A. Zubaryeva, and C. Thiel, *Driving and parking patterns of European car drivers - a mobility survey*. 2012, p. 112, ISBN: 9789279277382. DOI: 10.2790/7028. [Online]. Available: http://setis.ec.europa.eu/system/files/Driving_and_parking_patterns_of_European_car_drivers-a_mobility_survey.pdf.

[78] R. Davis, *Real-Time Scheduling and Automotive Networks Controller Area Network (CAN)*. [Online]. Available: https://www-users.cs.york.ac.uk/~robdavis/papers/CESEC_CAN.pdf.

[79] Ofcom, "Measuring Mobile Broadband Performance in the UK -4G and 3G network performance," no. November, pp. 1–70, 2014. [Online]. Available: http://www.epitiro.com/assets/files/20-101-1002_001%20Ofcom%20Mobile%20Broadband.pdf.

[80] HiveMQ, "Enabling the Connected Car with HiveMQ," 2019, [Online]. Available: https://www.hivemq.com/solutions/iot/enabling-the-connected-car/.

[81] J. Martin, *DDS - EProsima*, 2019. [Online]. Available: https://www.eprosima.com/index.php/resources-all/dds-all.

[82] Real-Time Innovations, *Data Distribution Service Architecture (DDS) — OMG*, 2019. [Online]. Available: https://www.rti.com/products/dds-standard.

[83] ISO, "ISO 17987: Road vehicles — Local Interconnect Network (LIN) Part 1 : General information and use case definition," ISO, Tech. Rep., 2016.

[84] M. Mody, *ADAS Front Camera: Demystifying Resolution and Frame-Rate — EE Times*, 2016. [Online]. Available: https://www.eetimes.com/author.asp?section_id=36&doc_id=1329109#.

[85] R. Bielby, *ADAS Camera Requirements – Driving Memory Needs*, 2017. [Online]. Available: https://www.micron.com/about/blog/2017/may/adas-camera-requirements.

[86] Velodyne, *Application Note: Packet structure & Timing Definition.* [Online]. Available: https://velodynelidar.com/docs/notes/63-9276%20Rev%20A%20VLP-16%20Application%20Note%20-%20Packet%20Structure%20&%20Timing%20Definition_Locked.pdf.

[87] T. F. Wendt, W. Bernhart, J. Behl, D. Mishoulam, and E. Goldsmith, "Consolidation in vehicle electronic architectures," Roland Berger Strategy Consultants LLC, Tech. Rep. July, 2015. [Online]. Available: https://www.rolandberger.com/publications/publication_pdf/roland_berger_tab_consolidation_in_vehicle_electronic_architectures_20150721.pdf.

[88] C. Berg, *ECU Consolidation and New Automotive Topologies using Hypervisors SYSGO // Brief Company Overview*, 2018. [Online]. Available: https://www.sysgo.com/fileadmin/user_upload/www.sysgo.com/redaktion/downloads/pdf/presentations/ECU_Consolidation_EW18.pdf.

[89] Intel, "ECU Consolidation Reduces Vehicle Cost , Weight , and Testing," 2017, [Online]. Available: https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/ecu-consolidation-white-paper.pdf.

[90] M. Miller, "10 Mbps Single Pair Ethernet (10SPE – 10 BASE-T1S)," in *IEEE Ethernet & IP @ Automotive Technology Day 2018*, 2018, pp. 1–13.

[91] R. Johansson, P. Johannessen, K. Forsberg, H. Sivencrona, and J. Torin, "On Communication Requirements for Control-by-Wire Applications," in *Proceedings of the 21st International System Safety Conference*, 2003. DOI: 10.16309/j.cnki.issn.1007-1776.2003.03.004. [Online]. Available: https://pdfs.semanticscholar.org/2dc0/3068c8c7a85399518f7ff7536b0f pdf.

[92] TETCOS, *NetSim-Network Simulator & Emulator — Version Comparision*. [Online]. Available: https://www.tetcos.com/version-comparison.html.

[93] nsnam, *ns-3 — a discrete-event network simulator for internet systems*, 2019. [Online]. Available: https://www.nsnam.org/.

[94] OMNeT++, *OMNeT++ Discrete Event Simulator*, 2019. [Online]. Available: https://omnetpp.org/.

[95] INET, *INET Framework - INET Framework*, 2019. [Online]. Available: https://inet.omnetpp.org/.

[96] L. S. James, *Network Optimisation Method*, 2020. [Online]. Available: https://www.ipo.gov.uk/p-ipsum/Case/ApplicationNumber/GB1814204.2.

[97] PricewaterhouseCoopers, *Consumer Intelligence Series: Protect.me*, 2017. [Online]. Available: https://www.pwc.com/us/en/advisory-services/publications/consumer-intelligence-series/protect-me/cis-protect-me-findings.pdf.

[98] UNECE World Forum for Harmonization of Vehicle Regulations (WP.29), "Proposal for a new UN Regulation on uniform provisions concerning the approval of vehicles with regards to cyber security and cyber security management system," vol. 2020, no. June, p. 27, 2020. [Online].

Available: http://www.unece.org/fileadmin/DAM/trans/doc/2020/wp29grva/ECE-TRANS-WP29-2020-079-Revised.pdf.

[99] Microsoft, *The STRIDE Threat Model*, 2009. [Online]. Available: https://docs.microsoft.com/en-us/previous-versions/commerce-server/ee823878(v=cs.20)?redirectedfrom=MSDN.

[100] National Institute of Standards and Technology, "Zero Trust Architecture - NIST Special Publication 800-207," *Nist*, p. 49, 2020. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-207-draft2.pdf.

[101] National Cyber Security Center, "Zero trust principles - beta release," pp. 2–5, 2021.

[102] T. C. Group, "TCG Platform Certificate Profile," 2019.

[103] NSA Cyber, *Host Integrity at Runtime and Start-up (HIRS)*, 2021. [Online]. Available: https://github.com/nsacyber/HIRS.

[104] T. C. Group, "DICE Attestation Architecture," pp. 0–35, 2020.

[105] IEEE Computer Society. LAN/MAN Standards Committee., *IEEE Std 802.1AX^{TM}-2020, IEEE Standard for Local and Metropolitan Area Networks - Port-based Network Access Control*. 2020, vol. 2020, ISBN: 9781504464307.