



Pós-Graduação em Ciência da Computação

Rhudney Alyson Menna Barreto König Simões

Uma Extensão da Arquitetura de Estações ITS ETSI para Comunicação entre Aplicações ITS, IoT e SOME/IP



Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
<http://cin.ufpe.br/~posgraduacao>

Recife

2017

Rhudney Alyson Menna Barreto König Simões

**Uma Extensão da Arquitetura de Estações ITS ETSI para Comunicação entre
Aplicações ITS, IoT e SOME/IP**

Trabalho apresentado ao Programa de Pós-graduação
em Ciências da Computação da Universidade Federal
de Pernambuco como requisito parcial para obtenção do
grau de Mestre em Ciência da Computação

Orientador(a): Prof. Divanilson Rodrigo de Sousa Campelo

Recife
2017

Catálogo na fonte
Bibliotecário Jefferson Luiz Alves Nazareno CRB 4-1758

S593e Simões, Rhudney Alyson Menna Barreto König.
Uma extensão da arquitetura de estações ITS ETSI para comunicação entre aplicações ITS, IoT e SOME/IP / Rhudney Alyson Menna Barreto König Simões. – 2017.
68f.: fig., tab.

Orientador: Divanilson Rodrigo de Sousa Campelo.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn. Ciência da Computação. Recife, 2017.
Inclui referências.

1. Ciência da computação. 2. Internet das coisas. 3. Avaliação de arquitetura. 4. Carros conectados. I. Campelo, Divanilson Rodrigo de Sousa. (orientador) II. Título.

004 CDD (22. ed.) UFPE-MEI 2018-118

Dissertação de Mestrado apresentada por **Rhudney Alyson Menna Barreto König Simões** à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título “**Uma Extensão da Arquitetura de Estações ITS ETSI para Comunicação entre Aplicações ITS, IoT e SOME/IP**”
Orientador: Divanilson Rodrigo de Sousa Campelo e aprovada pela Banca Examinadora formada pelos professores:

Prof. José Augusto Suruagy Monteiro Centro de Informática
UFPE

Prof. Fernando Antonio Aires Lins
Departamento de Estatística e Informática UFRPE

Prof. Orientador: Divanilson Rodrigo de Sousa Campelo
Centro de Informática UFPE

Visto e permitida a impressão.
Recife, 15 de Setembro de 2017.

Prof. Aluizio Fausto Ribeiro Araújo
Coordenador da Pós-Graduação em Ciência da Computação do
Centro de Informática da Universidade Federal de Pernambuco.

Dedico este trabalho à minha família e minha noiva, que foram porto seguro perante as dificuldades durante este percurso.

Agradecimentos

Inicio meus agradecimentos por DEUS, já que Ele colocou pessoas tão especiais a meu lado, sem as quais certamente não teria dado conta!

Aos meus pais, Francisco Simões e Lili Simões, meu irmão, Rhodney, minha noiva, Thamires, os meus amigos e meu orientador Divanilson.

Resumo

Atualmente, diversos dispositivos podem se conectar a veículos, como *smartphones*, relógios inteligentes e outros carros. Sendo assim, tecnologias de comunicação com diferentes finalidades serão usadas ao mesmo tempo em veículos, como, por exemplo, Bluetooth Smart, IEEE 802.15.4, IEEE 802.11p e 4G /5G / LTE. Ademais, nas redes intraveiculares de próxima geração baseadas em Ethernet, o *Middleware Escalável Orientado a Serviço sobre IP (SOME/IP Scalable service-Oriented MiddlewarE over IP)* apresenta-se como a solução de *middleware* orientado a serviços para a transmissão de dados não críticos em uma rede Ethernet automotiva. Através do SOME/IP, dados gerados por nós da rede Ethernet automotiva podem ser consumidos através de requisições orientadas a serviços. Segundo o Instituto Europeu de Normas de Telecomunicações (ETSI *European Telecommunications Standards Institute*), uma Estação de Sistema de Transporte Inteligente (ITS-S *Intelligent Transportation System Station*) pode ser qualquer dispositivo que implemente a arquitetura de comunicação ETSI ITS. Uma aplicação ITS tem finalidades diversas e opera na última camada da arquitetura de comunicação de uma ITS-S. Devido às diferentes tecnologias de comunicação que poderiam ser usadas, é extremamente complexo e dispendioso criar aplicações ITS para se comunicarem com dispositivos de Internet das Coisas (IoT *Internet of Things*) e consumirem dados oriundos da rede Ethernet automotiva. Este trabalho propõe uma extensão da arquitetura de uma ITS-S do padrão ETSI, que provê comunicação entre aplicações ITS, dispositivos IoT e nós da rede Ethernet. O objetivo deste trabalho é desenvolver e avaliar a extensão proposta. A avaliação foi realizada através da simulação em OMNeT++ de aplicações ITS que se comunicam com dispositivos IoT e nós da rede Ethernet intraveicular usando SOME/IP. A métrica avaliada foi o atraso fim-a-fim entre a realização de requisições por aplicações ITS e a chegada de suas respostas. Resultados de simulação mostram que a extensão proposta neste trabalho atende as restrições de atraso máximo permitido para a comunicação de dados não-críticos.

Palavras-chaves: Carros Conectados. Internet das Coisas. Avaliação de Arquitetura. ETSI Standards. SOME/IP.

Abstract

Nowadays, many devices can connect to vehicles such as smartphones, smartwatches, and other cars. Therefore, communication technologies with different purposes will be used at the same time in vehicles, such as Bluetooth Smart, IEEE 802.15.4, IEEE 802.11p, and 4G / 5G / LTE. In addition, in Ethernet-based next-generation intravehicular networks, Scalable service-Oriented MiddlewarE over IP (SOME/IP) is the service-oriented middleware solution for non-critical data transmission in an automotive Ethernet network. Using SOME/IP, data generated by automotive Ethernet network nodes can be consumed through service-oriented requests. According to European Telecommunications Standards Institute (ETSI), an Intelligent Transportation System Station (ITS-S) can be any device that implements the ETSI ITS communication architecture. An ITS application has several purposes and operates on the last layer of the communication architecture of an ITS-S. Because of the different communication technologies that could be used, it would be extremely complex and costly to create ITS applications to communicate with Internet of Things (IoT) devices and consume data from the automotive Ethernet network. This work aims to develop and evaluate an extension of the architecture of an ITS-S of the ETSI standard. The extension provides communication between ITS applications, IoT devices, and Ethernet network nodes. The evaluation was performed through simulation in OMNeT ++ of ITS applications that communicate with IoT devices and nodes of the intravehicular Ethernet network using SOME/IP. The measured metric was the end-to-end delay between the requests by ITS applications and the arrival of their responses. Simulation results show that the ITS-S architecture extension meets the maximum delay constraints allowed for non-critical data communication.

Key-words: Connected Cars. Internet of Things. Architecture Evaluation. ETSI Standards. SOME/IP.

Lista de ilustrações

Figura 1 – Comunicação entre cliente e servidor utilizando SOME/IP (AGLAIA, 2014)	25
Figura 2 – Diagrama de sequência do método publish/subscribe	26
Figura 3 – Middleware SOME/IP nas arquiteturas do Autosar e GENIVI (VÖLKER, 2014)	27
Figura 4 – Mensagem SOME/IP	27
Figura 5 – Arquitetura ETSI ITS-S (ITS-S, 2010)	31
Figura 6 – Arquitetura de uma estação ITS com as extensões para comunicar-se com dispositivos de IoT e rede intraveicular	36
Figura 7 – Gateway para comunicação com sistemas intraveiculares na arquitetura de uma estação ITS (ETSI, 2013)	37
Figura 8 – ECU SomeipGateway atuando como nó da rede Ethernet intraveicular	38
Figura 9 – ECU SomeipGateway atuando como nó da rede Ethernet intraveicular	39
Figura 10 – Extensão da arquitetura aplicada ao cenário de rede intraveicular . . .	40
Figura 11 – Fila de requisições da arquitetura	41
Figura 12 – Diagrama de sequência da comunicação entre aplicações ITS e ECUs utilizando SOME/IP	42
Figura 13 – Extensão da arquitetura aplicada ao cenário de Internet das Coisas . .	43
Figura 14 – Diagrama de sequência da comunicação entre aplicações ITS e dispositivos IoT	44
Figura 15 – Caminho dos dados no cenário de Redes Intraveiculares	48
Figura 16 – Caminho dos dados no cenário de Internet das Coisas	49

Lista de tabelas

Tabela 1 – Análise dos trabalhos relacionados	33
Tabela 2 – Tabela para referenciar informações às ECUs que fornecem os resultados	41
Tabela 3 – Tabela para referenciar aplicações que fizeram as requisições	41
Tabela 4 – Tabela para conexões com dispositivos IoT	44
Tabela 5 – Tabela de atrasos máximos fim-a-fim por domínio em redes intraveicu- lares (TUOHY et al., 2015)	48
Tabela 6 – Ferramentas usadas para o desenvolvimento do projeto	51
Tabela 7 – Estatística descritiva para o experimento 5.1.2	55
Tabela 8 – Estatística descritiva para o experimento 5.1.3	57
Tabela 9 – Estatística descritiva para o experimento 5.1.4	59
Tabela 10 – Estatística descritiva para o experimento 5.1.3	60

Lista de abreviaturas e siglas

6LoWPAN IPv6 over Low Power WPAN IPv6 sobre Redes de Baixa Potência WPAN.

ADAS Advanced Driver Assistance Systems Sistemas Avançados de Assistência ao Condutor.

AUTOSAR AUTomotive Open System ARchitecture Arquitetura de Sistema Aberto Automotivo.

BLE Bluetooth Low Energy Bluetooth de Baixa Energia.

CAM Cooperative Awareness Message Mensagem de Conscientização Cooperativa.

CAN Controller Area Network Rede de Área do Controlador.

CCTV Closed-Circuit Television Circuitos Fechados de Televisão.

CI Cidades Inteligentes.

CoAP Constrained Application Protocol Protocolo de Aplicação Restrita.

DENM Decentralized Environmental Notification Message Mensagem de Notificação Ambiental Descentralizada.

DSRC Dedicated Short Range Communication Comunicação Dedicada de Curto Alcance.

ECU Eletronic Control Unit Unidades de Controle Eletrônico.

ETSI European Telecommunications Standards Institute Instituto Europeu de Normas de Telecomunicações.

FD CAN - CAN with Flexible Data rates CAN com Taxa de Dados Flexíveis.

HMI Human Machine Interface Interface Humano Máquina.

HS CAN - High-Speed CAN CAN de Alta Velocidade.

IC Intervalo de Confiança.

IEEE Institute of Electrical and Electronics Engineers Instituto de Engenheiros Elétricos e Eletrônicos.

IoT *Internet of Things* Internet das Coisas.

IP *Internet Protocol* Protocolo de Internet.

ITS-S *Intelligent Transportation System Station* Estação de Sistema de Transporte Inteligente.

LIN *Local Interconnect Network* Rede de Interconexão Local.

LS CAN - *Low-Speed CAN* CAN de Baixa Velocidade.

M2M *Machine-to-Machine* Máquina-para-Máquina.

MOST *Media Oriented Systems Transport* Transportes de Sistemas Orientados à Mídia.

OBD2 *On-Board Diagnostic II* Diagnóstico a Bordo II.

OBU *On Board Units* Unidades de Bordo.

RFID *Radio-Frequency Identification* Identificação de Rádio Frequência.

RSU *Roadside Unit* Unidades do Lado da Estrada.

S.O *Operational Systems* Sistema Operacional.

SD *Service Discovery* Descoberta de Serviço.

SOME/IP *Scalable service-Oriented MiddlewarE over IP Middleware* Escalável Orientado a Serviço sobre IP.

V2D *Vehicle-to-Device* Veículo para Dispositivos.

V2I *Vehicle-to-Vehicle* Veículo para Infraestrutura.

V2P *Vehicle-to-Pedestrian* Veículo para Pedestres.

V2V *Vehicle-to-Infrastructure* Veículo para Veículo.

V2X *Vehicle-to-Everything* Veículo para Qualquer coisa.

VANET *Vehicular Ad Hoc Networks* Rede Veicular Ad Hoc.

WAVE *Wireless Access in Vehicular Environments* Acesso sem fio em ambientes de veículos.

WSN *Wireless Sensor Networks* Redes de Sensores Sem Fio.

Sumário

1	Introdução	15
1.1	<i>Caracterização do Problema</i>	15
1.2	<i>Objetivos</i>	16
<u>1.2.1</u>	<u>Objetivos Gerais</u>	16
<u>1.2.2</u>	<u>Objetivos Específicos</u>	17
1.3	<i>Apresentação da Dissertação</i>	17
1.4	<i>Considerações Finais</i>	18
2	Background e Trabalhos Relacionados	19
2.1	<i>Internet das Coisas</i>	19
<u>2.1.1</u>	<u>Sistema Operacional de IoT</u>	21
2.1.1.1	Contiki	21
2.1.1.2	Arm mbed	21
2.2	<i>Redes Veiculares</i>	22
<u>2.2.1</u>	<u>Redes Intraveiculares</u>	22
2.2.1.1	CAN	23
2.2.1.2	LIN	23
2.2.1.3	MOST	24
2.2.1.4	SOME/IP	24
<u>2.2.2</u>	<u>Redes Interveiculares</u>	28
2.2.2.1	ETSI TC ITS	29
2.3	<i>Trabalhos Relacionados</i>	31
2.4	<i>Considerações Finais</i>	34
3	Extensão Proposta da Arquitetura ITS-S	35
3.1	<i>Visão Geral da Extensão Proposta</i>	35
3.2	<i>Extensão da Arquitetura ITS-S para Comunicação com a Rede Ethernet</i>	37
	<i>intraveicular</i>	37
<u>3.2.1</u>	<u>Gateway para intercomunicação entre aplicações ITS e SOME/IP</u>	37
<u>3.2.2</u>	<u>Facility para comunicação entre aplicações ITS e SOME/IP</u>	39
3.3	<i>Extensão da Arquitetura ITS-S para Comunicação com Dispositivos IoT</i>	42
3.4	<i>Considerações Finais</i>	44

4	Metodologia de Avaliação	46
4.1	<i>Obtenção dos dados</i>	46
4.2	<i>Cenários</i>	47
<u>4.2.1</u>	<u>Redes Intraveiculares</u>	47
<u>4.2.2</u>	<u>Internet das Coisas</u>	49
<u>4.2.3</u>	<u>Cenário Integrado</u>	49
4.3	<i>Métricas</i>	50
4.4	<i>Ferramentas</i>	50
4.5	<i>Simulações</i>	51
<u>4.5.1</u>	<u>Simulações no contexto de redes intraveiculares</u>	51
4.5.1.1	Requisições com tempos definidos	52
4.5.1.2	Requisições por distribuição exponencial	52
<u>4.5.2</u>	<u>Simulações no contexto de Internet das Coisas</u>	53
<u>4.5.3</u>	<u>Simulações no contexto integrado da arquitetura</u>	53
4.6	<i>Considerações Finais</i>	53
5	Resultados	55
5.1	<i>Cenário de Rede Intraveicular</i>	55
<u>5.1.1</u>	<u>Requisições com tempos definidos</u>	55
<u>5.1.2</u>	<u>Requisições com distribuição exponencial</u>	57
5.2	<i>Cenário de Internet das Coisas</i>	59
5.3	<i>Cenário integrado</i>	59
5.4	<i>Discussão</i>	61
5.5	<i>Considerações Finais</i>	62
6	Conclusão e Trabalhos Futuros	63
6.1	<i>Principais Resultados</i>	63
6.2	<i>Trabalhos Futuros</i>	64
	Referências	66

1 Introdução

Este capítulo descreve a introdução deste trabalho. É iniciado com a caracterização do problema encontrado, o qual motivou a implementação deste trabalho de mestrado. É apresentando também os objetivos gerais e específicos. Por fim, faço uma breve apresentação dos capítulos e dos conteúdos que serão abordados.

1.1 Caracterização do Problema

A Internet das Coisas (IoT *Internet of Things*) deve revolucionar muitas áreas da sociedade, como, por exemplo, a saúde (YUEHONG et al., 2016), a educação (MOREIRA et al., 2017) e a segurança (FRAGA-LAMAS et al., 2016). De acordo com (RIVERA; MEULEN, 2014), mais de 25 bilhões de coisas conectadas serão utilizadas em 2020 e o setor automotivo mostrou a maior taxa de crescimento com relação à quantidade de dispositivos IoT usados. Portanto, espera-se que a IoT trará um grande número de novas aplicações especialmente no cenário automotivo.

As redes veiculares e automotivas têm sido extensivamente estudadas nos últimos anos. Atualmente com o avanço tecnológico em Sistemas Avançados de Assistência ao Condutor (ADAS *Advanced Driver Assistance Systems*), carros elétricos e autônomos também tornaram-se foco de pesquisas e investimentos, aumentando o interesse de pesquisadores e indústrias nessa área. Os veículos modernos oferecem aos condutores uma gama de funcionalidades de segurança, conforto e assistência. Com o aprimoramento tecnológico dos carros autônomos e carros conectados, o número de sensores e Unidades de Controle Eletrônico (ECU *Electronic Control Unit*) tende a aumentar significativamente. Além disso, diferentes tecnologias de rede nos veículos para diferentes requisitos devem existir, as quais tornarão o sistema de comunicação do carro ainda mais complexo que os atuais. O surgimento dessas funcionalidades automotivas aumentou os requisitos de largura de banda para a comunicação entre módulos eletrônicos ECUs nos veículos, o que fez surgir a necessidade de uma rede intraveicular de maior largura de banda. A Ethernet, amplamente usada em escritórios, residências e na indústria, passou a ser investigada para o cenário automotivo. Com o surgimento do BroadR-Reach (BROADCOM, 2011) e a padronização no IEEE da Ethernet Automotiva a 100 Mbps transmitida sobre um par de fios não blindados, a 100 BASE-T1 (100BASE-T1, 2015) permitiu a introdução da Ethernet em veículos. A introdução da Ethernet em automóveis tem ocorrido, primeiramente, no

domínio de Informação e Entretenimento (Infotainment), que exige uma grande largura de banda (SOARES et al., 2015), (LAJMI; ALIMI; AJILI, 2013).

A degradação ambiental, dificuldade de tráfego e de mobilidade nas grandes cidades e o aumento dos acidentes no trânsito representam apenas alguns dos transtornos decorrentes da utilização dos veículos automotores. Visando resolver estes problemas, um Sistema de Transporte Inteligente (ITS) é uma tecnologia ou plataforma que tem como objetivo criar novas soluções com o propósito de melhorar a qualidade ou tornar mais eficiente os sistemas de transportes. A comunicação interveicular é um dos principais recursos de um ITS, pois permite veículos compartilharem diferentes tipos de informação, desde segurança para prevenir acidentes ou reduzir os danos causados por eles, até informações sobre a saúde do condutor.

Segundo a ETSI *European Telecommunications Standards Institute*, uma estação ITS - *Intelligent Transportation System Station* pode ser qualquer dispositivo que implemente a arquitetura de comunicação ETSI ITS, como, por exemplo, um veículo, dispositivo pessoal ou uma estação fixa próxima a uma rodovia. Aplicações ITS têm finalidades diversas e operam na última camada da arquitetura de comunicação de uma estação ITS (ETSI, 2010-09a). Os veículos em alguns anos trocarão dados entre uma variedade de nós vizinhos, incluindo não apenas outros veículos, mas também através de dispositivos de comunicação pessoal de pedestres e ciclistas, cidades inteligentes, e-saúde e ambientes inteligentes. Usuários de veículos também terão assistência de vários dispositivos, como relógios inteligentes, óculos inteligentes e dispositivos com finalidade na área da saúde, por exemplo, medidor de pressão, de cansaço e estresse.

Futuramente, os carros processarão dados oriundos de muitas fontes e se comunicarão com diversos dispositivos. Várias tecnologias de comunicações estão sendo usadas em dispositivos IoT, como por exemplo, IEEE 802.15.4 e Bluetooth Low Energy ou Bluetooth Smart. Uma variedade de interfaces devem ser integradas ao veículo para aplicações ITS fazerem uso de dados oriundos de nós vizinhos, bem como de redes intraveiculares. Devido aos diversos elementos que integram os cenários apresentados, seria extremamente complexo e dispendioso criar aplicações ITS para se comunicarem com todos esses dispositivos IoT, que se comunicam através de protocolos e tecnologias de comunicação diferentes, além de fazerem uso de dados oriundos da rede Ethernet intraveicular. Portanto, a criação de uma ferramenta que possibilite a comunicação entre aplicações ITS, IoT e ECUs facilitaria a integração dessas tecnologias nos automóveis. Visto que nenhuma ferramenta integra as tecnologias SOME/IP, IoT e aplicações ITS juntas, assim como abordadas neste trabalho, o objetivo deste trabalho é propor uma solução para o cenário apresentado.

1.2 Objetivos

1.2.1 Objetivos Gerais

Objetivo da Pesquisa *O objetivo deste trabalho é desenvolver e avaliar uma extensão da arquitetura de uma estação ITS do padrão ETSI. Esta extensão permite que aplicações ITS se comuniquem com dispositivos de Internet das Coisas e receba dados de nós da Rede Ethernet intraveicular usando requisições por serviço com SOME/IP.*

1.2.2 Objetivos Específicos

- *Objetivo 1:* Avaliar o atraso fim-a-fim nos cenários de Internet das Coisas, rede Ethernet intraveicular e ambos juntos.
- *Objetivo 2:* Avaliar os resultados comparando se a extensão proposta da arquitetura nos diferentes cenários de aplicação, respeita os limites máximos de atraso fim-a-fim especificados por (TUOHY et al., 2015).

1.3 Apresentação da Dissertação

1. *Conhecimentos básicos:* Estudo cuidadoso de várias tecnologias elaboradas especificamente para ambientes veiculares e tecnologias que foram adaptadas para este cenário. Para estabelecer a intercomunicação proposta pela arquitetura, inicialmente são introduzidas informações necessárias para o entendimento dos capítulos subsequentes.
2. *Revisão da Literatura:* Com base no estado atual da arte e em soluções existentes no mesmo cenário de aplicação da extensão proposta da arquitetura, são apresentados detalhadamente os projetos mais relevantes e de propósitos equivalentes aos da arquitetura. Por fim, são comparadas as características de cada solução com a arquitetura do projeto, desta forma, visando evidenciar as vantagens de usar a extensão proposta.
3. *Arquitetura:* A proposta de extensão da arquitetura é apresentada em contextos de aplicação. Primeiramente, é explicado como é o processo de comunicação de aplicações veiculares com nós da rede Ethernet automotiva. Será apresentada também a arquitetura no contexto de Internet das Coisas, detalhando cada etapa de comunicação, desde a requisição por abertura de conexão até a troca de dados de aplicações veiculares com dispositivos IoT. Por fim, a arquitetura completa é analisada, apresentando os dois contextos citados anteriormente integrados.

4. *Processo de coleta de dados:* É apresentado como é o processo de coleta de dados e como foram coletados através das simulações das comunicações utilizando a extensão proposta da arquitetura.
5. *Métricas e Ferramentas:* É mostrada a métrica utilizada para analisar e validar a arquitetura para o desenvolvimento e execução do projeto.
6. *Experimentos:* Após a implementação da extensão proposta da arquitetura, é mostrado como foram executados experimentos nos contextos de Redes Veiculares, Internet das Coisas e no cenário integrado.
7. *Avaliação dos Resultados:* São detalhados os resultados obtidos, os quais são validados através de testes de hipótese.

1.4 Considerações Finais

Este capítulo apresentou a motivação do desenvolvimento deste projeto através da caracterização do problema, além disso, uma visão geral da estruturação e do conteúdo deste projeto.

No próximo capítulo, são explicados os conhecimentos fundamentais para a compreensão do conteúdo que será abordado neste trabalho e o estado atual da arte. Por fim, é feita uma comparação entre a extensão proposta da arquitetura ITS-S e outros projetos.

2 Background e Trabalhos Relacionados

Este capítulo apresenta uma descrição geral e o estado da arte de Redes Veiculares e de Internet das Coisas. São apresentadas algumas tecnologias fundamentais para o entendimento do projeto, como por exemplo, Sistemas Operacionais de IoT, *middlewares* e padrões automotivos. Por fim, é mostrado um comparativo entre a extensão proposta da arquitetura ITS-S e os trabalhos relacionados.

2.1 *Internet das Coisas*

IoT *Internet of Things* é um termo utilizado para descrever o paradigma tecnológico no qual os objetos do dia a dia estariam conectados e acessíveis a outros objetos e pessoas através da Internet. O termo “Coisa“ no contexto de IoT refere-se a qualquer elemento ou dispositivo que possua processamento e capacidade de comunicar-se, por exemplo, com eletrodomésticos, automóveis, monitores cardíacos, *wearables* (dispositivos vestíveis), lâmpadas, óculos inteligentes, relógios inteligentes, celulares e etc. Estes elementos atuam auxiliando as tarefas diárias, criando oportunidades e mudando a economia.

A ideia básica deste conceito é a presença de diversos dispositivos com diferentes finalidades agindo de modo inteligente, se comunicando de maneira constante e independente com outros dispositivos, além de interagir com indivíduos, tornando assim o ambiente responsivo (ATZORI; IERA; MORABITO, 2010).

A IoT representa uma solução que traz vários benefícios à sociedade, possibilitando a automação de tarefas e processos, ajudando indivíduos e empresas. O escopo de aplicações utilizando a tecnologia de IoT é extremamente vasto. Devido à grande quantidade de dados de diferentes fontes, poderão existir aplicações de que apenas necessitaremos no futuro.

A redução do consumo de energia, recursos e tempo são fatores essenciais para empresas de IoT se destacarem no mercado. Algumas das principais aplicações IoT e seus benefícios no contexto industrial são apresentados em (JESCHKE CHRISTIAN BRECHER, 2016). Destacam-se a reconfigurabilidade de equipamentos, permitindo o rápido aumento da escala de produção e o gerenciamento de mudanças, a avaliação da produção para a melhoria dos sistemas de produção na próxima geração, a manutenção preditiva e detecção automática de falhas, dentre outros. Fora do escopo do setor industrial, a incorporação de IoT nas vidas dos indivíduos, além de automatizar tarefas diárias, traz benefícios principalmente na saúde (YUEHONG et al., 2016), educação (MOREIRA et al., 2017) e segurança (FRAGA-LAMAS et al., 2016).

De acordo com (RIVERA; MEULEN, 2014), o setor automotivo apresentou uma maior taxa de crescimento com relação à quantidade de dispositivos IoT usados, comparado com setores voltados para empresas e consumidores. O relatório também afirma que mais de 25 bilhões de “Coisa” conectadas serão utilizadas em 2020. Uma expectativa mais ambiciosa foi apresentada por (ABIRESEARCH, 2013), com mais de 30 bilhões de dispositivos esperados no mesmo período.

Constata-se, então, que a IoT abrirá várias oportunidades para um grande número de novas aplicações, especialmente no cenário automotivo. Uma gama de soluções podem ser criadas usando desde dispositivos internos, que poderiam monitorar os condutores, até externos, que auxiliarão na condução do automóvel. Alguns tipos de dispositivos já existentes podem ter uma nova aplicação em veículos, como por exemplo, um relógio inteligente que analisa os batimentos cardíacos ou situações de estresse e envia dados para o carro ou óculos inteligentes que mostram informações sobre a rota.

Muitos protocolos de comunicação idealizados para outros ambientes tem-se apresentado como soluções para o universo IoT, como por exemplo, IEEE 802.15.4 e Bluetooth de Baixa Energia (BLE *Bluetooth Low Energy*) ou Bluetooth Smart. Atualmente, o Bluetooth é um dos protocolos mais utilizados para troca de dados entre dispositivos pessoais, no entanto, o IEEE 802.15.4 é amplamente utilizado em Redes de Sensores Sem Fio (Wireless Sensor Networks, WSN).

O Protocolo de Internet (IP *Internet Protocol*) é necessário para qualquer conexão com a Internet. Considerando o potencial para o crescimento exponencial do número de sensores com a IoT, o IPv4 não é mais capaz de dar suporte à quantidade de dispositivos devido à limitação de endereços. Desta forma, o IPv6 oferece um esquema de endereço altamente escalável, indicado para o cenário de IoT. Porém, o IPv6 requer um suporte de tamanhos de pacotes maiores do que os protocolos comuns de IoT, tornando-o inadequado para dispositivos de baixa potência devido à baixa capacidade de energia. O protocolo IPv6 sobre Redes de Baixa Potência WPAN (6LoWPAN IPv6 *over Low Power WPAN*) (KUSHALNAGAR, 2007) foi criado para definir encapsulamento e mecanismos de compressão de cabeçalho, permitindo o amplo uso de IPv6 em sensores. Ambos BLE e IEEE 802.15.4 já possuem padrões RFCs para operar através de IPv6 usando 6LowPan (MONTENEGRO, 2007) (NIEMINEN, 2015).

Há diversas plataformas de IoT disponíveis que oferecem várias funcionalidades para aplicações de Internet das Coisas. Essas plataformas podem operar desde de drivers de hardwares, camadas de rede, até serviços em nuvem (KAA, 2017) (MBED, 2013) (RIOT, 2013) (DUNKELS et al., 2012). Contiki (DUNKELS et al., 2012) é um sistema operacional de código aberto para a Iot, dá suporte totalmente ao IPv6 padrão e 6LowPan, conectando microcontroladores de baixo custo e de baixa potência com a Internet. Ele foi criado por uma equipe de desenvolvedores de diversos países e possui contribuições de grandes empresas como Atmel e Cisco, dentre outros. O Contiki dá suporte a muitas plataformas

de hardware, como o Nordic nRF52, para Bluetooth Low Energy e Texas Instruments cc2538, para IEEE 802.15.4.

Embora os dados gerados por dispositivos IoT internos e externos aos veículos sejam importantes para muitas aplicações, como as baseadas em saúde e assistentes de condução, propagar essas informações para outros carros, infraestrutura e a Internet torna o escopo de aplicações veiculares quase ilimitado. Exemplos incluem tráfego seguro e eficiente, mobilidade, entretenimento, assistente de estacionamento, assistentes de condução, controle remoto diagnósticos e telemática para veículos autônomos (UHLEMANN, 2015).

2.1.1 Sistema Operacional de IoT

Um Sistema Operacional (S.O. *Operational Systems*) possui várias funções. Pode-se destacar o gerenciamento dos recursos de um sistema e o fornecimento de serviços de hardware de maneira transparente ao usuário. Qualquer S.O precisa de um gerenciamento de recursos eficiente, especialmente um S.O de IoT, devido às limitações de hardware. Outra característica importante de um S.O. de IoT é a conectividade, pois o principal objetivo de IoT é fazer objetos se comunicarem. Assim, tornar possível a comunicação com objetos através de protocolos distintos é um fator relevante em um S.O. de IoT.

Existem vários sistemas operacionais de IoT. Embora cada S.O. possua características que se destacam dos demais, alguns fatores são interessantes antes da escolha, tais como: Comunidade de desenvolvedores, comunicatividade, funcionalidades, consumo de energia, eficiência, segurança, escalabilidade e documentação.

2.1.1.1 Contiki

O Contiki se destaca de outros S.Os de IoT por vários motivos. O Contiki dá suporte ao Protocolo de Aplicação Restrita (CoAP *Constrained Application Protocol*), que é um protocolo para dispositivos ou redes com restrições de recursos no cenário de Internet das Coisas. Na camada de rede, o Contiki pode utilizar o protocolo de roteamento para Redes de Baixa Potências e com Perdas (RPL), IPv6 e 6LoWPAN na camada de adaptação. Por usar o simulador Cooja, é possível emular drivers de diversas plataformas de placas, por exemplo, micaz (MICAZ, 2011), sky (SKY, 2006) e avr-raven (AVR-RAVEN, 2008).

2.1.1.2 Arm mbed

A plataforma ARM mbed (MBED, 2013) fornece um sistema operacional de Internet das Coisas. Suas principais características são a quantidade de funcionalidades disponíveis, a conectividade e os grandes parceiros da plataforma, por exemplo, IBM, TOSHIBA, ACCENTURE, NXP e Ericsson. Esta plataforma possui suporte a IPv6 e 6LowPan, serviços na nuvem, ferramentas e ecossistema de desenvolvimento para implantação de soluções de IoT, no entanto, não fornece nenhuma ferramenta de simulação de sensores.

2.2 Redes Veiculares

Segundo (LIN et al., 2007), redes veiculares são divididas em redes intraveiculares, comumente chamadas de redes automotivas, e redes interveiculares. As redes intraveiculares são redes internas ao veículo e tratam da comunicação entre ECUs, sensores e interfaces de interação com usuários. As redes interveiculares são formadas pela comunicação entre veículos e outros veículos ou infraestruturas.

2.2.1 Redes Intraveiculares

Desde meados dos anos 90, a utilização de ECU *Electronic Control Unit* em veículos tem aumentado devido à modernização dos automóveis. A necessidade de comunicação entre as ECUs nos diversos domínios automotivos fez com que diversas interconexões fossem feitas, o que elevou o peso e o custo dos veículos. Dentre essas interconexões destacam-se a Rede de Área do Controlador (CAN *Controller Area Network*), Rede de Interconexão Local (LIN *Local Interconnect Network*) e Transportes de Sistemas Orientados à Mídia (MOST *Media Oriented Systems Transport*).

Os veículos modernos oferecem aos condutores uma ampla gama de funcionalidades de segurança, conforto e assistência. Diferentes tecnologias de rede nos veículos para diferentes requisitos tornarão o sistema de comunicação interna do carro ainda mais complexo que os atuais, com requisitos de maior largura de banda. Devido a esse aumento de recursos nos veículos, criou-se a necessidade de uma rede mais escalável e robusta. A Ethernet, que é amplamente usada em escritórios, residências e indústrias, mostrou-se adequada para o cenário automotivo. Uma das principais motivações para o uso da Ethernet em veículos é que seus protocolos estão disponíveis e, como foi construídos em camadas, é possível alterar ou adaptar protocolos sem gerar grandes impactos na pilha (MATHEUS; KÖNIGSEDER, 2017). A Ethernet pode substituir e coexistir com diversas redes automotivas atuais, o domínio mais perceptível é o de entretenimento, pois exige uma grande largura de banda (SOARES et al., 2015) (LAJMI; ALIM; AJILI, 2013).

Os novos recursos e o aumento da complexidade dos softwares automotivos impulsionaram a criação de arquiteturas de softwares padronizadas para os futuros carros. A Arquitetura de Sistema Aberto Automotivo (AUTOSAR *AUTomotive Open System ARchitecture*), é uma arquitetura de software criada em uma parceria entre grandes fabricantes de automóveis e de peças. Tem como objetivo padronizar os softwares para ECUs, definindo uma arquitetura, interfaces de comunicação, metodologia e testes (AUTOSAR, 2009). GENIVI é outra aliança sem fins lucrativos de grandes empresas empenhadas em impulsionar a ampla adoção do software livre para os domínios de informação e entretenimento veicular (GENIVI, 2012).

Existem também *middlewares* com propósitos mais específicos, um exemplo é o Middleware Escalável Orientado a Serviço sobre IP (SOME/IP - Scalable service-Oriented

MiddlewarE over IP). Este *middleware* é uma solução para prover comunicação orientada a serviços em uma rede Ethernet automotiva. O SOME/IP atua nos domínios cujos dados não possuem altas restrições de tempo (SOME/IP, 2012).

2.2.1.1 CAN

O CAN foi uma das primeiras redes automotivas e ainda existe até os dias atuais. Não é usado apenas em automóveis mas também em vários seguimentos industriais, por exemplo, automação e aeroespacial. É um padrão que utiliza barramento, logo, todas as ECUs são conectadas por um único meio de comunicação. Além disso, o uso de dados é baseado em mensagens geradas por broadcast. Devido a troca de mensagens ser através de broadcast, CAN utiliza um sistema de priorização de mensagens que serão enviadas ao barramento, este sistema é chamado Arbitragem. Mensagens com maior prioridade, ou seja, menor valor identificador possuem prioridade no envio. CAN é um sistema simples de fio e opera com dois pares de cabos, por esse motivo, é capaz de funcionar mesmo que um cabo pare de funcionar, isso devido a redundância (MATHEUS; KÖNIGSEDER, 2017). Atualmente, é dividido em três versões diferentes.

- CAN de Alta Velocidade (HS CAN - *High-Speed CAN*), usado para comunicação com taxas de dados de até 1 Mbps. É uma das redes automotivas mais usadas e incluem sistemas antibloqueio de freio, módulos de controle de motores e sistemas de emissões
- CAN de Baixa Velocidade (LS CAN - *Low-Speed CAN*), usado para troca de dados com taxas até 125 Kbps. Inclui principalmente dispositivos relacionados a conforto, por exemplo, altura do vidro e ajustes de banco.
- CAN com Taxa de Dados Flexíveis (FD CAN - *CAN with Flexible Data rates*), é a mais nova geração de comunicação CAN de alta velocidade, podendo atingir taxas de até 8 Mbits/s.

2.2.1.2 LIN

Algumas aplicações veiculares são simples o suficiente para serem compostas por apenas um sensor e um cabo de conexão, além disso, essas aplicações requerem redes pequenas e com baixa taxa de transmissão. São exemplos dessas aplicações: vidros elétricos, travamento central de portas, iluminação, ar-condicionado, sensor de chuva, regulação de bancos, entre outros que requerem baixo poder de processamento e de recursos de comunicação. A utilização de CAN nesses cenários estaria subaproveitando a rede, devido a este motivo, o barramento LIN foi desenvolvido para criar um padrão de baixo custo de comunicação. O LIN é um sistema de rede pequeno e lento utilizado como uma opção barata para subredes CAN. Assim como CAN, o barramento LIN também é bastante usado

e, mesmo com a chegada da Ethernet automotiva, por ser uma opção de baixo custo, é possível que esta rede coexista com a Ethernet e outras com funções específicas. Uma rede LIN foi desenvolvida para suportar até 16 ECUs compartilhando o mesmo barramento, sua taxa de transmissão chega até 19.2 kbps e utiliza um sistema de Mestre e Escravos para o acesso ao canal. Em um sistemas Mestre/Escravo, um nó da rede é eleito o Mestre e os demais Escravos, quando um nó precisa enviar dados através do barramento, ele precisa ser autorizado pelo Mestre. Por ser um sistema de comunicação simples, todos os nós da rede recebem todas as informações, assim como CAN, mas utilizam apenas as informações referentes ao próprio nó (MATHEUS; KÖNIGSEDER, 2017).

2.2.1.3 MOST

No final dos anos 90, começaram a surgir demandas por aplicações multimídias nos veículos, não apenas sistemas de áudio, mas também, por exemplo, sistemas de navegação. Devido a essa necessidade de altas taxas de dados, alguns fabricantes automotivos, como BMW, Daimler e Oasis fundaram a MOST co, para o desenvolvimento de uma rede automotiva para aplicações multimídia de alta velocidade. MOST é uma tecnologia de rede de alta velocidade, desenvolvido para sistemas no domínio de informação e entretenimento. MOST suporta até 64 dispositivos compartilhando o mesmo barramento em uma configuração de rede em anel. Dispositivos podem ser anexados e removidos da rede MOST facilmente, pois utilizam um sistema de adesão Plug and Play. MOST utiliza um sistema de prioridade de envio de mensagens Mestre/Escravo, uma mensagem é enviada por um nó requerendo prioridade no envio de dados, assim, os demais nós são sincronizado para manter a prioridade. MOST oferece taxas de transmissão que variam de 25 Mbits/s até 150 Mbits/s dependendo da versão que estiver sendo utilizada. Com a chegada da Ethernet em veículos, é esperado que MOST seja totalmente substituída dos veículos. Isto se deve ao fato da Ethernet atuar diretamente nos domínios em que MOST é atualmente usado (MATHEUS; KÖNIGSEDER, 2017).

2.2.1.4 SOME/IP

O SOME/IP foi projetado inicialmente para adequar dispositivos de tamanhos e sistemas operacionais diferentes. Isso inclui dispositivos pequenos como: câmeras, AUTOSAR (AUTOSAR, 2009) ou telemáticos. O SOME/IP dá suporte aos recursos de diferentes domínios automotivos, permitindo que substitua, por exemplo, cenários em que MOST é utilizado.

Na pilha de comunicação TCP/IP o SOME/IP está localizado entre as camadas de Aplicação e a camada de Transporte, ver Figura 1. As mensagens da aplicação são encapsuladas e recebem os cabeçalhos necessários para os nós se comunicarem utilizando o *middleware*.

Em sistemas que não possuem restrições de tempo, por exemplo, sistemas não críticos, a utilização de comunicação baseada em serviços, como o SOME/IP, pode ser uma

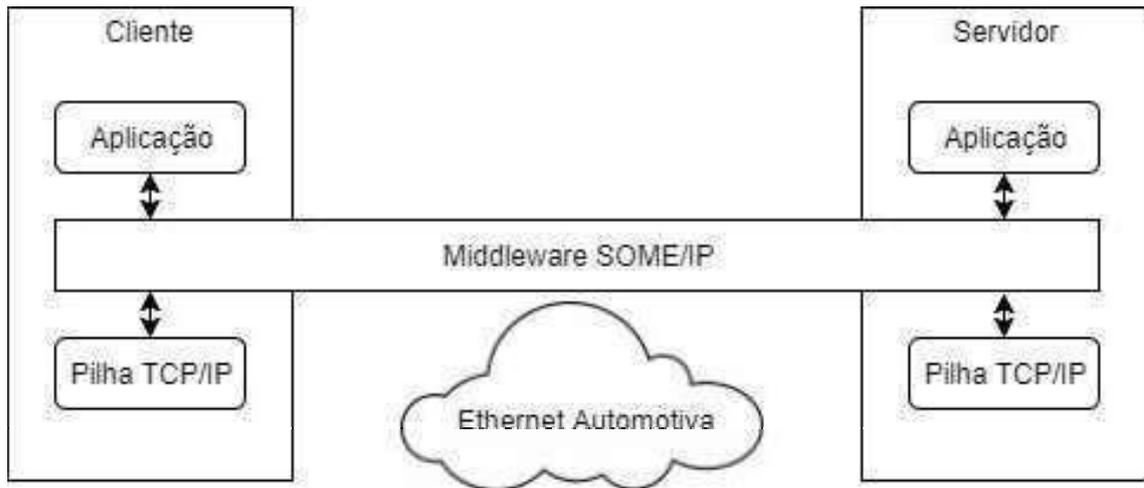


Figura 1 – Comunicação entre cliente e servidor utilizando SOME/IP (AGLAIA, 2014)

solução eficiente e reduzir os custos de cabos e hardwares, sendo implantada na Ethernet automotiva (SEYLER et al., 2015).

A utilização do SOME/IP com a comunicação baseada em serviços em uma rede Ethernet intraveicular pode trazer vários benefícios:

- Com a comunicação baseada em serviços, nós da rede, como ECUs, podem conter mais recursos computacionais, e utilizando o conceito de *middlewares* orientados a mensagens, fornecer não apenas um serviço, mas vários. Além disso, o SOME/IP permite a entrega de mensagens unicast, multicast e broadcast.
- Com a utilização de menos ECUs, diminuiria a complexidade da rede e, portanto, reduziria a quantidade de switches nas futuras redes Ethernet automotivas.
- A comunicação não é essencialmente broadcast, como acontece, por exemplo, com CAN. A comunicação é direcionada aos nós que fazem a requisição e aos que esperam respostas, diminuindo a sobrecarga na rede.
- Ao contrário da abordagem utilizada por alguns padrões automotivos como TTEthernet (STEINER et al., 2011) e AVB (AVB, 2011), com a comunicação baseada em serviços do SOME/IP, os nós não precisam ser sincronizados durante a inicialização da rede.
- Nós que utilizam o SOME/IP podem subscrever apenas uma vez em serviços de outros nós, assim, sempre que mensagens forem geradas para aquele serviço, todos os subscretores receberão a mensagem.

A comunicação entre um cliente e servidor no *middleware* SOME/IP pode ser realizada das duas maneiras mais comuns de *middlewares* orientados a mensagens, publish/subscribe e request/response, ver Figura 2. A primeira através de publish/subscribe, nesse

método o publisher, nó da rede que fornece serviços, publica seus serviços em um módulo chamado de Descoberta de Serviço (*SD Service Discovery*), assim, os subscribers, nós da rede que pretendem consumir dados de publishers, inscrevem ou subscrevem aquele serviço. Desta forma, a cada novo evento emitido por um determinado serviço, todos os assinantes daquele serviço serão notificados.

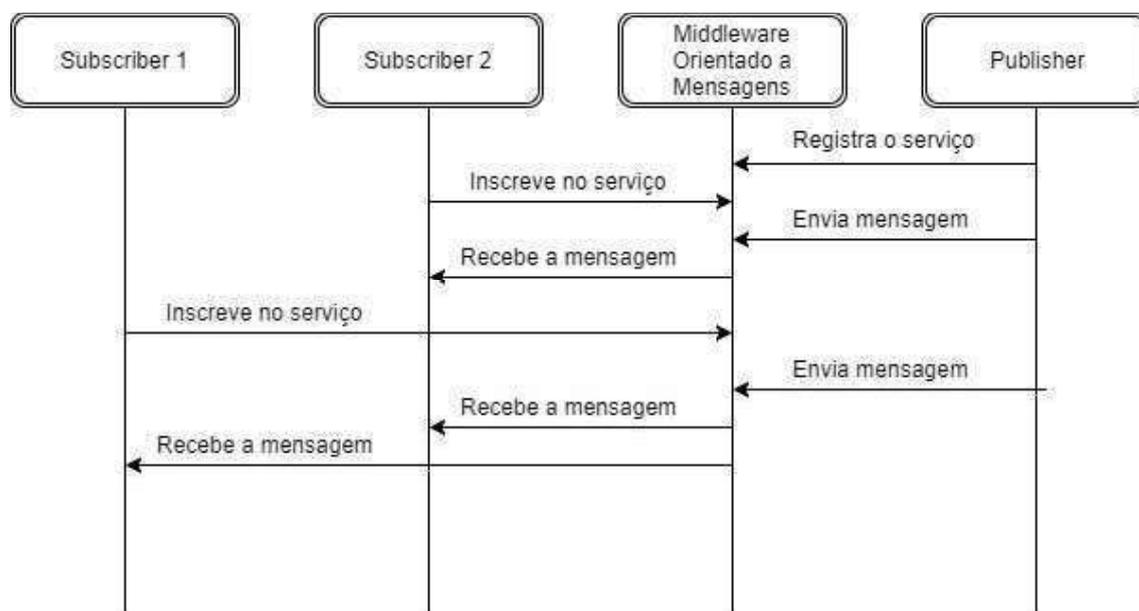


Figura 2 – Diagrama de sequência do método publish/subscribe

No método publish/subscribe, um nó publisher da rede pode fornecer diversos serviços, assim, subscribers poderão subscrever em serviços diferentes oferecidos por um mesmo nó. A vantagem desta abordagem em veículos é que aumentando a robustez dos recursos de uma ECU, seria possível concentrar mais serviços em menos hardwares, diminuindo a quantidade de ECUs na rede intraveicular.

O segundo e mais simples método, a troca de mensagens entre cliente e servidor é de um para um. O cliente faz a requisição de um serviço no servidor e este responde com o resultado. Este método é chamado de request/response.

O SOME/IP foi inicialmente criado com objetivo de prover comunicação baseada em serviços e logo foi integrado como componente nas arquiteturas de *middlewares* automotivos, ver Figura 3.

Além do Autosar, SOME/IP também foi integrado ao *middleware* GENIVI, o Vsomeip (VSOMEIP, 2015) é uma implementação aberta e disponibilizada na Internet para desenvolvedores poderem contribuir com o GENIVI.

A comunicação utilizando SOME/IP consiste no envio de mensagens entre dispositivos utilizando o protocolo IP.

Na Figura 4 é mostrada a mensagem do SOME/IP. A mensagem é formada pelo Header e pelo Payload. Cada campo da mensagem é definido como:

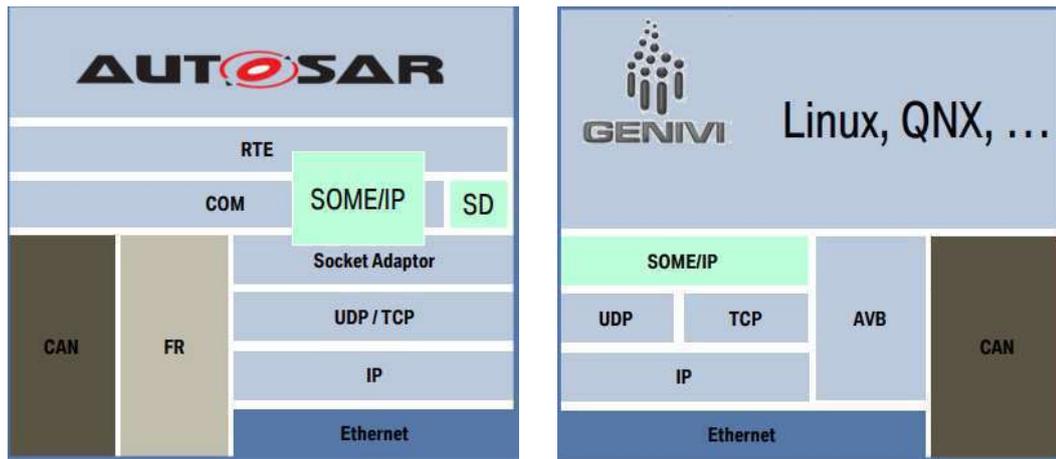


Figura 3 – Middleware SOME/IP nas arquiteturas do Autosar e GENIVI (VöLKER, 2014)

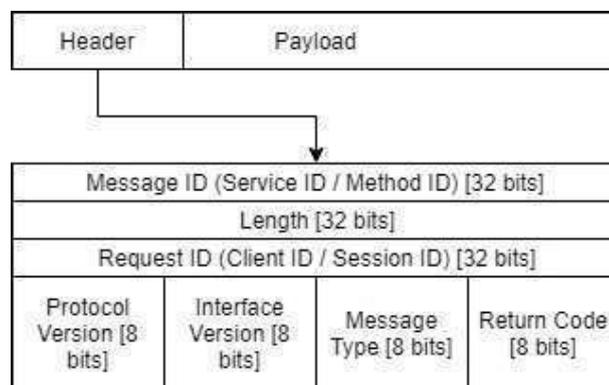


Figura 4 – Mensagem SOME/IP

- ID do serviço: identificador único de cada serviço;
- ID do Método: identificador do método, varia de 0 à 32.767 para métodos e 32.768 a 65.535 para eventos;
- Length: tamanho do Payload em Bytes;
- ID do Cliente: identificador único de cada cliente no veículo;
- ID da Sessão: identificador de cada sessão. Este valor é incrementado em cada chamada;
- Versão do Protocolo: versão do protocolos, definido como código de indentificação hexadecimal 0x01;
- Versão da Interface: versão principal da interface do serviço;
- Tipo da Mensagem: tipo da mensagem, podendo ser de dez tipos diferentes, os principais são:

- REQUEST (0x00): mensagem de requisição;
- RESPONSE (0x80): mensagem de resposta;

2.2.2 Redes Interveiculares

A degradação ambiental, dificuldade de tráfego nas grandes cidades e o aumento dos acidentes no trânsito representam apenas alguns dos transtornos decorrentes da utilização dos veículos automotores. Esta série de problemas fomentou o interesse de grandes indústrias e de pesquisas da área acadêmica na tentativa de reverter os danos a curto e longo prazo causados. Devido a essa concentração de interesses, várias soluções foram propostas e outras estão sendo aprimoradas, como: Sistemas de Transportes Inteligentes (ITS), comunicação entre veículos, carros elétricos e carros autônomos.

Um ITS é uma tecnologia ou plataforma que tem como objetivo criar novas soluções com o propósito de melhorar a qualidade ou tornar mais eficiente os sistemas de transportes. Através do uso de sensores, Circuitos Fechados de Televisão (CCTV *Closed-Circuit Television*), processamento de dados, nuvem, comunicação entre veículos e outras tecnologias, os ITS podem fornecer aplicações avançadas que monitoram ou gerenciam os transportes tornando-os mais inteligentes e melhorando seus desempenhos.

A comunicação entre veículos permite estes compartilharem diferentes tipos de informação, desde segurança para prevenir acidentes ou reduzir os danos causados por eles, até informações sobre a saúde do condutor. Uma Rede Veicular Ad Hoc (VANET *Vehicle Ad Hoc Networks*) é uma rede em que cada nó é equipado com tecnologia de comunicação Wireless e cujo objetivo final é servir como um canal de comunicação para trocas de dados entre aplicações veiculares. Esses dados podem incluir quaisquer tipo de informação que trafega em um veículo, como velocidade, localização, condições climáticas, direção, dados oriundos dos Sistemas Avançados de Assistência ao Motorista (ADAS) e, até mesmo, dados de dispositivos de IoT que podem estar conectados com veículos.

Atualmente existem diversas aplicações veiculares já propostas (WHAIDUZZAMAN et al., 2014) (ARANITI et al., 2013) (VEGNI; AGRAWAL, 2016), contemplando desde monitoramento de tráfego até aplicações de entretenimento, como jogos interativos, vídeos e compartilhamento de dados.

Os benefícios dessas aplicações citadas são possíveis devido a duas grandes frentes de pesquisa para comunicação em redes veiculares, a primeira agrupa tecnologias amplamente conhecidas como 3G/4G/LTE/5G e a segunda o Comunicação Dedicada de Curto Alcance (DSRC *Dedicated Short Range Communication*). O DSRC é protocolo open source similar ao WiFi com sinal de curta e médio alcance e resistente a interferências, projetado especificamente para o cenário automotivo, em que a transferência de dados ocorrem sob altas velocidades, curvas, obstáculos e interferências climáticas. As DSRCs que foram projetadas pelo Instituto de Engenheiros Elétricos e Eletrônicos (IEEE *Institute of Electrical and Electronics Engineers*) e possuem padrões diferentes para Europa e América do Norte,

são eles Acesso sem fio em ambientes de veículos (WAVE *Wireless Access in Vehicular Environments*) (WAVE, 2010) e ETSI ITS G5 (G5, 2012) respectivamente.

A comunicação nessa rede segue, geralmente, dois modelos: Veículo para Veículo (V2V *Vehicle-to-Infrastructure*), ocorre quando automóveis se comunicam diretamente entre si e Veículo para Infraestrutura (V2I *Vehicle-to-Vehicle*), quando veículos se comunicam com estações de comunicação comumente localizadas próximas às vias. Nos últimos anos, com disseminação de IoT, vários conceitos como Tráfego Inteligente e Cidades Inteligentes (CI) se tornaram mais populares. Devido à importância dos automóveis no cotidiano, veículos sempre receberam a atenção de pesquisadores e indústrias, nesse contexto, era de se esperar que Redes Veiculares e IoT convergissem em algum momento. Esse conceito é, normalmente, chamado de Veículo para Qualquer coisa (V2X *Vehicle-to-Everything*). V2X incorpora os tipos de comunicação tradicionais V2V e V2I, além de outros conceitos que não foram amplamente disseminados como Veículo para Pedestres (V2P *Vehicle-to-Pedestrian*), Veículo para Dispositivos (V2D *Vehicle-to-Device*) e outros.

Nos últimos anos algumas áreas como carros elétricos e carros autônomos estão despertando bastante interesse em grandes empresas como Google, Apple, BMW, Nissan e, conseqüentemente, fomentando pesquisas acadêmicas. Devido a isso, uma nova nomenclatura para comunicação entre carros e Coisas surgiu ou foi atualizada de V2X para os chamados Carros Conectados. Segundo (FORNI; MEULEN, 2016) é prevista uma produção de 61 milhões de carros conectados apenas em 2020, totalizando um quarto de bilhão de veículos que possuem módulos sem fio com capacidade de se comunicar nesse mesmo ano, além disso, é esperado que existam 25 bilhões de dispositivos de internet das coisas (MEULEN; RIVERA, 2015).

2.2.2.1 ETSI TC ITS

ETSI TC ITS (ETSI, 2010-09a) é um conjunto de padrões criados pela ETSI aplicados a ITS. A arquitetura de estações ITS (ITS-S) no padrão ETSI foi desenvolvida para que qualquer dispositivo que a implemente possa se comunicar com outros que tenha a mesma arquitetura. A arquitetura de referência de uma estação ITS 5 segue os princípios do modelo OSI para protocolos de comunicação em camadas e o estende para inclusão de aplicações de ITS. A camada de Acesso da arquitetura de uma estação ITS representa as camadas de Enlace e Física do modelo OSI, as camadas de Rede e Transporte são equivalentes nas duas abordagens e, por fim, as camadas de Facilities e Aplicação, representam as camadas de Sessão, Apresentação e Aplicação no modelo OSI.

A camada de Facilities fornece suporte a aplicações de ITS, que podem compartilhar funções genéricas e dados de acordo com seus respectivos requisitos funcionais e operacionais. Esta camada é composta por um conjunto de funcionalidade que fornecem algum tipo de serviço para a camada de Aplicação. Cada funcionalidade, geralmente, é representada por uma Facility. São exemplos de funcionalidades da camada Facilities (ETSI,

2010-09a):

- Suporte a aplicações.
- Suporte de informações.
- Suporte à comunicação.
- Suporte de sessão.

A camada de Facilities possui várias funcionalidades especificadas, como por exemplo, gerenciamento de classes de tráfego, suporte à Interface Humano Máquina (HMI *Human Machine Interface*), controle de congestionamento, referências de localização e acesso seguro (ETSI, 2010-09b). Apesar de estas funcionalidades estarem definidas, a especificação técnica da camada de Facilities (ETSI, 2013) adverte que novas funcionalidades podem ser criadas e integradas a esta camada.

No ETSI, dois tipo de mensagens são padronizados, são elas, Mensagem de Conscientização Cooperativa (CAM *Cooperative Awareness Message*) (CAM, 2011) e Mensagem de Notificação Ambiental Descentralizada (DENM *Decentralized Environmental Notification Message*) (DENM, 2010). CAM são mensagens periódicas e fornecem informações de presença, posições, bem como o status básico da comunicação de estações ITS para estações ITS vizinhas. Ao contrário de CAM, DENM são mensagens usadas em situações que gatilhos precisam ser disparados, ou seja, são mensagens baseadas em eventos.

Uma vez que aplicações conseguem acesso aos dados do veículo e de outros próximos, e de dispositivos IoT, a disseminação desses dados, criam oportunidades ilimitadas de aplicações automotivas. A comunicação entre automóveis na arquitetura de estações ITS, ver Figura 5, pode ser efetuado utilizando tanto mensagens periódicas ou por evento, são elas, respectivamente, CAM e DENM. Uma vez que a estrutura das mensagens CAM e DENM são diferentes, duas Facilities foram criadas para manipular estas mensagens. CaService e DenmService tratam o envio, reenvio e recebimento das respectivas mensagens.



Figura 5 – Arquitetura ETSI ITS-S (ITS-S, 2010)

2.3 Trabalhos Relacionados

Atualmente, IoT e redes veiculares estão recebendo bastante atenção tanto na comunidade acadêmica quanto da indústria. Várias aplicações mesclando essas duas tecnologias foram propostas.

(GENG; CASSANDRAS, 2013) propôs utilizar a infraestrutura existente de detecção de vagas para estacionamento e criou um sistema para reserva de vagas. Quando uma vaga é desocupada, o carro mais próximo recebe o direito de ocupá-la e através do ID RFID do veículo a cancela do estacionamento libera a entrada do automóvel. Estações Unidades do Lado da Estrada (RSU *Roadside Unit*) são responsáveis pelas notificações que o sistema envia e requisições feitas pelas Unidades de Bordo (OBU *On Board Units*). Apesar da utilidade, essas aplicações são específicas para determinado problema e limitadas a poucos protocolos de sensores.

ICSI (CONSORTIUM, 2015) foi um projeto que integrou IoT e ITS cooperativo, propondo uma arquitetura que interconecta Redes de Sensores Sem Fio (WSN *Wireless Sensor Networks*), dispositivos pessoais e estações ITS. A conexão entre esses ambientes é possível através de gateways que recebem os dados e representam a inteligência da arquitetura. Esses elementos hospedam a plataforma para as aplicações de ITS. A arquitetura também provê comunicação Máquina-para-Máquina (M2M *Machine-to-Machine*) e 6LoWPAN/CoAP utilizando rede com IPv6. Embora o ICSI receba dados de WSN através dos gateways, não é demonstrado como a informação é adquirida dos sensores, visto que seriam necessárias múltiplas interfaces para protocolos de comunicação diferentes nas estações de ITS.

(CARIGNANI et al., 2015) propõem uma arquitetura que envolve a coleta de dados de IoT de diferentes protocolos através de múltiplas interfaces, rede intraveicular de origem

do barramento CAN *Controller Area Network* e comunicação entre estações ITS usado M2M. A comunicação entre estações ITS usam a pilha de rede ETSI ITS G5. A arquitetura também utiliza o sistema operacional de IoT, Contiki, assim é possível coletar dados de diferentes protocolos. Ao contrário da ICSI, a arquitetura de (CARIGNANI et al., 2015) não possui suporte ao protocolo IEEE 1609.x. Além disso, dados intraveiculares são coletados apenas da rede CAN, não sendo compatíveis com a rede Ethernet automotiva ou *middlewares* como AUTOSAR e GENIVI.

iDrive (QIN et al., 2013) é um serviço automotivo criado pela IBM para dar assistência e entretenimento aos condutores. Uma série de sensores e um GPS são utilizados para rastrear o veículo e as condições da pista, assim o sistema pode auxiliar o condutor na direção do veículo e ajudar a tomar melhores decisões sobre o percurso. Vários serviços são oferecidos através de sua plataforma em nuvem, como: jogos virtuais, conexões em redes sociais, manutenção de carro e informações de entretenimento.

(HE; YAN; XU, 2014) criou um sistema para serviços veiculares em nuvem e desenvolveu serviços de estacionamento inteligente e mineração de dados veiculares em nuvem. Na arquitetura proposta, cada veículo carrega um processador e um transmissor, como Zigbee e bluetooth. Esses módulos serão utilizados para se comunicar com a infraestrutura da vaga do estacionamento, identificando se a vaga foi preenchida ou não. Através da comunicação v2i, veículos podem ser notificados e requisitar informações sobre vagas. Apesar de usar v2i, nessa arquitetura v2v não é aproveitado, nem informações das redes internas do veículo.

Na CoMoSeF (LIN; REN; CERRITOS, 2013) é utilizado um modelo de serviço centrado em smartphones em estações de ITS. Dados coletados da OBD2 através de um módulo Bluetooth são enviados a um smartphone com um sistema operacional Windows Phone. O smartphone envia dados a uma estação veicular de ITS que repassa esses dados para a central ITS, composta por servidores que processam as informações, através de v2v e v2i. A arquitetura proposta é limitada ao protocolo de comunicação ETSI ITS G5 para v2v e v2i. Uma vez que o smartphone recebe os dados dos barramentos do veículo, essa informação poderia ser enviada à central ITS pelo próprio aparelho. O modelo centrado em smartphones, neste caso, não parece ser apropriado.

Segunda a Tabela 1, é possível comparar a extensão proposta nesse trabalho com os demais projetos citados nesta sessão. Foram avaliados quatro itens, são eles, compatibilidade com *middlewares* Automotivos, se possui integração com redes intraveiculares, com redes interveiculares e comunicação através de V2X.

O projeto desenvolvido por (GENG; CASSANDRAS, 2013) não envolve nuvem e redes intraveiculares, logo não possui compatibilidade com nenhum *middleware* automotivo. A comunicação v2v e v2i fazem parte do projeto, porém não foi informado o protocolo de comunicação utilizado. Cada carro precisará de uma tag Identificação de Rádio Frequência (RFID *Radio-Frequency Identification*) para associar a vaga ao ID do veículo, logo, apenas

Tabela 1 – Análise dos trabalhos relacionados

Comparação dos trabalhos relacionados				
	Compatibilidade com <i>Middlewares</i> automotivos	Rede In-traveicular	Rede Inter-veicular	v2x
(GENG; CASSANDRAS, 2013)	x	x	Não informado	v
(CONSORTIUM, 2015)	x	x	v	v
(CARIGNANI et al., 2015)	x	v	v	v
(QIN et al., 2013)	x	x	x	v
(HE; YAN; XU, 2014)	x	x	Não informado	v
(LIN; REN; CERRITOS, 2013)	x	v	v	v
Extensão Proposta	v	v	v	v

RFID representa v2x. O escopo da arquitetura proposta em (CONSORTIUM, 2015) é mais ampla, apesar de não envolver redes e *middlewares* intraveiculares, a comunicação v2v e v2i pode ser feita tanto pelo IEEE 1609.x quanto ETSI ITS G5, possuindo também serviço de nuvem. Embora a arquitetura de (CONSORTIUM, 2015) informe a compatibilidade com WSN, não é informado como a coleta dos dados de sensores de diferentes protocolos seria feita. (QIN et al., 2013) propôs uma arquitetura que soluciona o problema de multi interfaces no veículo para coletar dados de dispositivos IoT com protocolos diferentes, foi utilizado o sistema operacional de IoT, Contiki, que dá suporte a vários protocolos. A comunicação v2v e v2i é feita apenas pelo ETSI ITS G5, não sendo compatível com IEEE 1609.x. Dados intraveiculares são coletados através do barramento CAN, contudo, não dá suporte a *middlewares* intraveiculares. iDrive (QIN et al., 2013) é um sistema focado em serviços em nuvem, não possui comunicação v2v, v2i e não tem acesso a redes intraveiculares. O iDrive é um serviço já disponibilizado e como a comunicação entre veículos e infraestrutura não é amplamente usado, logo não está no escopo atual desse trabalho. Contudo, possui suporte a diversas tecnologias de comunicação como Zigbee, Bluetooth, Wi-Fi, 3g/4g e redes por satélite. (HE; YAN; XU, 2014) propõe um sistema para serviços veiculares em nuvem. Embora uma das principais características do projeto seja a comunicação v2i, o autor não informa qual protocolo das camadas superiores ao IEEE 802.11p é utilizado, a arquitetura não suporta v2v, nem acesso a dados internos do veículo. Estações ITS do projeto CoMoSeF (LIN; REN; CERRITOS, 2013) não são compatíveis com o protocolo IEEE 1609.x, a coleta de dados dos barramentos do veículo são feitas por um módulo Bluetooth, logo a arquitetura não dá suporte a *middlewares* intraveiculares automotivos.

2.4 *Considerações Finais*

Este capítulo apresentou os conhecimentos fundamentais para a compreensão do estado atual das plataformas e arquiteturas que unem Internet das Coisas e Redes Automotivas. Foi apontado que muitas empresas como Google, Apple, BMW e UBER estão fazendo grandes investimentos nessa área, criando um cenário competitivo e aberto para novas pesquisas.

Com a convergência entre Internet das Coisas e Redes Automotivas, muitas oportunidades foram criadas e uma vasta quantidade de aplicações poderão existir no futuro. As principais áreas de aplicação são segurança, mobilidade e saúde.

No próximo capítulo, é explicada a arquitetura da extensão proposta e como os componentes dos cenários utilizados se comunicam.

3 Extensão Proposta da Arquitetura ITS-S

Este capítulo descreve a extensão proposta da arquitetura de uma estação ITS no padrão ETSI. Para um melhor entendimento, a arquitetura é apresentada em contextos, relacionados a cada área de aplicação em que a extensão proposta pode ser utilizada para se comunicar. Para prover intercomunicação entre aplicações veiculares e as áreas de IoT, redes intraveiculares e outros veículos, um conjunto de softwares e especificações foram escolhidos para fazer parte da arquitetura. Os softwares utilizados pela arquitetura são formados por outras arquiteturas de comunicação, *middlewares* e sistemas operacionais, todos de código aberto e disponíveis na Internet. As integrações entre a extensão proposta e os softwares citados acima, assim como a implementação de especificações, são apresentadas ao longo deste capítulo.

3.1 Visão Geral da Extensão Proposta

O capítulo anterior apresentou a arquitetura de estações ITS, ou ITS-S. A função proposta desta arquitetura é prover suporte a comunicação interveicular e entre veículos e dispositivos que implementem a arquitetura. Como a arquitetura de uma estação ITS não possui suporte à comunicação com dispositivos de IoT e redes intraveiculares, a extensão proposta neste projeto visa introduzir estas áreas de aplicação no escopo do padrão ETSI ITS.

Para que a comunicação entre ITS-S e dispositivos de IoT fosse possível, foi escolhido um S.O de IoT com o objetivo ampliar a quantidade de dispositivos comunicáveis com a arquitetura. O Contiki foi escolhido devido ao seu código fonte ser aberto, sua linguagem ser C/C++ e existir uma grande comunidade e fóruns de consulta na internet. Além disso, foram encontrados vários artigos que foram usados como base para a atual implementação da extensão (CARIGNANI et al., 2015), (SELIEM; ELSAYED; KHATTAB, 2014), (KOVATSCHEK; DUQUENNOY; DUNKELS, 2011).

Assim como foi escolhido um S.O de IoT, para que fosse coletada informações oriundas dos veículos, foi preciso escolher uma interface de comunicação com a rede interna do automóvel. Atualmente, a interface de comunicação mais conhecida é a Diagnóstico a Bordo II (OBD2 *On-Board Diagnostic II*), esta porta de comunicação permite que dados das redes internas ao veículo sejam lidos por dispositivos externos. Embora a OBD2 seja amplamente usada, esta porta não recebe dados oriundos da rede Ethernet automotiva. O cenário proposto neste trabalho visa atuar em automóveis com redes Ethernet, uma

vez que este padrão tende a ser predominante nos futuros veículos. Seguindo a padronização dos *middlewares* mais utilizados no mercado automotivos para redes intraveicular baseadas em Ethernet, AUTOSAR e GENIVI, foi escolhido o *middleware* de comunicação SOME/IP, através dele a arquitetura ITS-S foi adaptada para receber dados da rede Ethernet intraveicular.

A extensão proposta é aplicada na camada de Facilities da arquitetura de uma estação ITS no padrão ETSI. Duas Facilities foram criadas para prover intercomunicação entre as aplicações ITS, nós da rede Ethernet automotiva que utilizam SOME/IP e dispositivos de IoT: a SomeipFacility e a ContikiFacility.

Como mostrado na Figura 6, a arquitetura, além de permitir a intercomunicação entre IoT e a rede Ethernet, permite também que aplicações ITS se comuniquem com outras aplicações que estejam em outros veículos. A comunicação interveicular pode ser feita através de dois protocolos de mensagens especificados pela ETSI ITS, CAM e DENM.

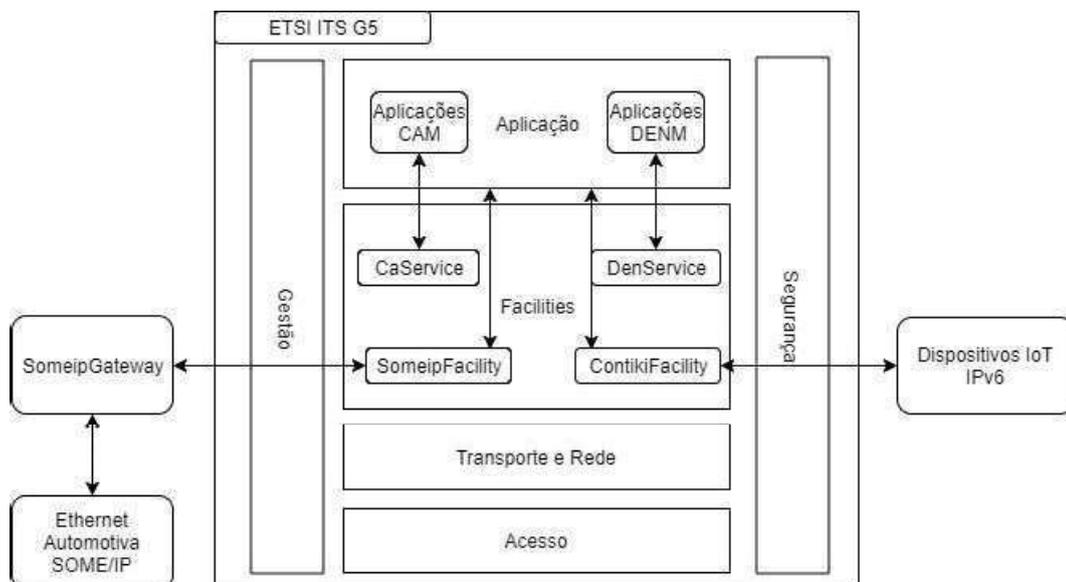


Figura 6 – Arquitetura de uma estação ITS com as extensões para comunicar-se com dispositivos de IoT e rede intraveicular

As requisições feitas por aplicações ITS para dispositivos de IoT ou nós da rede intraveicular são tratadas simultaneamente pelas Facilities responsáveis. As aplicações ITS podem ter a finalidade de comunicar-se com outras aplicações veiculares. Desta forma, é possível serem desenvolvidas para apenas transmitir mensagens do tipo CAM e DENM. Contudo, a arquitetura não cria limitações quanto à finalidade das aplicações, podendo utilizar-se de diferentes tipos de fontes de dados.

Através das mensagens DENM, aplicações podem enviar mensagens geradas por eventos para aplicações em outros veículos. A DenService é responsável por tratar mensagens geradas por aplicações que são emitidas por eventos. Ao contrário da DENM, a CAM é uma mensagem que possui um período definido e são enviadas às camadas inferiores

através da CaService Facility.

A extensão proposta utiliza as duas primeiras camadas da arquitetura de uma estação ITS, Aplicação e Facilities. A camada de Facilities de uma estação ITS possibilita conectar redes internas do veículo através de um gateway (ETSI, 2013). A estrutura de gateways intraveiculares na arquitetura é mostrada na Figura 7.

Para que dados da rede Ethernet pudessem chegar a camada de Facility, um *gateway* foi criado para tornar possível a comunicação entre as aplicações ITS e ECUs que implementam SOME/IP, este *gateway* foi chamado de SomeipGateway. Assim, uma aplicação de uma estação ITS recebe dados do veículo e pode construir mensagens do tipo CAM ou DENM para comunicar-se com outros veículos ou estações fixas.

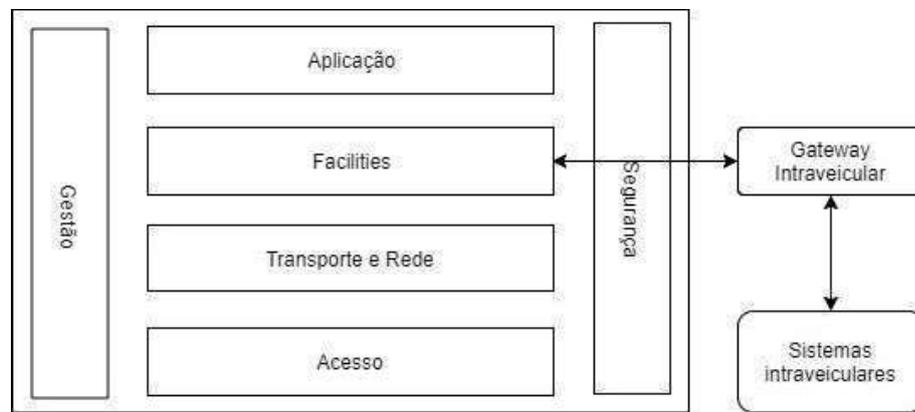


Figura 7 – Gateway para comunicação com sistemas intraveiculares na arquitetura de uma estação ITS (ETSI, 2013)

3.2 Extensão da Arquitetura ITS-S para Comunicação com a Rede Ethernet intraveicular

Para que aplicações ITS possam se comunicar com ECUs que implementam o SOME/IP foi preciso criar uma *Facility*, para que dados oriundos da rede Ethernet cheguem as devidas aplicações requisitantes, e um *gateway*, necessário para traduzir dados da rede Ethernet que utilizam SOME/IP para aplicações ITS. Estes módulos criados foram denominados SomeipFacility e SomeipGateway.

3.2.1 Gateway para intercomunicação entre aplicações ITS e SOME/IP

Seguindo a estrutura de *gateway* intraveicular definida na documentação da ETSI, foi criado um *gateway* que é responsável por prover comunicação entre aplicações ITS e ECUs da rede Ethernet automotiva que usam SOME/IP. O SomeipGateway é responsável por rotear e traduzir as mensagens de requisições feitas pelas aplicações ITS para os nós provedores das informações de interesse.

Requisições que são desempilhadas são enviadas ao SomeipGateway, que encaminha a chamada a ECU que fornece o serviço da informação desejada. Para poder fazer a ponte entre as requisições e a rede automotiva, o SomeipGateway é utilizado como um componente ou nó da rede intraveicular Ethernet. Portanto, como observado na Figura 8, ele implementa o *middleware* do SOME/IP e pode se comunicar com os demais nós da rede através de request/response e publish/subscribe.

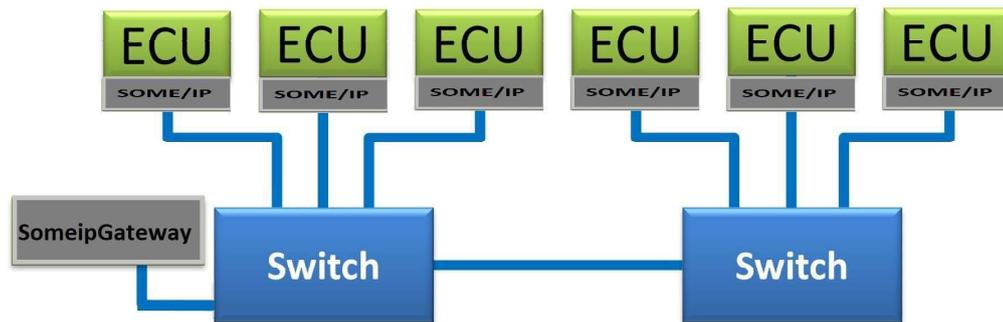


Figura 8 – ECU SomeipGateway atuando como nó da rede Ethernet intraveicular

Como o SomeipFacility é executado no OMNeT++ e o SomeipGateway no Vsomeip, foi preciso criar uma conexão entre estes dois ambientes para que pudessem se comunicar. Para isto, um *socket* foi criado na porta 31123 e todas as requisições feitas pela SomeipFacility, assim como as devidas respostas, puderam trafegar entre os diferentes simuladores, ver Figura 9.

O SomeipGateway, assim como qualquer nó da rede Ethernet que implemente o SOME/IP, possui três métodos principais, seguindo a documentação do Vsomeip (VSOMEIP, 2015).

- *Run*, executa a chamada para requisitar dados de outro nó da rede Ethernet;
- *On message*, é responsável por tratar as mensagens recebidas através da rede;
- *On availability*, este método verifica se os nós fornecedores de serviços na rede estão disponíveis para tratar novas requisições e se estão operando.

Para a execução das simulações foi criado um algoritmo para determinar uma sequência lógica das operações do SomeipGateway, esse algoritmo foi implementado dentro do componente responsável por direcionar e ordenar as ações, esse componente foi chamado de Orquestrador. Quando a SomeipFacility envia uma requisição ao SomeipGateway, o Orquestrador do SomeipGateway recebe a mensagem através do *socket* e abre uma *Thread* para tratar a requisição isoladamente, assim, o Orquestrador consegue operar com quantas requisições forem feitas. Após a abertura da *Thread*, é executado o comando *On*

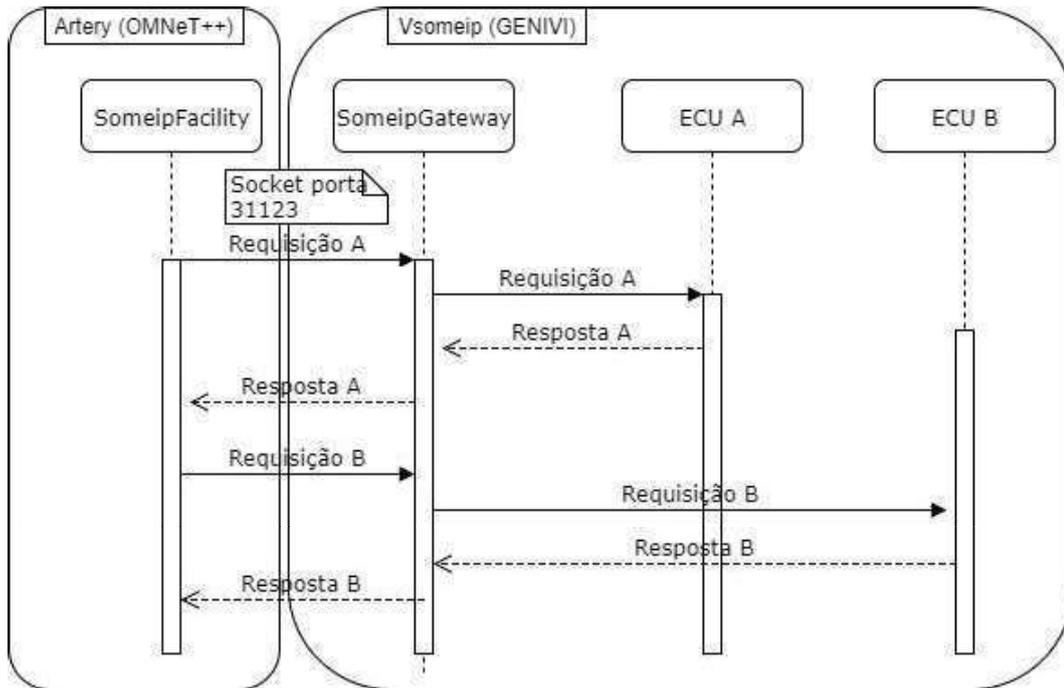


Figura 9 – ECU SomeipGateway atuando como nó da rede Ethernet intraveicular

availability, verificando se o nó de destino está disponível para receber uma requisição e, após isto, executa o método *Run* para fazer uma requisição na rede. Por fim, o Orquestrador executa o método *On message* e espera a resposta chegar. Quando a resposta for recebida da ECU, a mensagem é automaticamente enviada para a SomeipFacility e a *Thread* é fechada.

3.2.2 Facility para comunicação entre aplicações ITS e SOME/IP

Aplicações veiculares que usam a extensão proposta da arquitetura de uma estação ITS fazem requisições a nós da rede Ethernet automotiva através da SomeipFacility. Os nós da rede que se comunicam por meio do SOME/IP podem ser qualquer tipo de hardware que tenha processamento, se comunique através de Ethernet e tenha serviços disponíveis. Os nós da rede Ethernet podem ser ECUs que forneçam dados, como por exemplo, o nível de combustível, quilometragem, ajuste de bancos, dados gerados por ADAS e outras informações que não tenham restrições críticas de tempo.

A SomeipFacility é desenvolvida dentro da camada *Facilities* da arquitetura ITS-S, esta arquitetura é implementada na extensão do OMNeT++ (OMNET, 2001), o Artery (RIEBL et al., 2015). O Artery é uma extensão do Veins (VEINS, 2006), e é usado para simular comunicação interveicular utilizando o padrão ETSI ITS. O SomeipGateway é executado dentro da rede Ethernet automotiva, para simular esta rede foi utilizado o Vsomeip, o *middleware* SOME/IP da GENIVI.

A Figura 10 mostra a aplicação ITS da arquitetura ITS-S no cenário de redes intraveiculares. Nesse cenário, aplicações podem requisitar dados de nós da rede Ethernet

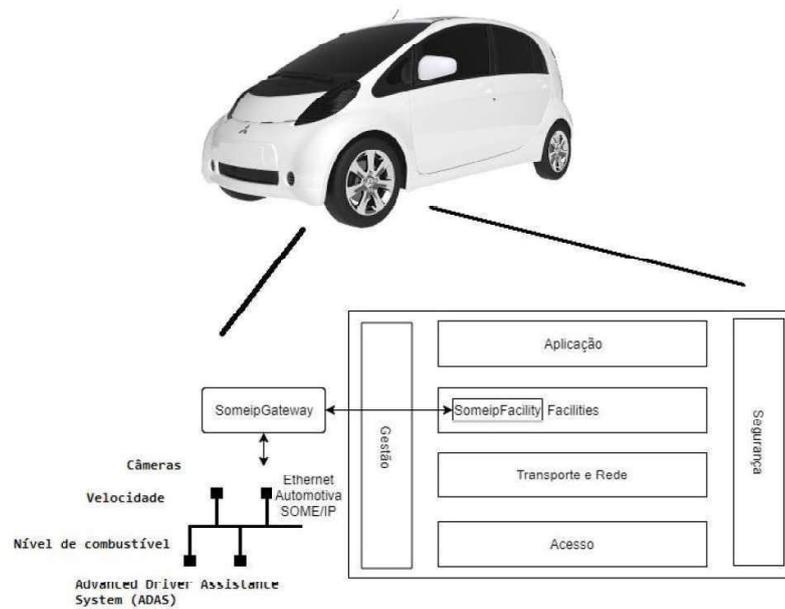


Figura 10 – Extensão da arquitetura aplicada ao cenário de rede intraveicular

automotiva através da SomeipFacility. Esta *Facility* é responsável por fazer a requisição da informação ao nó correto da rede intraveicular e encaminhar a resposta à aplicação que requisitou.

Várias aplicações podem usar a arquitetura para consumir dados da rede Ethernet automotiva. Para isso, requisições podem ser feitas simultaneamente à arquitetura, que as ordena por ordem de chegada e sequência através de uma fila de requisições. Para tratar as diversas requisições, um sistema de filas foi criado na SomeipFacility, em que as novas requisições são inseridas no final da fila e as mais antigas são as primeiras a serem tratadas.

A fila de requisições é compartilhada com dois módulos, como mostrado na Figura 11. O primeiro módulo retira as requisições mais antigas e faz a chamada do serviço para o SomeipGateway. O segundo módulo trata as novas requisições feitas à arquitetura e as insere no final da fila. Como os dois módulos podem requerer acesso à fila ao mesmo tempo, um sistema de contenção foi implementado. Portanto, quando um módulo estiver operando a fila, o outro fica aguardando o sinal de liberação.

Após desempilhar uma requisição, a SomeipFacility usa uma tabela de referência para determinar qual o nó da rede Ethernet automotiva e o serviço que fornece a informação requerida pela aplicação. Desta forma, usando a comunicação orientada a serviços do SOME/IP, não é necessário fazer *broadcasts* na rede, reduzindo o tráfego.

A SomeipFacility possui duas tabelas de referência para tratar requisições feitas pelas aplicações e para encaminhar as respostas enviadas pelo serviços para as aplicações veiculares. As tabelas guardam informações referentes à aplicação que requisitou um dado, sobre a ECU que fornece o serviço e o método da ECU que fornecerá a resposta a requisi-

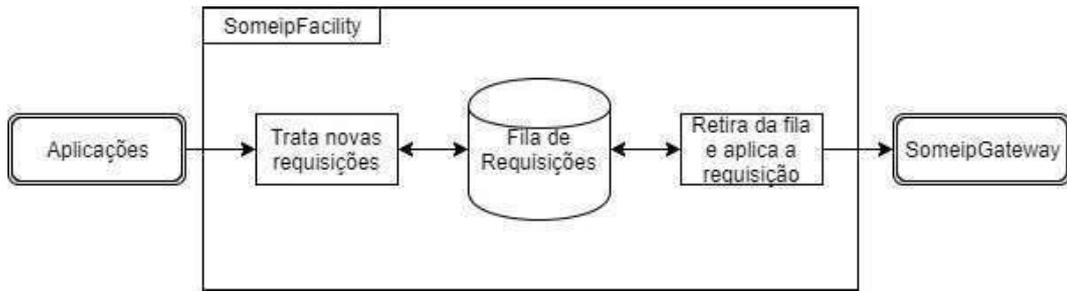


Figura 11 – Fila de requisições da arquitetura

ção. Todas as informações nas tabelas, exceto o Serviço, são códigos hexadecimais. Uma vez que apenas uma ECU fornece um determinado serviço, logo, o ClientID e ServiceID são únicos e não podem ser repetidos. A Tabela 2 é utilizada para identificar qual é o clientID, o atributo para determinar a ECU que fornece a informação, e o serviceID, que é o método chamado para receber o resultado referente à informação que foi requisitada pela aplicação. A SomeipFacility guarda o ApplicationID da aplicação que fez a requisição, o ClientID e o ServiceID. Essas informações serão usadas quando o resultado da requisição chegar à Facility.

Tabela 2 – Tabela para referenciar informações às ECUs que fornecem os resultados

Serviço	ClientID	ServiceID
Nível de combustível	0x10	0x11
Quilometragem	0x12	0x13
Regulagem de assento	0x14	0x15

Com o ClientID e o ServiceID, a SomeipFacility faz a requisição ao SomeipGateway. A Facility informa ao SomeipGateway o ClientID e o ServiceID, este faz a requisição diretamente à ECU que fornece a informação 12.

No caminho inverso, o SomeipGateway recebe o resultado da requisição do serviço e envia a informação à SomeipFacility. Esta Facility faz uso da Tabela 3, que é utilizada para saber qual aplicação fez determinada requisição. A SomeipFacility recebe o resultado e analisa qual foi o ClientID e o ServiceID dos dados vindos do gateway. Então, através da tabela, e consultando o ApplicationID, é possível saber qual aplicação fez a requisição dos dados. Por fim, a Facility encaminha a informação à aplicação requerente.

Tabela 3 – Tabela para referenciar aplicações que fizeram as requisições

ClientID	ServiceID	ApplicationID
0x10	0x11	1
0x12	0x13	2
0x14	0x15	1

Na Figura 12 é mostrado um diagrama de sequência de toda a operação relacionada a comunicação entre aplicações ITS e ECUs que implementam o SOME/IP. A Aplicação ITS com ID 1 faz uma requisição a SomeipFacility para o Serviço de Nível de Combustível. Segundo a Tabela 2, o serviço Nível de Combustível refere-se ao ClientID com código hexadecimal 0x10 e cujo o serviço específico para esta requisição é o 0x11. A requisição entra na Fila de Requisições e é enviada ao SomeipGateway, com o código da ECU (ClientID) e o código do serviço (ServiceID), o SomeipGateway consegue fazer a requisição à ECU e espera a resposta. A resposta é enviada novamente a SomeipFacility, esta compara o ClientID e o ServiceID da resposta, no caso, 0x10 e 0x11, e, através da Tabela 3, decide qual a aplicação fez a requisição. Por fim, encaminha a resposta para o requerente.

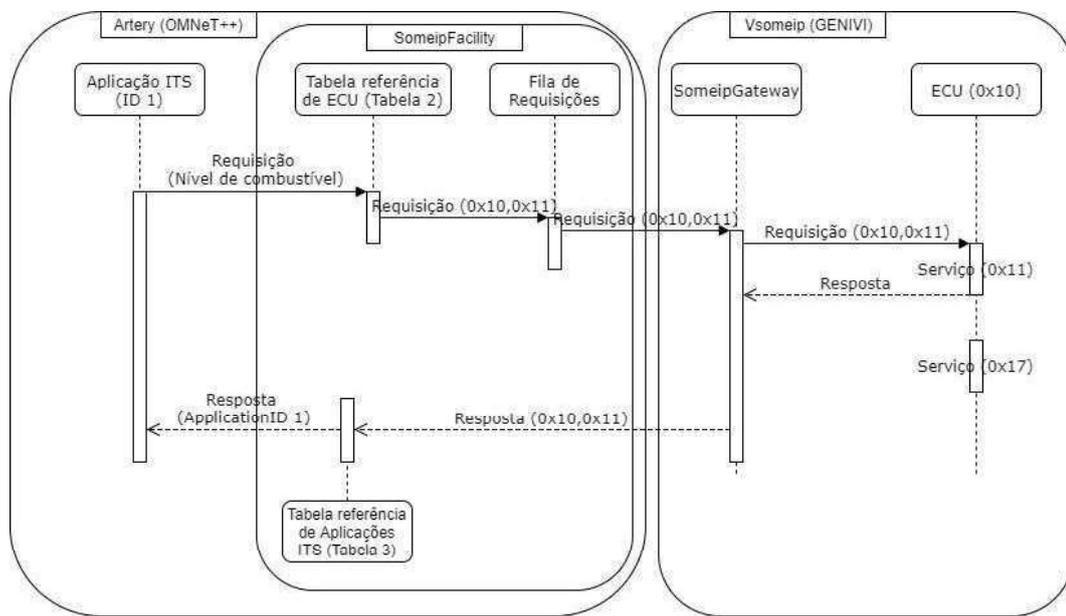


Figura 12 – Diagrama de sequência da comunicação entre aplicações ITS e ECUs utilizando SOME/IP

3.3 Extensão da Arquitetura ITS-S para Comunicação com Dispositivos IoT

Dispositivos pessoais podem ser considerados estações ITS, logo, podem fazer uso da arquitetura de comunicação da estação. Entretanto, dispositivos IoT não fazem parte deste grupo e, geralmente, dispõem da pilha TCP/IP para se comunicar. Para não alterar o protocolo de comunicação de dispositivos de IoT, foi criada uma Facility para tornar esta intercomunicação possível.

A extensão da arquitetura também promove a comunicação entre aplicações ITS e dispositivos IoT, como mostrado na Figura 13. A comunicação entre os dispositivos IoT e as aplicações ITS é feita utilizando IPv6. Os IPs dos dispositivos conectados ao veículo através da extensão da arquitetura da estação ITS são disponibilizados às aplicações por uma tabela criada na ContikiFacility.

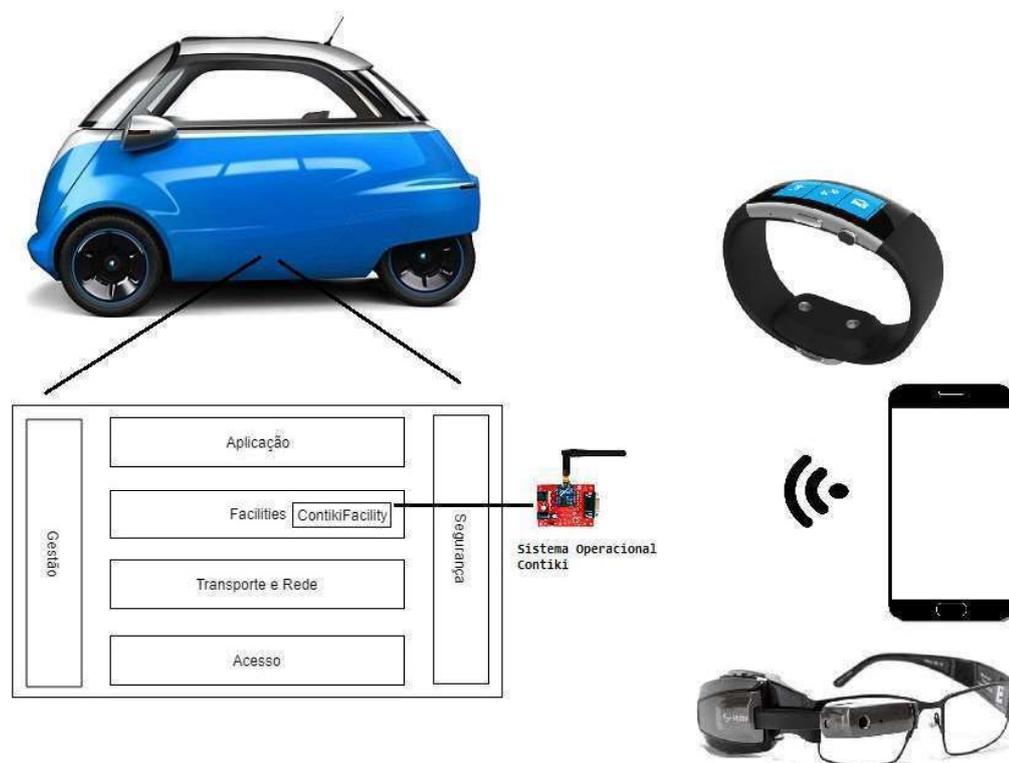


Figura 13 – Extensão da arquitetura aplicada ao cenário de Internet das Coisas

O sistema operacional Contiki para IoT possui compatibilidade com vários hardwares e protocolos de comunicação que usam IP. Por este motivo foi integrado à arquitetura, desta forma, aumentando o escopo de dispositivos compatíveis com a arquitetura.

Para a integração entre a arquitetura e o Contiki, é necessário ter uma placa que seja compatível com este sistema operacional de IoT (Seção 2.1.1.1 do capítulo 2). Para não obter as placas para os testes, uma opção usada para não utilização de placas físicas no projeto foi a simulação da comunicação entre aplicações ITS e dispositivos através de IPv6 usando os exemplos de utilização da pilha de comunicação do Contiki. Foi utilizado o no projeto a simulação de placas que implementam o exemplo com o protocolo rpl-upd (UDP, 2010). As placas simuladas implementam o código do udp-server, enquanto as requisições são feitas através da ContikiFacility executando o código do udp-client.

Os principais métodos utilizados para a execução das simulações foram:

- *tcpip handler*, método responsável por tratar as respostas;
- *send packet*, método utilizado para envio de mensagens;
- *PROCESS THREAD*, orquestrador do algoritmo, decide o IPv6 utilizado, abre a conexão e inicia a comunicação.

Para a abertura de conexão entre as aplicações e dispositivos foi criada uma tabela que contém informações sobre os dispositivos que podem se comunicar através da arquitetura.

Tabela 4 – Tabela para conexões com dispositivos IoT

Dispositivo	IPv6	Porta
Dispositivo 1	2001:db8:a0b:12f0::1	1234
Dispositivo 2	2001:cdba::3257:9652	4321

Na Tabela 4, são disponibilizadas três informações que serão usadas para uma aplicação se comunicar com um dispositivo. A primeira é a identificação do dispositivo, que pode ser o nome ou alguma descrição, a segunda é o IPv6 e, por fim, a porta.

Como mostrado na Figura 14, uma aplicação ITS pode fazer uma requisição de comunicação com dispositivos IoT usando a extensão proposta, para isto, a aplicação consulta o dispositivo na Tabela de endereços. Após isso, a ContikiFacility relaciona o IPv6 e a porta do dispositivo IoT, no caso da simulação, placas Sky Motes. Por fim, a conexão é reservada entre a aplicação ITS e o dispositivo, todo o tráfego após iniciada a conexão é feita sem o auxílio da ContikiFacility.

Para facilitar as simulações, não foi implementado a descoberta dinâmica de IP, todos os IPs são colocados estaticamente na Tabela 4.

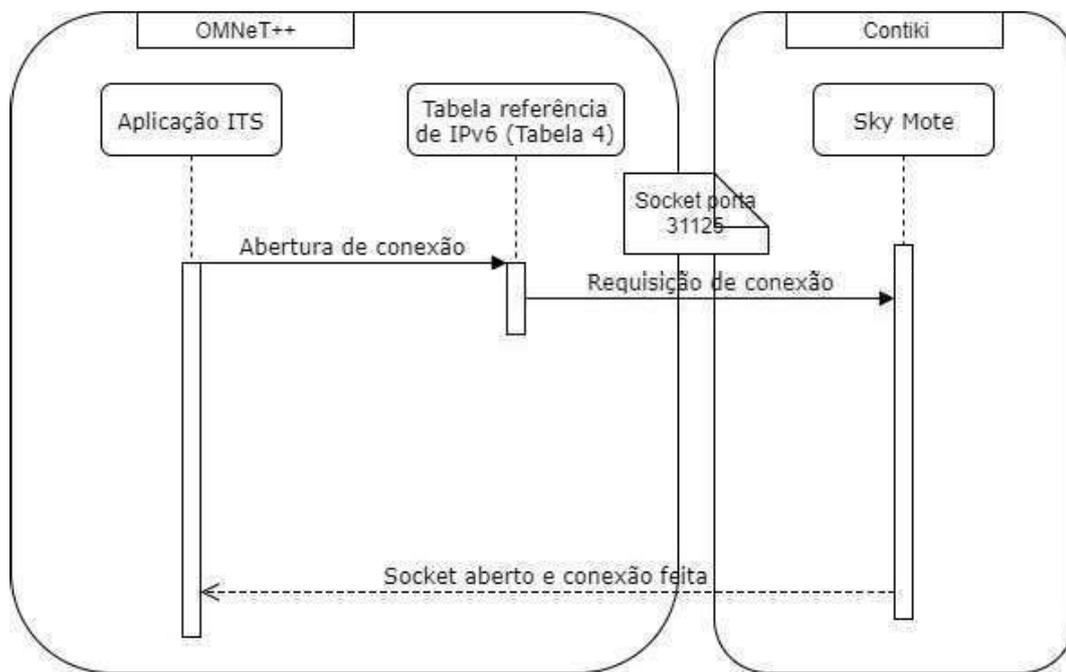


Figura 14 – Diagrama de sequência da comunicação entre aplicações ITS e dispositivos IoT

3.4 Considerações Finais

Este capítulo apresentou a arquitetura da extensão proposta nesse projeto. Para um melhor entendimento, a extensão foi dividida em dois cenários. O primeiro foi relacionado à comunicação entre aplicações ITS e ECUs que implementam o SOME/IP. O segundo foi

a comunicação entre aplicações ITS e dispositivos IoT que implementam a pilha do S.O Contiki.

No próximo capítulo, é explicada a metodologia utilizada para criar e validar a extensão proposta da arquitetura de estações ITS para conectar dispositivos IoT e nós das redes intraveiculares.

4 Metodologia de Avaliação

Este capítulo descreve a metodologia adotada para avaliar o pressuposto de que aplicações que usam a extensão proposta da arquitetura de estações ITS satisfazem a restrição de atraso fim-a-fim máximo permitido em cada domínio de aplicação. Será explicado como foi o processo de obtenção dos dados definindo o seu tipo, onde foram gerados e como foram adquiridos. Será explicado também como foram adquiridos os valores para definir o atraso fim-a-fim nos domínios de rede intraveicular e IoT. Serão apresentadas as métricas utilizadas para a análise da arquitetura e as ferramentas utilizadas para o desenvolvimento e simulação da arquitetura. Por fim, será descrita a simulação.

4.1 *Obtenção dos dados*

Arquiteturas definem arranjos de componentes, propriedades e relacionamentos entre elementos próprios e externos. Geralmente, arquiteturas de comunicação resultam na criação de abstrações para modularizar e facilitar a manutenção. Contudo, adicionar abstrações pode causar atrasos na comunicação.

No contexto de redes intraveiculares, cada domínio de aplicação possui uma restrição de atraso máximo permitido. Violar essas restrições pode causar desde mau funcionamento em equipamentos a acidentes. O mesmo acontece com qualquer aplicação que possui restrição de tempo, independente do ambiente empregado.

O principal objetivo da arquitetura é prover integração com relação a domínios diferentes e, através disso, fornecer dados de diferentes fontes a aplicações veiculares. O propósito a ser alcançado por essa avaliação é estimar o impacto de usar a extensão proposta da arquitetura ITS-S para a comunicação nos domínios de rede intraveicular e Internet das Coisas.

Para garantir que a extensão proposta da arquitetura ITS-S respeite as restrições de tempo, é necessário determinar o atraso fim-a-fim médio da comunicação usando a arquitetura. Essa informação é adquirida calculando várias vezes o tempo gasto desde um dado gerado por um componente da rede Ethernet automotiva ou um dispositivo de IoT, até essa informação chegar na aplicação interessada que utiliza a arquitetura.

Os valores dos atrasos fim-a-fim nos cenários de aplicação foram coletados a partir de simulações do uso da extensão proposta da arquitetura ITS-S. A arquitetura, apresentada no capítulo 3, foi simulada no OMNeT++, utilizando a extensão para o padrão ETSI, o Artery, e com codificada em C++.

Para avaliar os dados obtidos das simulações, foi preciso determinar qual a distribuição que as amostras de atrasos fim-a-fim seguiam nos contextos de redes intraveiculares, Internet das Coisas e no contexto integrado. Portanto, foram aplicados testes de normalidade para determinar se as amostras seguiam a distribuição normal. Para cada amostra da população nos contextos de aplicação da arquitetura, foi aplicado o teste de normalidade de Kolmogorov-Smirnov.

Com a finalidade de comparar as médias encontradas nas amostras e o valor de atraso máximo permitido, sabendo a distribuição que as amostras seguiam, foi aplicado o teste não paramétrico de Wilcoxon para amostras que não seguem a distribuição normal. Assim, foi possível calcular se a extensão cumpria os requisitos de atraso especificados.

Para caracterizar o contexto de aplicação da arquitetura para o qual a metodologia será proposta, é importante analisar os três cenários de aplicação. Os cenários de IoT e redes intraveiculares, a princípio, são analisados separadamente, pois podem ser utilizados independentemente através da extensão proposta. Contudo, com a finalidade de analisar a integração desses módulos à arquitetura, o último cenário é a utilização simultânea dos mesmos.

4.2 Cenários

Com o objetivo de obter o tempo médio do atraso fim-a-fim nos diferentes cenários do escopo da arquitetura, os domínios de atuação da arquitetura foram simulados inicialmente separadamente para obter resultados específicos. Por fim, todos os domínios foram simulados simultaneamente. O cenário da arquitetura será apresentado por domínio de aplicação, logo, foi dividido nos contextos de IoT e Ethernet automotiva.

Os cenários de simulação foram construídos sobre o S.O Linux Mint (MINT, 2006), contudo, os *frameworks* utilizados Vsomeip e Contiki são multiplataformas, podendo rodar em Windows também.

4.2.1 Redes Intraveiculares

Em uma rede intraveicular, são definidos domínios de aplicação. Cada domínio possui uma restrição de atraso fim-a-fim. Portanto, uma aplicação de entretenimento, por exemplo, a visualização de um vídeo da câmera de ré em um *display*, possui uma restrição de atraso fim-a-fim menor que unidades de sistemas críticos. Os atrasos máximos por domínio são especificados na Tabela 5.

O escopo de aplicação do SOME/IP é para comunicação de dados não críticos e se encaixa nos domínios de informação, entretenimento e dados de segurança. São exemplos de aplicações no domínio de dados de segurança os sensores de estacionamento, detectores de pedestres e controle de condução adaptativa. Aplicações que utilizarem a arquitetura

Tabela 5 – Tabela de atrasos máximos fim-a-fim por domínio em redes intraveiculares (TUOHY et al., 2015)

Classe de tráfego	Atraso fim-a-fim máximo	Tempo de serviço
Dados de controle	2,5 ms (STEFFEN et al., 2008)	10 - 100 ms
Dados de segurança	45 ms (RAHMANI et al., 2008)	0,05 - 1 ms
Informação/ Entretenimento	150 ms (WOLF; GRIWODZ; STEINMETZ, 1997)	1 ms

poderão requisitar dados de diversas finalidades, como por exemplo, velocidade, sistema avançado de assistência ao condutor, nível de combustível, entre outros.

O caminho que os dados percorrem na arquitetura segue a Figura 15. A comunicação na rede Ethernet intraveicular utilizando o SOME/IP é orientada a serviços, então uma aplicação precisa fazer a requisição de uma informação e a arquitetura vai buscar a ECU que fornece o serviço e entregar à aplicação de maneira transparente.

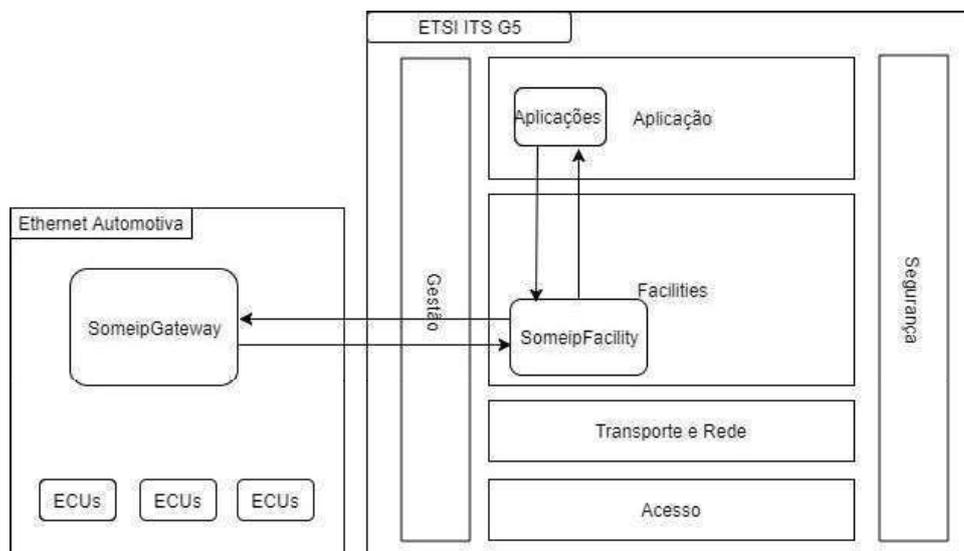


Figura 15 – Caminho dos dados no cenário de Redes Intraveiculares

Durante a simulação deste cenário, o caminho de dados é iniciado pelas requisições feitas pelas aplicações ITS, após isto, a extensão da arquitetura enfileira as requisições e as organiza em ordem de chegada. As primeiras a chegar serão as primeiras a serem tratadas. Quanto maior a quantidade de requisições, maiores serão as filas e aumentando, assim, o atraso fim-a-fim da comunicação. As requisições que são desenfileiradas são recebidas pelo SomeipGateway, este fará a requisição por serviço a ECU responsável pela informação. A resposta gerada pela ECU é recebida pelo SomeipGateway, este enviará a resposta para a SomeipFacility que encaminhará a resposta a aplicação requisitante. Ao contrário da requisição, a resposta não passa por sistemas de filas.

4.2.2 Internet das Coisas

O cenário de Internet das Coisas possui menos restrições de tempo com relação ao de redes intraveiculares. Vários dispositivos que sejam compatíveis com o Contiki podem ser utilizados, como por exemplo, relógios e óculos inteligentes. A arquitetura utiliza o sistema operacional Contiki e pode utilizar hardwares integrados aos veículos que possuam interfaces de comunicação para trocar dados com dispositivos IoT.

A extensão proposta da arquitetura disponibiliza os IPs dos dispositivos que podem se comunicar com as aplicações veiculares através da ContikiFacility. A Figura 16 mostra o caminho dos dados no contexto de IoT. A comunicação começa quando uma aplicação faz uma requisição para se comunicar com um dispositivo IoT. A requisição é feita à ContikiFacility, que escolhe uma porta disponível para abrir a conexão. Após isso, a ContikiFacility aguarda o sinal da aplicação para fechar a conexão.

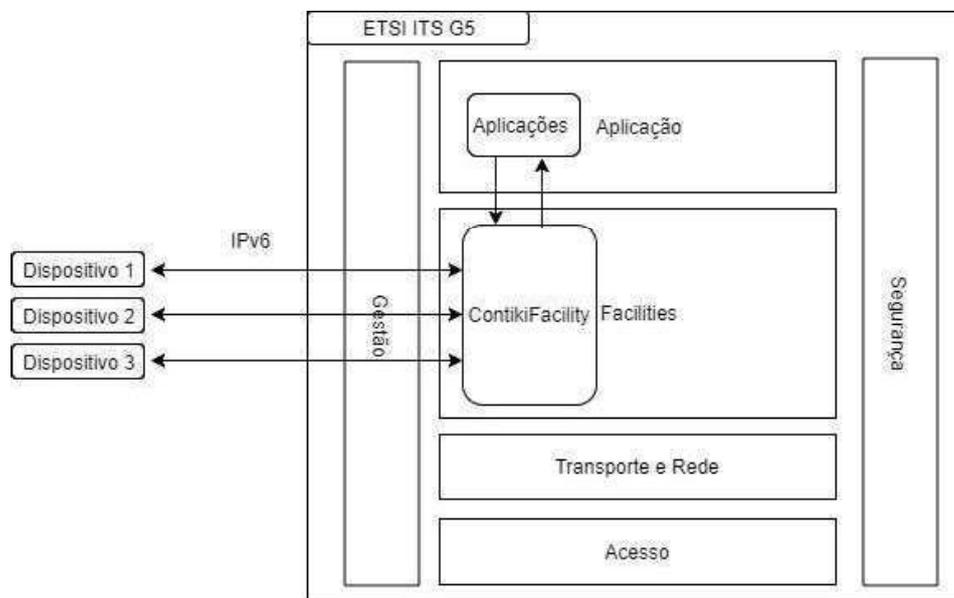


Figura 16 – Caminho dos dados no cenário de Internet das Coisas

A partir de uma requisição feita por uma aplicação veicular para se comunicar com um dispositivo IoT, o atraso fim-a-fim é calculado através da subtração do instante final e o instante inicial. O instante inicial é adquirido no momento em que a requisição é feita pela aplicação para abrir uma conexão. A ContikiFacility abre a conexão e a aplicação envia uma requisição para o dispositivo. O dispositivo recebe a requisição e transmite a resposta, o instante final é aquele em que a aplicação recebe a resposta.

4.2.3 Cenário Integrado

O objetivo da arquitetura é prover integração nos domínios de Internet das Coisas e rede intraveicular, criando intercomunicação entre estes domínios e aplicações veiculares. Para

cumprir os objetivos desta pesquisa, este terceiro cenário será analisado com o propósito de integrar os dois domínios explorados anteriormente de forma simultânea.

A abordagem do cenário integrado vai efetivar a arquitetura no seu propósito, ou seja, aplicações veiculares consumindo e trocando dados em um ambiente de comunicação híbrida, abstraindo a complexidade de vários protocolos de comunicação. Os caminhos percorridos pelos dados são os mesmos dos outros cenários apresentados nas Figuras (15 e 16), contudo, uma vez que os módulos são independentes, as comunicações são feitas paralelamente. Uma aplicação poderá solicitar abertura de conexão com um dispositivo IoT, enquanto outra requisita dados de ECUs na rede Ethernet intraveicular.

Seguindo os outros cenários, a avaliação da arquitetura será feita separadamente por domínios de aplicação. Os atrasos fim-a-fim serão calculados e, para o contexto de rede intraveicular, o resultado será comparado com o atraso máximo permitido apresentado na Tabela 5.

4.3 Métricas

Para análise dos atrasos fim-a-fim das aplicações que usam a extensão proposta da arquitetura de estações ITS para se comunicar com dispositivos de IoT e receber informações de dados do veículo, foram definidas métricas a serem utilizadas.

- Atraso fim-a-fim: intervalo entre uma mensagem sendo enviada por uma ECU ou de um dispositivo de IoT, até chegar na aplicação. A partir dos valores gerados será calculado o atraso fim-a-fim médio de cada domínio.

4.4 Ferramentas

Para a realização dos experimentos foram utilizadas ferramentas com finalidades distintas, para tanto, pode ser observado na Tabela 6.

Tabela 6 – Ferramentas usadas para o desenvolvimento do projeto

Nome	Funcionalidade
Artery	Simulador de redes baseado em OMNeT++ para comunicação v2v ou v2i, é uma extensão do Veins, contudo, utiliza o padrão ETSI ITS G5
Contiki	Sistema Operacional de Internet das Coisas, implementa uma pilha de comunicação otimizada para suportar dispositivos com protocolos diferentes no universo de IoT
vsomeip	Implementação do <i>middleware</i> intraveicular SOME/IP, desenvolvido pela GENIVI, cria um ambiente de comunicação baseada em serviço na rede Ethernet automotiva
C++	Linguagem de programação utilizada para o desenvolvimento da arquitetura
R	Linguagem de programação utilizada para a análise dos resultados. Execução dos testes de normalidade, testes de hipóteses e geração das estatísticas descritivas dos resultados.

4.5 Simulações

Os experimentos foram divididos em três etapas. Em cada etapa, a extensão da arquitetura foi avaliada com base no domínio de aplicação. Por fim, a arquitetura foi avaliada por completo, ou seja, com os contextos de IoT e rede intraveicular operando simultaneamente. O objetivo de cada etapa é avaliar o atraso fim-a-fim de requisição e resposta de aplicações que utilizam a extensão proposta da arquitetura em cada cenário de aplicação.

4.5.1 Simulações no contexto de redes intraveiculares

Em um cenário de Carros Conectados, um automóvel pode requisitar informações sobre outro carro, assim como receber uma requisição de uma aplicação na nuvem ou localmente. Aplicações ITS compreendem aplicações locais veiculares convencionais, por exemplo, tela de vídeo de câmera de ré ou painel de informações, assim como, aplicações WEB que usarão aplicações clientes locais e através da extensão proposta da arquitetura de ITS-S, consumir dados da rede intraveicular Ethernet.

Devido as indústrias automotivas não divulgarem dados sobre o comportamento da rede intraveicular, não foi possível definir a distribuição apropriada no cenário de requisições de aplicações veiculares para uma rede intraveicular Ethernet. A aquisição destas informações se tornam ainda mais complicadas de se conseguir se for inserido o SOME/IP no contexto. Portanto, como aplicações ITS podem ser aplicações WEB ou receber requisições através das mesmas, foi simulado um cenário de aplicações ITS com um comportamento de aplicações de serviço Web, desta forma foi utilizado a distribuição exponencial que melhor representa aplicações WEB (LEE; GUPTA, 2007). Já para aplicações exclusi-

vamente locais que, geralmente, necessitam de sincronismo e periodismo com os sensores existentes no veículo foi definido o experimento com tempo fixo entre as requisições. Portanto, os experimentos no contexto de redes intraveiculares seguiram duas abordagens. A primeira com requisições definidas em espaços de tempos definidos e, na segunda, foi usada uma distribuição exponencial entre tempos de chegada de requisições.

No experimento da arquitetura no contexto de redes veiculares, foi analisado o atraso fim-a-fim no cenário de aplicações veiculares recebendo dados de nós da rede Ethernet automotiva. Foram definidos 5 nós na rede Ethernet que implementam o *middleware* SOME/IP para comunicação baseada em serviços. Cada nó ou ECU possui um ClientID e um ServiceID, que irão determinar quem e qual método que retorna o resultado esperado. O método de comunicação utilizado no experimento foi o request/response, visto que simplificaria o cálculo do tempo de atraso fim-a-fim quando estão envolvidos apenas um emissor e um receptor na comunicação.

Foram definidas também 5 aplicações que faziam requisições às ECUs. Cada aplicação podia solicitar informações de quaisquer dos 5 nós da rede Ethernet. A escolha de qual aplicação iria fazer a requisição era feita aleatoriamente.

Cada experimento no contexto de rede intraveicular consistiu em fazer 3000 requisições, ou seja, uma amostra de valores de atrasos fim-a-fim de tamanho 3000. Não foram incluídos nos atrasos os tempos de inicialização e configuração dos nós da rede Ethernet automotiva para o uso do SOME/IP, pois não influenciaria nos resultados durante a requisição e resposta.

Para a validação dos resultados no cenário da extensão proposta da arquitetura com rede intraveicular, testes de hipóteses apropriados para a distribuição da amostra foram utilizados para comparar a média do atraso fim-a-fim com o atraso máximo permitido para cada domínio de aplicação mostrado na Tabela 5.

4.5.1.1 Requisições com tempos definidos

Neste cenário, as requisições foram feitas com uma taxa de uma requisição a cada 100 ms. O objetivo foi validar a interconexão entre as aplicações veiculares e a rede intraveicular utilizando o *middleware* SOME/IP.

4.5.1.2 Requisições por distribuição exponencial

Foram geradas requisições seguindo a distribuição exponencial. Esta distribuição define o intervalo entre as partidas das requisições, onde o parâmetro λ foi variado entre 1, 5 e 10. Tais valores têm o propósito de representar a quantidade de fontes de dados que uma aplicação pode vir a consumir para gerar informação ao usuário. Por exemplo, um aplicativo de navegação pode vir a requisitar dados oriundos de sensores como GPS, temperatura, velocidade, etc, além de dispositivos de IoT como *smart band*, *smart watch*,

dentre outros. O conjunto desses dados podem vir a fornecer informação precisa sobre navegação levando em consideração a condição física do condutor.

4.5.2 Simulações no contexto de Internet das Coisas

Neste experimento, foram definidas 5 aplicações que fariam requisições a 5 dispositivos de Internet das Coisas. A escolha do dispositivo por parte da aplicação foi feita de maneira aleatória. Foram geradas requisições a cada 100 ms, totalizando 3000 requisições e uma amostra de 3000 valores de atrasos fim-a-fim.

O pool de conexões foi configurado para ter 5 portas disponíveis para serem usadas por aplicações. Como a avaliação da comunicação é feita enviando um pacote de dados e esperando a resposta, a simultaneidade da utilização das portas é percebida quando há um aumento significativo na quantidade de requisições por tempo.

4.5.3 Simulações no contexto integrado da arquitetura

Este experimento tem como objetivo avaliar o funcionamento da extensão da arquitetura nos contextos de IoT e rede intraveicular. Portanto, neste experimento foram executados simultaneamente os cenários anteriores de requisições com tempos definidos.

Foram utilizadas 5 aplicações, 5 nós da rede Ethernet automotiva e 5 dispositivos de IoT. Todas as requisições são feitas de maneira aleatória com relação ao destinatário. As chamadas por requisições aconteceram uma vez a cada 100 ms. No contexto de IoT, foram disponibilizadas 5 portas no pool de conexões.

Com o valor do atraso fim-a-fim gerado no contexto de rede intraveicular, aplicamos um teste de hipótese apropriado para a distribuição da amostra, com o objetivo de comparar a média dos atrasos adquiridos com o valor máximo dos atraso fim-a-fim nos domínios que o SOME/IP pode ser aplicado. Nesse caso, os atrasos máximos são de 45 ms e 150 ms, mostrados na Tabela 5 .

4.6 Considerações Finais

Este capítulo apresentou a metodologia utilizada para avaliação dos resultados obtidos nas simulações da extensão proposta da arquitetura ITS-S. Inicialmente foi apresentado como foram obtidos os dados através de simulações no OMNeT++. Foi mostrado que a avaliação da extensão foi segmentada em cenários, esta separação foi feita para poder avaliar a extensão através de comunicação entre aplicações ITS, IoT e rede Ethernet intraveicular. Por fim, foram avaliados ambos cenários juntos com o objetivo de testar a integração da extensão.

Diante dos resultados obtidos das simulações e o atraso fim-a-fim máximo permitido mostrado por (TUOHY et al., 2015), foi apresentada a utilização do teste não paramétrico de Wilcoxon para comparar a média dos resultados com o atraso máximo definido.

As ferramentas utilizadas foram descritas na seção 4.4 e a utilização de cada uma foi apresentada no capítulo 3.

No próximo capítulo, são apresentados os resultados obtidos através das simulações e discussões sobre eles.

5 Resultados

Os resultados a seguir são provenientes dos experimentos apresentados no capítulo anterior. A partir dos limiares de atraso máximo permitidos em cada domínio de aplicação no cenário de Carros Conectados, foi analisado o impacto da utilização da extensão proposta da arquitetura de estações ITS do padrão ETSI nos diferentes cenários propostos. Resultados obtidos serão apresentados nesse capítulo separados por domínios e explicados através de gráficos e tabelas.

5.1 *Cenário de Rede Intraveicular*

No cenário de rede intraveicular, o atraso fim-a-fim máximo de 45 ms foi definido para aplicações no domínio de segurança e 140 ms no domínio de entretenimento (TUOHY et al., 2015). O objetivo do experimento é avaliar se a extensão proposta da arquitetura neste contexto respeita as restrições de atraso definidas.

No cenário de comunicações entre aplicações veiculares e rede Ethernet intraveicular utilizando a arquitetura, o caminho seguido pelos dados é mostrado na Figura 15. Como uma aplicação precisa requisitar uma informação e receber o retorno, o atraso é medido somando o tempo gasto no caminho de ida e de volta.

5.1.1 Requisições com tempos definidos

Este experimento foi executado isolando o cenário de comunicação entre aplicações ITS e nós da rede intraveicular. O objetivo deste experimento foi analisar se a extensão proposta da arquitetura obedecia aos requisitos de atraso fim-a-fim máximos especificados. Para comprovar esta hipótese, pode-se observar através da Tabela 7 que a média dos atrasos para o ambiente simulado foi de 0,954561 ms.

Foram calculados 3000 valores de atrasos fim-a-fim. O valor máximo observado foi de 1,299968 ms e o valor mínimo foi de 0,701801 ms. O Intervalo de Confiança (IC) encontrado para o nível de confiança de 95% foi de 0,002668. Este resultado mostra que aproximadamente 95% da amostra esteja contida no intervalo [0,951893, 0,957229].

Tabela 7 – Estatística descritiva para o experimento 5.1.2

Tamanho amostral	Mínimo	Máximo	Média	Desvio Padrão	IC
3000	0,701801 ms	1,299968 ms	0,954561 ms	0,074538	0,002668

Para verificar a hipótese da amostra coletada seguir a distribuição normal, foi utilizado o teste de normalidade de Kolmogorov-Smirnov. No teste foi definida a hipótese nula (H_0) de que a amostra seguia a distribuição normal e a hipótese alternativa (H_1), na qual a amostra não segue a distribuição normal.

O resultado do teste é representado pelo valor de P (p-value). O p-value é uma probabilidade que mede a evidência contra a hipótese nula. Um p-value menor fornece uma evidência mais forte contra a hipótese nula. Para determinar se os dados não seguem uma distribuição normal, basta comparar o p-value com o nível de significância determinado. Foi determinado um nível de significância de 0,05 ou nível de confiança de 95%, estes valores indicam um risco de 5% de concluir que os dados não seguem a distribuição normal quando eles realmente a seguem.

O valor do p-value obtido no teste de Kolmogorov-Smirnov foi igual a $2 \cdot 2^{-16}$. Portanto, como o valor do p-value foi menor que o nível de significância determinado de 0,05, então foi rejeitado H_0 , logo, é rejeitado a hipótese nula de que a amostra segue a distribuição normal.

H_0 : Os dados seguem uma distribuição normal

H_1 : Os dados não seguem uma distribuição normal

Como foi verificado que a distribuição da amostra nesse contexto não segue a normal com nível de confiança de 95%, para verificar se a média dessa amostra é menor que o valor máximo de atraso fim-a-fim definido, então é necessário comparar a média usando um teste não paramétrico. O teste não paramétrico usado foi o Wilcoxon.

Para realizar o teste de Wilcoxon, primeiramente, foi estabelecido $H_0 : \mu = \mu_0$, esta expressão apresenta a hipótese da média amostral μ ser igual ao valor máximo definido do atraso fim-a-fim μ_0 . Na hipótese alternativa $H_1 : \mu \leq \mu_0$, aquela que acredita-se que pode ser ou espera provar ser verdadeira, foi definido que μ era menor que μ_0 , ou seja, espero provar que a média encontrada do atraso fim-a-fim seja menor que o valor máximo permitido.

$$H_0 : \mu = \mu_0$$

$$H_1 : \mu \leq \mu_0$$

Os parâmetros que foram definidos nesse teste de hipótese foram: $\mu_0 = 45$ ms e a hipótese alternativa H_1 de que a média amostral μ seja menor que μ_0 .

- μ_0 : 45 ms
- H_1 : rejeita-se a hipótese nula e não rejeita a hipótese da média da amostra ser menor que o atraso máximo fim-a-fim permitido de 45 ms.

$$H_0 : \mu = 45ms$$

$$H_1 : \mu \leq 45ms$$

O resultado do teste de Wilcoxon foi o p-value = $2,2^{-16}$, este valor mostra a rejeição de H_0 e confirma a não rejeição à H_1 , isto significa que a média dos dados da amostra no contexto de Rede Intraveicular no cenário isolado com requisições com tempos definidos é menor que a restrição máxima do atraso fim-a-fim de 45 ms.

5.1.2 Requisições com distribuição exponencial

Para a geração de carga de trabalho ou requisições, foram utilizadas aplicações veiculares que poderiam vir a ser acessadas pela internet. Desta forma, foram geradas requisições seguindo a distribuição exponencial. Esta distribuição define o intervalo entre as partidas das requisições, onde o parâmetro λ foi variado entre 1, 5 e 10. Tais valores tem o propósito de representar a quantidade de fontes de dados que uma aplicação pode vir a consumir para gerar informação ao usuário. Por exemplo, um aplicativo de navegação pode vir a requisitar dados oriundos de sensores como GPS, temperatura, velocidade, etc, além de dispositivos de IoT como smart band, smart watch, dentre outros. O conjunto desses dados podem vir a fornecer informação precisa sobre navegação levando em consideração a condição física do condutor.

Este experimento isolado tem o mesmo objetivo do cenário anterior, verificar se a média dos atrasos fim-a-fim da amostra são menores que o atraso máximo definido de 45 ms. A diferença desse experimento foi a utilização da distribuição exponencial para determinar os intervalos entre as requisições. Os valores de λ definidos foram 1, 5 e 10.

O objetivo desse experimento é analisar se o pressuposto de que a arquitetura respeita o requisito de atraso máximo fim-a-fim, mesmo em situações de grandes quantidades de requisições.

Tabela 8 – Estatística descritiva para o experimento 5.1.3

λ	Tamanho	Mínimo	Máximo	Média	Desvio Padrão	IC
1	3000	0,701202 ms	1,129984 ms	0,95437 ms	0,078704	0,002817
5	3000	0,708201 ms	3,689984 ms	1,180144 ms	0,463923	0,016608
10	3000	0,701102 ms	6,19008 ms	1,264142 ms	0,640006	0,022915

Como apresentando na tabela 10, inicialmente foi escolhido o λ igual a 1, devido ao grande intervalo de tempo entre as requisições, não foi observado atraso fim-a-fim maior que 1,5 ms, o mesmo comportamento foi apresentado no experimento anterior, cujo intervalo entre as requisições era de 100 ms.

Como a extensão proposta da arquitetura utiliza fila de requisições, é esperado que quando houver uma diminuição significativa no intervalo entre as requisições ocorra um congestionamento na fila. Para λ igual a 5 e 10, os intervalos entre as requisições variam entre poucos milissegundos.

Aumentando o λ para 5, houve um aumento evidente na média e em todos os outros fatores, esse crescimento se reflete com $\lambda = 10$, concluímos dessa análise que, à medida que as requisições aumentam, as filas congestionam e os atrasos fim-a-fim aumentam.

Aumentando o λ para 5, houve um aumento evidente no valor máximo obtido e um aumento sutil na média, esse crescimento se reflete com $\lambda = 10$, concluímos dessa análise que, à medida que as requisições aumentam, as filas congestionam e os atrasos fim-a-fim aumentam. O leve aumento na média nos casos em que o λ foram iguais a 5 e 10, mesmo com o alto crescimento dos máximos, se deve ao tamanho grande da amostra e da maior partes dos valores estarem próximos da média. Portanto, apesar das médias crescerem, ainda se aproximam da média quando λ é igual a 1, pois a porcentagem de valores próximos aos máximos foi pequena comparado aos valores próximos de 1 ms.

Com o objetivo de comparar a média amostral com o atraso máximo fim-a-fim definido, inicialmente foi verificado se a amostra seguia a distribuição normal. Para isso, foram aplicados os testes de Kolmogorov-Smirnov e definidos as hipóteses nulas de que as amostras de diferentes λ s não seguiam a distribuição normal.

Os resultados dos testes mostraram os valores do p-value iguais a $2, 2^{-16}$ em todos os casos. Portanto, para todos os três casos de λ , p-values $< 0,05$ e foram rejeitadas as hipóteses nulas de que as amostras seguiam a distribuição normal.

Como foi verificado que a distribuição da amostra, para $\lambda = 1, 5$ e 10 não seguem a normal, para um nível de confiança de 95%. Para verificar se a média dessa amostra é menor que o valor máximo de atraso fim-a-fim definido, foi usado o teste paramétrico de Wilcoxon. Para aplicar o Wilcoxon as seguintes considerações precisam ser feitas:

- μ_0 : 45 ms
- H_1 : rejeita-se a hipótese nula e não rejeita a hipótese da média da amostra ser menor que o atraso máximo fim-a-fim permitido de 45 ms.

$$H_0 : \mu = 45ms$$

$$H_1 : \mu \leq 45ms$$

O resultado do p-values $< 2, 2^{-16}$ encontrado, para $\lambda = 1, 5$ e 10, confirmam a não rejeição às hipóteses alternativas. Isto significa que a média dos dados da amostra nesse cenário de Rede Intraveicular é menor que a restrição máxima do atraso fim-a-fim de 45 ms. A partir desse resultado é possível concluir que, para um nível de significância de 0,05, é rejeitada a hipótese nula de que as médias são iguais ao valor máximo permitido

do atraso e não é rejeitada a hipótese alternativa das médias das amostras serem menores que o atraso fim-a-fim máximo.

5.2 Cenário de Internet das Coisas

Uma abordagem diferente foi usada no contexto do uso da arquitetura para comunicação com dispositivos IoT. Como não foram definidas métricas de avaliação ou restrições de tempo no contexto de aplicação de IoT em veículos, foram calculados os atrasos fim-a-fim com o objetivo de estes valores servirem de referência para determinar atrasos de aplicações veiculares usando a extensão proposta da arquitetura.

O atraso fim-a-fim é calculado a partir de uma requisição de abertura de conexão por uma aplicação a um dispositivo de IoT e termina quando a resposta chega a aplicação. O caminho percorrido pelos dados é mostrado na Figura 16. Assim como no contexto de rede intraveicular, o atraso é obtido somando o caminho de ida e de volta dos dados.

Como mostrado na Tabela 9, a média de atraso fim-a-fim encontrada no contexto da arquitetura provendo comunicação com dispositivos de Internet das Coisas foi de 0,957133 ms. A amostra obtida contém 3000 valores de atrasos fim-a-fim coletados. Foi observado um valor máximo de 1,44 ms e um valor mínimo de 0,701402 ms.

Tabela 9 – Estatística descritiva para o experimento 5.1.4

Tamanho amostral	Mínimo	Máximo	Média	Desvio Padrão	IC
3000	0,701402 ms	1,44 ms	0,957133 ms	0,076334	0,002733

5.3 Cenário integrado

Como o objetivo da extensão proposta da arquitetura é prover integração entre os domínios de Rede Intraveicular e Internet das Coisas, após apresentar os resultados obtidos nos contextos isoladas, tornou-se necessária a avaliação da arquitetura com os domínios integrados.

Para avaliação neste cenário foi utilizada a métrica de atraso fim-a-fim e os instantes iniciais e finais foram coletados de maneira semelhante aos dos cenários isolados.

Apesar de os contextos de IoT e Rede Intraveicular na arquitetura executarem de maneira paralela e não concorrente, o cenário integrado dos domínios foi avaliado para determinar se a execução em paralelo no mesmo hardware poderia causar alguma influência no resultado.

No cenário integrado das tecnologias, a média obtida do atraso fim-a-fim no contexto de Rede Intraveicular foi de 0,963221 ms. Este valor, apesar de menor, é similar aos valores encontrados no cenário isolado com tempos entre requisições definidos de 0,954561 ms e com tempo entre requisições usando $\lambda = 1$, que foi de 0,95437 ms. Além disso, as médias

obtidas nos cenários integrado e isolado usando IoT foram de 0,989091 ms e de 0,957133 ms, respectivamente.

Tabela 10 – Estatística descritiva para o experimento 5.1.3

Aplicação	Tamanho	Mínimo	Máximo	Média	Desvio Padrão	IC
IoT	3000	0,790016 ms	1,270016 ms	0,989091 ms	0,058233	0,002085
SOME/IP	3000	0,70016 ms	1,580032 ms	0,963221 ms	0,069189	0,002477

Após se calcular a média de uma amostra aleatória, provavelmente, se uma nova amostra for realizada, a média aritmética obtida será diferente da primeira obtida. A variabilidade das médias é estimada pelo seu erro padrão. Assim, o erro padrão avalia a precisão do cálculo da média populacional. Os erros padrões encontrados no cenário integrado da arquitetura foram de 0,001063 e 0,001263, para os contextos de Rede Intraveicular e IoT, respectivamente. No cenário de Rede Intraveicular, a diferença entre as médias foi menor que 0,05 ms comparando as simulações nos ambientes isolados e integrados. Contudo, como a diferença da média é pequena e com o erro apresentado, é plausível concluir que a execução da extensão da arquitetura no ambiente integrado não influenciou no aumento do atraso fim-a-fim.

Para o nível de confiança de 95% no cenário integrado de Rede Intraveicular, o intervalo de confiança obtido foi de 0,002085, logo, concluímos que aproximadamente 95% da amostra esteja contida no intervalo de [0,987006 – 0,991176]. Para o mesmo nível de confiança, encontramos no cenário integrado de IoT o intervalo de 0,002477, portanto, um intervalo de [0,960744 – 0,965698].

No contexto de Rede Intraveicular, os testes de normalidade de Kolmogorov-Smirnov foi aplicados e o resultado foi um p-value = $2,2 \cdot 10^{-16}$. Logo, neste contexto foi rejeitada, com nível de confiança de 95%, a hipótese nula da amostra seguir a normalidade.

Para verificar se a média dessa amostra no contexto de Rede Intraveicular é menor que o valor máximo de atraso fim-a-fim definido e sabendo que a distribuição da amostra não segue a normal, com nível de significância de 0,05, para comparar as médias é necessário usar um teste não paramétrico. O teste não paramétrico usado é o teste de Wilcoxon e foram usados os parâmetros μ_0 de 45 ms e a hipótese alternativa de menor que a média.

$$H_0 : \mu = 45ms$$

$$H_1 : \mu \leq 45ms$$

O resultado do teste de hipótese de $2,2 \cdot 10^{-16}$ foi um p-value = $2,2 \cdot 10^{-16}$ e a hipótese alternativa foi verdadeira, isto significa que a médias dos dados da amostra no contexto de Rede Intraveicular no cenário integrado é menor que a restrição máxima do atraso fim-a-fim de 45 ms

5.4 Discussão

O objetivo geral do projeto é apresentar uma extensão da arquitetura de estações ITS para que aplicações ITS possam consumir dados da Rede Intraveicular e se comunicar com dispositivos de Internet das Coisas. Os resultados obtidos mostraram que as médias dos atrasos fim-a-fim utilizando a extensão da arquitetura ITS-S, foram significativamente menores que o máximo permitido de 45 ms. A integração dos módulos funcionou corretamente devido às adaptações que foram necessárias, como por exemplo a inclusão do SomeipGateway, ContikiFacility e SomeipFacility. O atraso máximo fim-a-fim especificado de 45 ms foi cumprido pela arquitetura, cujo valor máximo de atraso observado foi de 6,19008 ms, este valor foi obtido no cenário isolado de Rede Intraveicular e usando requisições com distribuição exponencial.

O p-value é definido como a probabilidade de obter um resultado igual ou extremo ao que é realmente observado, no caso 45ms (atraso máximo permitido), quando a hipótese nula é verdadeira. Um menor valor de P significa que a probabilidade de obter o caso de 45ms foi pequena. Portanto, o valor obtido nos testes iguais a $2 \cdot 2^{-16}$, é apenas o menor número possível que o sistema pode mostrar, é basicamente zero, logo, a probabilidade de que um valor maior que o máximo permitido ocorra tende a 0.

As médias permaneceram similares nos ambientes isolados e integrados com requisições com tempos definidos. Entretanto, quando foi aumentado o valor do λ , logo, reduzindo o tempo entre as requisições, a fila de requisições da SomeipFacility começou a crescer, o que causou um aumento no atraso fim-a-fim.

No contexto IoT, uma aplicação pede a ContikiFacility para abrir uma conexão com um dispositivo IoT, após isso, a comunicação é feita apenas com o envio de um pacote único pela aplicação para o dispositivo, e o dispositivo responde com um pacote para a aplicação, por fim, a conexão é fechada. Como o processo de comunicação utilizado na avaliação é rápido, as portas disponíveis no pool de conexões eram liberadas com velocidade maior que as requisições feitas, portanto, a arquitetura não ficou sobrecarregada. A utilização de 5 portas foi suficiente para a demanda de uma requisição a cada 100 ms.

Vários trabalhos foram propostos para os contextos abordados por este projeto, contudo, nenhum criou uma integração desses cenários e buscou adaptá-los aos futuros carros. O *middleware* SOME/IP para redes Ethernet automotivas faz parte da versão 4.3 da AUTOSAR, a mais recente, e do *middleware* do GENIVI.

Embora (CARIGNANI et al., 2015) tenha proposto uma arquitetura que utiliza o sistema operacional Contiki, não foi especificado como aplicações veiculares fariam para se comunicar com os dispositivos IoT. Na extensão proposta da arquitetura deste trabalho, mostramos desde a abertura da conexão entre a aplicação ITS e o dispositivo, até a finalização da comunicação no capítulo 3.

5.5 *Considerações Finais*

Este capítulo apresentou os resultados obtidos neste trabalho. Foram mostradas as estatísticas descritivas através de tabelas e testes de hipóteses para todas as amostras de resultados adquiridos. Discussões foram apresentadas com o objetivo de explicar os resultados obtidos e evitar dúvidas causadas por interpretações dos resultados.

6 Conclusão e Trabalhos Futuros

Devido ao escopo abrangente de Internet das Coisas e a importância de Redes Veiculares, era esperado que em algum momento estas áreas convergissem. Estas tecnologias podem se completar de várias maneiras, por exemplo, em Cidades Inteligentes ou criando novas funcionalidades para auxiliar os condutores de veículos. Para tornar possível esse ambiente de tecnologias integradas, é preciso que elas se comuniquem e, o mais importante, que aplicações diversas possam utilizá-las de maneira simples. Um grande obstáculo é a variedade de protocolos de comunicação que são utilizados nessas duas áreas, aumentando a complexidade da criação de qualquer aplicação.

O objetivo da extensão proposta da arquitetura de estações ITS é proporcionar intercomunicação entre essas tecnologias e aplicações veiculares. A utilização de uma camada de abstração torna a comunicação transparente quanto aos protocolos de comunicação. Com a utilização da arquitetura, aplicações veiculares podem ter acesso aos dados da rede Ethernet intraveicular, dispositivos de IoT e outros veículos.

Cada domínio de aplicação veicular possui requisitos de atraso máximo definidos, embora existam mais de um valor, como, por exemplo, 140 ms para aplicações de entretenimento, foi usado o atraso máximo de 45 ms nas avaliações por ser o menor. A métrica utilizada para avaliar a arquitetura foi o atraso fim-a-fim desde uma requisição feita por uma aplicação até a resposta chegar a ela novamente.

No cenário de Redes Veiculares, várias requisições poderiam acontecer simultaneamente. Foi feita uma avaliação baseada em quantidade de requisições. Essa avaliação foi apropriada, pois era possível mensurar se a arquitetura suportaria grandes quantidades de requisições.

6.1 Principais Resultados

Diante do cenário apresentado na primeira seção do capítulo 1, seria complexo criar aplicações que pudessem consumir dados oriundos da rede Ethernet intraveicular e se conectar com dispositivos IoT. Este trabalho apresentou a extensão proposta da arquitetura ITS-S cujo objetivo é facilitar a comunicação entre aplicações ITS, dispositivos IoT e ECUs da rede Ethernet intraveicular que implementem o SOME/IP. Esta extensão simplifica a criação de aplicações ITS, pois estas aplicações não precisam conhecer vários protocolos e padrões, uma vez que a extensão cria uma abstração sobre essas regras de comunicação. Segundo mostrado no capítulo 3, as aplicações ITS precisam apenas conhecer os nomes

dos Serviços, no caso da rede Ethernet intraveicular, para fazer uma requisição a ECU que fornece a informação. Para aplicações ITS que desejam se comunicar com dispositivos de IoT, uma tabela é disponibilizada pela extensão contendo as informações para abrir uma conexão.

Foi apresentando também o atraso máximo causado pela utilização da arquitetura nos contextos de IoT e Redes Veiculares separadamente e em uma cenário integrado. Foram utilizados variações de λ em uma distribuição exponencial com o objetivo de alterar os intervalos entre as requisições, assim, trazendo um fator realista e avaliando a arquitetura sob diferentes quantidades de requisições.

Dentre os principais resultados pode-se destacar:

- A criação da extensão da arquitetura de estações ITS para comunicação entre aplicações ITS, dispositivos IoT e ECUs da rede Ethernet intraveicular que implementam SOME/IP.
- A avaliação da extensão proposta criada com a metodologia apresentada no capítulo 4.
- A utilização da arquitetura nos cenários de aplicação isoladas apresentaram um atraso fim-a-fim abaixo da restrição máxima definida.
- No cenário integrado de rede intraveicular e IoT, o atraso fim-a-fim também ficou abaixo da restrição máxima permitida.
- A extensão proposta funcionou corretamente tanto no cenário isolado quanto no integrado e garantiu o tempo de atraso menor que máximo permitido em todos os casos.

6.2 Trabalhos Futuros

Durante o desenvolvimento do trabalho foi observada a possibilidade de expansão do mesmo para diferentes vertentes. Além disso, com a análise dos resultados foi identificada a necessidade da execução de diferentes experimentos utilizando vários outros cenários. No entanto, estes não estavam presentes no escopo deste trabalho.

Diante de uma análise micro do trabalho apresentado, as principais contribuições para trabalhos futuros que possam agregar e aprimorar são:

- Foi usada a distribuição exponencial para simular o comportamento das requisições feitas por aplicações veiculares a ECUs da rede Ethernet intraveicular. Como foi explicado na seção 4.5.1 do capítulo 4, a distribuição exponencial, talvez, não seja a melhor distribuição para a avaliação. Portanto, identificar uma distribuição estatística que represente melhor as requisições de aplicações veiculares aos nós das

redes intraveiculares seria uma grande contribuição para este trabalho. Para conseguir esta distribuição, duas maneiras são possíveis. A primeira é conseguir os dados direto de uma montadora de veículos, a segunda seria usar um *sniffer* na rede intraveicular de um veículo que possua Ethernet e encontrar a distribuição a partir dos dados obtidos.

- Alguns trabalhos relacionados citados na seção 2.3 do capítulo 2, por exemplo, (QIN et al., 2013), (HE; YAN; XU, 2014) e (CONSORTIUM, 2015), utilizam sistemas em nuvem. Aplicações produzidas para os Carros Conectados serão, grande parte, construídas como aplicações WEB e hospedadas na nuvem. A utilização de aplicações em nuvem irá agregar valor ao trabalho, além disso, aproximará o real cenário da arquitetura de comunicação dos Carros Conectados.
- Neste trabalho foram apresentadas apenas aplicações ITS que se comunicam através do padrão ETSI que será utilizado na Europa. Contudo, na América do Norte, outro padrão será adotado, o IEEE WAVE. Portanto, estender a extensão criada neste trabalho para que aplicações ITS possam se comunicar utilizando os padrões ETSI e IEEE WAVE seria uma grande contribuição. Esta integração poderia ser simulada criando uma integração entre os simuladores Veins, que simula comunicação entre carros utilizando o IEEE WAVE e o Artery, que simula a comunicação utilizando o padrão ETSI.

Referências

- 100BASE-T1. 2015. <<https://standards.ieee.org/findstds/standard/802.3bw-2015.html>>. Accessed: 2017-07-10.
- ABIRESEARCH. *More Than 30 Billion Devices Will Wirelessly Connect to the Internet of Everything in 2020*. 2013. <<https://www.abiresearch.com/press/more-than-30-billion-devices-will-wirelessly-conne/>>. Accessed: 2017-07-10.
- AGLAIA, H. *ISO 17215 and SOME/IP. A solution for more than Cameras*. 2014. <https://vector.com/portal/medien/cmc/events/commercial_events/Automotive_Ethernet_Symposium_AES14/AES14_05_Gutzmer_HellaAglaia_Lecture.pdf>. Accessed: 2017-07-10.
- ARANITI, G.; CAMPOLO, C.; CONDOLUCI, M.; IERA, A.; MOLINARO, A. Lte for vehicular networking: a survey. *IEEE Communications Magazine*, IEEE, v. 51, n. 5, p. 148–157, 2013.
- ATZORI, L.; IERA, A.; MORABITO, G. The internet of things: A survey. *Computer networks*, Elsevier, v. 54, n. 15, p. 2787–2805, 2010.
- AUTOSAR. Autosar—a worldwide standard is on the road. In: *14th International VDI Congress Electronic Systems for Vehicles, Baden-Baden*. [S.l.: s.n.], 2009. v. 62.
- AVB. *IEEE 802.1 AVB Task Group. IEEE 802.1 Audio/Video Bridging (AVB)*. 2011. <<http://www.ieee802.org/1/pages/avbridges.html>>. Accessed: 2017-07-10.
- AVR-RAVEN. 2008. <<http://www.atmel.com/tools/avraven.aspx>>. Accessed: 2017-07-10.
- BROADCOM. *Product Brief BCM89810, BroadR-Reach® Single-Port Automotive Ethernet Transceiver*. 2011. <<http://www.broadcom.com/collateral/pb/89810-PB00-R.pdf>>.
- CAM, E. *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service*. 2011. <http://www.etsi.org/deliver/etsi_ts/102600_102699/10263702/01.02.01_60/ts_10263702v010201p.pdf>. CARIGNANI, M.; FERRINI, S.; PETRACCA, M.; FALCITELLI, M.; PAGANO, P. A prototype bridge between automotive and the iot. In: IEEE. *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on*. [S.l.], 2015. p. 12–17.
- CONSORTIUM icSi. *ICT ICSI*. 2015. <<http://www.ict-icsi.eu/>>.
- DENM, E. *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic*

- Service. 2010. <http://www.etsi.org/deliver/etsi_ts/102600_102699/10263703/01.01.01_60/ts_10263703v010101p.pdf>.
- DUNKELS, A.; QUATTLEBAUM, R.; ÖSTERLIND, F.; OIKONOMOU, G.; ALVIRA, M.; TSIFTES, N.; SCHMIDT, O. *Contiki: The open source OS for the Internet of things*. 2012. <<http://www.contiki-os.org/>>.
- ETSI, E. 302 665 v1. 1.1: Intelligent transport systems (its). *Communications architecture*, 2010–09.
- ETSI, T. Ts 102 637-1,“. *Intelligent Transport System (ITS)*, 2010–09.
- ETSI, T. Ts 102 894-1. *Intelligent Transport System (ITS)*, 2013.
- FORNI, A. A.; MEULEN, R. van der. *Gartner Says Connected Car Production to Grow Rapidly Over Next Five Years*. 2016. <<http://www.gartner.com/newsroom/id/3460018>>.
- FRAGA-LAMAS, P.; FERNÁNDEZ-CARAMÉS, T. M.; SUÁREZ-ALBELA, M.; CASTEDO, L.; GONZÁLEZ-LÓPEZ, M. A review on internet of things for defense and public safety. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 16, p. 1644, 2016.
- G5. *ETSI ITS G5*. 2012. <http://www.etsi.org/deliver/etsi_en/302600_302699/302663/01.02.00_20/en_302663v010200a.pdf>. Accessed: 2017-07-10.
- GENG, Y.; CASSANDRAS, C. G. New “smart parking” system based on resource allocation and reservations. *IEEE Transactions on Intelligent Transportation Systems*, IEEE, p. 1129–1139, 2013.
- GENIVI. The genivi alliance. *AUTOSAR-AUTomotive Open System ARchitecture: High-impact Strategies-What You Need to Know: Definitions, Adoptions, Impact, Benefits, Maturity, Vendors*, Emereo Publishing, p. 50, 2012.
- HE, W.; YAN, G.; XU, L. D. Developing vehicular data cloud services in the iot environment. *IEEE Transactions on Industrial Informatics*, IEEE, p. 1587–1595, 2014.
- ITS-S. *Intelligent Transport Systems (ITS); Communications Architecture*. 2010. <http://www.etsi.org/deliver/etsi_en/302600_302699/302665/01.01.01_60/en_302665v010101p.pdf>. JESCHKE CHRISTIAN BRECHER, H. S. D. B. R. e. S. *Industrial Internet of Things: Cybermanufacturing Systems*. [S.l.]: Springer, 2016.
- KAA. *Discover unlimited opportunities for the Internet of Things*. 2017. <https://www.etsi.org/deliver/etsi_en/302600_302699/302665/01.01.01_60/en_302665v010101p.pdf>.

- [//www.kaaproject.org/](http://www.kaaproject.org/)>. KOVATSCH, M.; DUQUENNOY, S.; DUNKELS, A. A low-power coap for contiki. In: IEEE. *Mobile Adhoc and Sensor Systems (MASS), 2011 IEEE 8th International Conference on*. [S.l.], 2011. p. 855–860.
- KUSHALNAGAR, e. a. *IPv6 over Low power WPAN (6lowpan)*. 2007. <<https://datatracker.ietf.org/wg/6lowpan/>>. Accessed: 2017-07-10.
- LAJMI, H.; ALIMI, A. M.; AJILI, S. Using ethernet technology for in-vehicle's network analysis. In: IEEE. *Computational Intelligence, Communication Systems and Networks (CICSyN), 2013 Fifth International Conference on*. [S.l.], 2013. p. 353–358.
- LEE, J. J.; GUPTA, M. A new traffic model for current user web browsing behavior. *Intel corporation*, 2007.
- LIN, F. J.; REN, Y.; CERRITOS, E. A feasibility study on developing iot/m2m applications over etsi m2m architecture. In: IEEE. *Parallel and Distributed Systems (ICPADS), 2013 International Conference on*. [S.l.], 2013. p. 558–563.
- LIN, X.; SUN, X.; HO, P.-H.; SHEN, X. Gsis: A secure and privacy-preserving protocol for vehicular communications. *IEEE Transactions on vehicular technology*, IEEE, v. 56, n. 6, p. 3442–3456, 2007.
- MATHEUS, K.; KÖNIGSEDER, T. *Automotive ethernet*. [S.l.]: Cambridge University Press, 2017.
- MBED. *Arm mbed project*. 2013. <<https://www.mbed.com/en/>>.
- MEULEN, R. van der; RIVERA, J. *Gartner Says By 2020, a Quarter Billion Connected Vehicles Will Enable New In-Vehicle Services and Automated Driving Capabilities*. 2015. <<http://www.gartner.com/newsroom/id/2970017>>.
- MICAZ. 2011. <http://www.memsic.com/userfiles/files/Datasheets/WSN/micaz_datasheet-t.pdf>. Accessed: 2017-07-10.
- MINT. *Linux MINT*. 2006. <<https://linuxmint.com/>>.
- MONTENEGRO, e. a. G. *Transmission of IPv6 Packets over IEEE 802.15.4 Networks*. 2007. <<https://tools.ietf.org/html/rfc4944>>. Accessed: 2017-07-10.
- MOREIRA, F. T.; MAGALHÃES, A.; RAMOS, F.; VAIRINHOS, M. The power of the internet of things in education: An overview of current status and potential. In: *Conference on Smart Learning Ecosystems and Regional Development*. [S.l.]: Springer, 2017. p. 51–63.
- NIEMINEN, e. a. J. *IPv6 over BLUETOOTH(R) Low Energy*. 2015. <<https://tools.ietf.org/html/rfc7668>>. Accessed: 2017-07-10.
- OMNET. 2001. <<https://www.omnetpp.org/>>. Accessed: 2017-07-10.