UNIVERSIDADE FEDERAL DE SERGIPE

PRÓ-REITORIA DE PÓS-GRADUAÇÃO E PESQUISA

NÚCLEO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

GEORGE LEITE JUNIOR

I9VANET: um modelo de arquitetura de software para rede veicular em nuvem

São Cristóvão

2017

GEORGE LEITE JUNIOR

I9VANET: um modelo de arquitetura de software para rede veicular em nuvem

Versão original

Dissertação apresentada à Pró-Reitoria de Pós-Graduação e Pesquisa da Universidade Federal de Sergipe para obtenção do título de Mestre em Ciência da Computação pelo Programa de Pós-graduação em Ciência da Computação.

Área de concentração: Redes de Computadores e Sistemas Distribuídos

Orientador: Prof. Dr. Douglas D. J. de Macedo

Coorientador: Prof. Dr. Rogério Patrício Chagas do Nascimento

São Cristóvão

2017

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL UNIVERSIDADE FEDERAL DE SERGIPE

Leite Junior, George
L533i I9VANET : um modelo de arquitetura de software para rede veicular em nuvem / George Leite Junior ; orientador Douglas D. J. de Macedo. – São Cristóvão, 2016.
102 f. : il.

Dissertação (mestrado em Ciências da computação)– Universidade Federal de Sergipe, 2016.

1. Programas de computador - Testes. 2. Redes de computadores. 3. Computação em nuvem. 4. Engenharia de software. 5. Software - Testes. I. Macedo, Douglas D. J. de. II. Título.

CDU 004.4

Dissertação de autoria de George Leite Junior, sob o título **"I9VANET: um modelo de arquitetura de software para rede veicular em nuvem"**, apresentada à Pró-Reitoria de Pós-Graduação e Pesquisa da Universidade Federal de Sergipe, para obtenção do título de Mestre em Ciência da Computação pelo Programa de Pós-graduação em Ciência da Computação, na área de concentração Redes de Computadores e Sistemas Distribuídos, aprovada em _____ de _____ de _____ pela comissão julgadora constituída pelos doutores:

Prof. Dr. Douglas D. J. de Macedo
Presidente
Instituição: Universidade Federal de Santa Catarina (UFSC)

Prof. Dr. Edward David Moreno Ordonez
Instituição: Universidade Federal de Sergipe (UFS)

Prof. Dr. Mario Antonio Ribeiro Dantas
Instituição: Universidade Federal de Santa Catarina (UFSC)

Dedico a minha família, fonte de inspiração, compreensão e resiliência

Agradecimentos

Quero agradecer a todos aqueles que sempre confiaram em mim, desde sempre. À minha esposa Manoela Vieira, pelo apoio incondicional e constante incentivo. Às minhas queridas filhas Tâmara e Brenda. Aos meus pais, por me proporcionar educação e valores, nunca deixaram de me amar e nem de confiar em mim. A todos os meus familiares, irmãos, primos e tios, seus corações estão comigo e o meu com vocês. Aos meus verdadeiros amigos. Dedico também ao meu orientador Prof. Dr. Douglas D. J. de Macedo e coorientador Prof. Dr. Rogério P. C. do Nascimento, pela confiança, paciência, incentivo e orientação.

"Ama-se mais o que se conquista com esforço."

(Benjamin Disraeli)

Resumo

LEITE, George Junior. **I9VANET: um modelo de arquitetura de software para rede veicular em nuvem**. 2017. 103 f. Dissertação (Mestrado em Ciência da Computação) – Pró-Reitoria de Pós-Graduação e Pesquisa, Universidade Federal de Sergipe, Sergipe, 2017.

Em consequência do crescimento populacional, as grandes cidades enfrentam problemas cotidianos relacionados à mobilidade urbana tais como: congestionamentos, baixa qualidade das rodovias, ineficiência de transportes públicos, entre outros. Iniciativas de sistemas de transportes inteligentes (ITS) agem como uma solução eficiente para melhorar o funcionamento e desempenho dos sistemas de tráfego, reduzindo congestionamentos e aumentando a segurança para os cidadãos. Atualmente, pesquisadores vem buscando nas redes veiculares ad-hoc (VANET) uma possível solução para os problemas referentes à mobilidade urbana. Contudo, VANETs ainda apresentam uma série de desafios que devem ser resolvidos para que seu uso seja consolidado. Desse modo, o presente trabalho apresenta uma arquitetura e plataforma denominada I9VANET, cujo intuito é o gerenciamento de uma rede veicular de maneira virtualizada por meio da computação em nuvem, para auxiliar nas soluções dos principais desafios relacionados à VANETs tais como: interferências na comunicação devido a árvores e prédios; alta mobilidade dos nós; alta e baixa densidade dos nós em uma área; garantia de confidencialidade, integridade e disponbilidade, autenticidade e não repúdio. Após realização de experimentos em laboratório, foi constatada a viabilidade técnica para a utilização de velocidades definidas para os modelos de telefonia móvel 3G, 4G e 5G, satisfazendo os critérios de eficiência apresenados por Papadimitratos et al. (2008).

Palavras-chaves: Rede Veicular Computação em Nuvem e Arquitetura Distribuída.

Abstract

LEITE, George Junior. **I9VANETs: a software architecture model for a vehicular network**. 2017. 103 p. Dissertation (Master of Science of Computer) – Dean of Graduate Studies and Research, Federal University of Sergipe, Sergipe, 2017.

As a result of population growth, large cities face daily problems related to urban mobility such as congestion, poor quality of roads, inefficiency of public transportation, among others. Intelligent Transport Systems (ITS) initiatives act as an efficient solution to improve the functioning and performance of fraffic systems, reducing congestion and increasing safety for citizens. Currently, researchers have been searching the ad-hoc vehicular networks (VANET) for a possible solution to the problems related to urban mobility. However, VANET still has a number of challenges that must be addressed in order for its use to be consolidated. In this way, the present work shows an architecture called I9VANET, whose intention is the management of a vehicular network in a virtualized way through cloud computing, to assist in the solutions of the main challenges related to VANET such as: interference in communication due to trees and buildings; High node mobility; High and low density of nodes in an area; Guarantee of confidentiality, integrity and availability, authenticity and non-repudiation. After conducting experiments in the laboratory, it was verified the technical viability for the use of defined speeds for 3G, 4G and 5G mobile telephony models, satisfying the efficiency criteria when compared to other works in the literature, which used Ad- Hoc.

Keywords: Vehicle Network. Cloud Computing and Distributed Architecture .

Lista de figuras

Lista de tabelas

Lista de abreviaturas e siglas

AI2AI	Agente da Infra-estrutura para outro Agente da Infra-estrutura
AODV	Ad hoc On Demand Distance Vector
AV2AI	Agente do Veículo para Agente da Infra-estrutura
AV2AV	Agente do Veículo para outro Agente do Veículo
AVC	Autonomous Vehicular Clouds
BROADCOMM	BROADcast COMMunications
BS	Base Station
CBLR	Cluster-Based Location Routing algorithm
COIN	Clustering for Open IVC Networks
DDoS	Distributed Denial of Service
DMS	Decision Making subsystem
DoS	Denial of Service
DRG	Distributed Robust Geocast
DSDV	Destination Sequenced Distance Vector
DSR	Dynamic Source Routing
Geocast	Geocast Routing
GeOpps	Geographical Opportunistic routing for vehicular networks
GPS	Global Position System
GPSR	Greedy Perimeter Stateless Routing
GSR	Global State Routing
HTTP	Hypertext Transfer Protocol
HVC	Hybrid Vehicular Clouds

ICP	Infraestrutura de Chaves Públicas
I2AI	Infra para Agente da Infra
ITS	Intelligent Transport System
IVC	Inter-Vehicle Communications
MANET	Mobile Ad-hoc Network
MCC	Mobile Cloud Computing
MCDS	Minimum Connected Dominating Set
OBU	OnBoard Unit
OLSR	Optimzed Link State Routing
PRAODV-M	PReemptive AODV - Maximum
PRAODV	PReemptive AODV
QoS	Quality of Service
RMI	Remote Method Invocation
ROVER	RObust Vehicular Routing
RPC	Remote Procedure Call
RSU	Road Side Unit
SD	Sistemas Distribuídos
SDN	Software-Defined Networking
SNR	Signal to Noise Ratio
SSL	Secure Socket Layer
TLS	Transport Layer Security
V2AV	Veículo para Agente do Veículo
V2I	Vehicle to Infra

V2V	Vehicle to Vehicle
VANET	Vehicle Ad-hoc Network
VC	Vehicular Clouds
VCC	Vehicle Cloud Computing
VCC	Vehicle Cloud Computing
VCN	Vehicular Cloud Networking
VUC	VANET using Clouds
WRP	Wireless Routing Protocol
WS	WebSocket
WSN	Wireless Sensor Network
WSS	WebSocket Secure
ZOR	Zone of Relevance

Sumário

1 Introdução

A conectividade inerente às cidades inteligentes abre uma fronteira bastante promissora no que se refere ao controle de acesso às informações (LI, 2010) (ZHU; LIU, 2015) e arquiteturas distribuídas para sistemas inteligentes de transporte (ICE et al., 2001). O que torna necessário a criação de propostas que auxiliem idéias no âmbito de cidades inteligentes.

Um sistema inteligente de transporte ou *Intelligent Transportation System* (ITS) representa "a aplicação de sensores avançados, computadores, dispositivos eletrônicos e tecnologias de comunicação e gerenciamento estratégico integrado visando melhorar a segurança e a eficiência do sistema de gerenciamento de tráfego" (NASIM; KASSLER, 2012).

Ainda segundo Nasim e Kassler (2012), o ITS possui subsistemas e tem como objetivo atuar de forma direcionada e específica sobre subáreas do gerenciamento de transporte, buscando garantir a eficiência e qualidade da mobilidade urbana e são subdivididos em seis áreas de gestão, são elas:

- Sistema Avançado de Gestão de Tráfego (ATMS)
- Sistema Avançado de Informações para Viajantes (ATIS)
- Sistema Avançado de Transporte Público (APTS)
- Sistema de Operação de Veículos Comerciais (CVO)
- Sistema Avançado de Controle de Veículos (AVCS)
- Sistema de Coleta Eletrônica de Pedágio (ETC)

Os Sistemas Inteligentes de Transporte são, as mais importantes aplicações de uma rede veicular, fornecendo serviços de segurança rodoviária (XU et al., 2003) ou informações relativas às situações de tráfego. De acordo com Sumra, Hasbullah e Manan (2011) e Luís (2009), o principal objetivo de uma VANET (*Vehicular ad-hoc Network*) é prover segurança aos motoristas e passageiros nas estradas.

De acordo com Ball e Dulay (2010) e Losilla et al. (2011), a segurança no trânsito é a motivação principal para as pesquisas no âmbito das redes veiculares. Da mesma maneira, conforme a Organização Mundial da Saúde (OMS), acidentes acometidos por veículos são responsáveis por mais de um milhão de mortes e 50 milhões de feridos anualmente em todo o mundo (PEDEN et al., 2004). Nos Estados Unidos e no Brasil, acidentes relacionados a

veículos são a terceira maior causa de mortalidade evitável e incapacitação profissional precoce (SYSTEMATICS; MEYER, 2011) (DIB et al., 2007).

Ainda afirmado por Losilla et al. (2011), os congestionamentos também são uma grande preocupação como mostra a Figura 1, congestionamento na pista expressa, entre a rodovia Castelo Branco até a ponte Imigrante Nordestino. Só nos EUA, o congestionamento representa 115 bilhões de dólares em custos de combustível (TTI, 2014), com números semelhantes em outros países desenvolvidos. As baixas de tráfego no mundo ascendem a 1,17 milhões por ano. Neste contexto, os Sistemas de Transporte Inteligentes visam melhorar a eficiência e a segurança dos transportes através do uso avançado de processamento de informação, comunicação, controle, bem como, o uso de novas tecnologias.

Figura 1 – Imagem do trânsito de São PauloBrasil (em 02/092016) (DANTAS, 2011)
.

Segundo o IPEA (Instituto de Pesquisa Econômica Aplicada do Brasil), a mobilidade urbana, nos dias atuais, é um dos principais problemas dos grandes centros. Os reflexos sobre o transporte urbano são evidentes, caracterizados principalmente pelo aumento do tráfego nas vias e consequentemente dos congestionamentos (CARVALHO et al., 2010).

Porém, baseado em Gupte e Younis (2012), é possível gerir o tráfego de forma eficiente utilizando redes de veiculos ad-hoc (VANET) de maneira eficiente e autonoma.

Segundo Cavalcanti (2008), a crescente quantidade de dispositivos eletrônicos que podem ser embarcados em veículos automotores como DVD, TV, GPS e telefone celular, faz com que os veículos deixem de ser apenas um meio de transporte e passem a oferecer uma rede de serviços e entretenimento para os condutores e/ou passageiros.

Diante desse contexto, o presente trabalho apresenta como principal contribuição um modelo de arquitetura de software flexível e extensível, que utiliza rede veicular e computação em nuvem, com a finalidade de proporcionar uma alternativa para os principais desafios relacionados à VANETs tais como:

- **meio físico**: interferências devido a prédios, árvores e outros obstáculos;
- **alta mobilidade:** dificulta a troca de informações mais completas;
- **topologia**:VANET possui uma característica dinâmica, devido à velocidade que os veículos se movimentam;
- **baixa densidade**: quando a densidade de tráfego é baixa e os veículos estão distantes uns dos outros;
- **alta densidade**: muitos veículos em uma pequena área faz com que a quantidade de mensagens trocadas seja um problema;
- **segurança**: como as VANETs suportam aplicações de emergência em tempo real e lidam com informações críticas de segurança no trânsito, estas devem satisfazer os seguintes requisitos de segurança: confidencialidade, integridade, disponibilidade, autenticidade, privacidade e não repudiação para prover segurança na comunicação dos dados (SAMARA; AL-SALIHY; SURES, 2010) (MATOS et al., 2013).

1.1 Justificativa, Problemática e Hipótese

1.1.1 Justificativa

É crescente o número de pesquisas sobre VANETs com intuito de aumentar a segurança viária, bem como ofertar e comercializar novos serviços a motoristas e passageiros. VANETs que usam veículos como nós móveis, são uma subclasse de rede móveis ad hoc chamadas de MANETs (*Mobile Ad hoc Network*), elas fornecem comunicação entre os

veículos próximos e entre veículos e equipamentos à beira da rodovia mas, aparentemente difere de outras redes por suas próprias características do ambiente.

Diante da evolução das redes veiculares e da necessidade de garantir o tráfego de dados mediante a alta mobilidade dos nós em uma VANET, constata-se a importância de utilizar tecnologias que permitam auxiliar na solução dos principais desafios relacionados a este tipo de rede. Portanto, para melhorar a segurança, eficiência e conforto relacionados ao trânsito nas grandes cidades, é preciso contornar os obstáculos antes mesmo de uma solução ser colocada em prática.

Os nós numa rede VANET são muito dinâmicos pois os veículos possuem velocidade e direção variável. A alta mobilidade dos nós conduz a uma topologia de rede dinâmica caracterizada pela constante perda de comunicação fazendo com que os algoritmos de roteamentos tornem-se complexos e limitados. Normalmente, VANETs apresentam três aspectos para pesquisas: roteamento, segurança e privacidade, bem como suas aplicações (LIANG et al., 2015).

O roteamento em VANETs é baseado na comunicação sem fio e devido à topologia dinâmica e conectividade intermitente, a maioria dos algoritmos em MANETs não estão disponíveis para os vários cenários VANETs, tornando o tema ainda em aberto e sendo foco de várias pesquisas na atualidade com objetivo de dar confiabilidade na comunicação levando em consideração números de emissores e receptores e tipos de roteamento.

Comentado por Cavalcanti (2008), os nós de uma rede veicular apresentam como principal caracteristica, a alta mobilidade, por serem dinâmicos e apresentarem rápidas variações das condições do meio sem fio são os principais geradores de desafios. O alto dinamismo representa um contratempo para as comunicações, visto que nem sempre o tempo de contato dos veículos é suficiente para estabelecer uma comunicação efetiva.

Trabalhos como Hadaller et al. (2007), Nandan et al. (2005) e Chen et al. (2006), apresentam propostas que vão desde a implementação de uma infra-estrutura fixa até definir esquemas de agrupamento de nós, visando reduzir a latência de entrega de mensagens para sistemas de segurança. Nandan et al. (2005) propuseram uma comunicação eficiente para sistemas P2P com disponibilização de conteúdo utilizando um mecanismo de partição dos arquivos disponíveis em pequenos pedaços. Este particionamento visa promover uma forma de transferência distribuída, na qual é possível que um nó esteja recebendo dois pedaços distintos do mesmo arquivo de duas fontes diferentes e simultaneamente. Cabe ressaltar que, em grande parte, as propostas visam agilizar o processo de transferência de

dados em VANETs e implementam mecanismos específicos para minimizar os problemas gerados pela alta mobilidade dos nós.

Segundo Falchetti, Azurdia-Meza e Cespedes (2015), os principais desafios técnicos que são abordados pela maioria dos pesquisadores podem ser agrupados em seis categorias: mobilidade, latência, confiabilidade, criticidade (prioridade das mensagens), topologia e segurança.

Portanto, pesquisar sobre redes veiculares e computação em nuvem, traz a possibilidade de construção de uma plataforma capaz de criar uma VANET com gerenciamento virtualizado em nuvem, facilitando a comunicação entre os nós virtuais da rede e simplificando a implementação dos algoritmos de roteamento, segurança e aplicações.

Com o uso da computação em nuvem, os autores Liu, Chen e Chakraborty (2015) utilizaram um controlador SDN (*Networking Defined Software*) para enviar mensagens para os veículos a partir de um RSU (*Road-side Unit*) conectado ao controlador, porém os veículos deveriam pertencer na mesma geolocalização.

Foi criado por HAJJI e BARGAOUI (2015), a plataforma Testbed que serve para testar a implementação real dos roteadores móveis em uma rede veicular virtualizada. A característica principal consiste na sua capacidade em satisfazer ao mesmo tempo as exigências de mobilidade do veículo e o número elástico de roteadores móveis aplicados.

Eltoweissy, Olariu e Younis (2010) pela primeira vez realizaram uma mudança de paradigma, imaginando um ambiente de VANET baseado em nuvem. Eles vieram com uma nova noção de VANET chamado AVC (*Nuvens Veiculares Autónomas*). Yan et al. (2013) delinearam os desafios de segurança e privacidade nas nuvens veiculares. No entanto, Eltoweissy, Olariu e Younis (2010), discutiram sobre nuvens veiculares de maneira abstrata, sem mencionar uma arquitetura em particular.

Hussain et al. (2012) propuseram três cenários de arquitetura para VANETs baseados em nuvem ou seja VC (*Vehicular Clouds*), VUC (*VANET using Clouds)*), e HVC (*Hybrid Vehicular Clouds*) e propôs um esquema conhecido como TIaaS (*Information of Traffic as Service*), onde a infraestrutura de nuvem atua para fornecer informações de tráfego para os veículos que se deslocam sobre a estrada.

Qin, Huang e Zhang (2012) propuseram um esquema de roteamento, para VANETs, baseado em nuvem chamado VehiCloud que fornece serviços de roteamento através da infraestrutura de nuvem. Veículos compartilham a sua atual e futuras informações de

localização na forma de *waypoints* com infraestrutura de nuvem e, em seguida, a nuvem lhes proporciona uma melhor informação de roteamento.

De acordo com Falchetti, Azurdia-Meza e Cespedes (2015), a integração de VANETs com a nuvem aumenta a utilização das capacidades computacionais que são subutilizadas por aplicações de segurança e supera os problemas de roteamento em comunicação V2V (*Vehicle to Vehicle*).

Lee et al. (2014) visam interligar recursos OBU (*On Board Unit*) e RSU (*Road Side Unit*) em nuvem para tarefas cooperativas sensoriais, de armazenamento e de computação, enquanto outros (HUSSAIN et al., 2012) (MERSHAD; ARTAIL, 2013) propõem que as RSUs ajam como *gateways* para as OBUs para acesso às nuvens tradicionais.

Foi proposto por Lee et al. (2014), o VCN (*Vehicular Cloud Networking*) que está sendo visto como uma revolução para modernizar a tradicional VANET, que integra informações de redes e *Cloud Computing* com VANETs tradicionais. No VCN, veículos e RSU compartilham seus recursos em uma plataforma virtual como ilustrado na Figura 2.

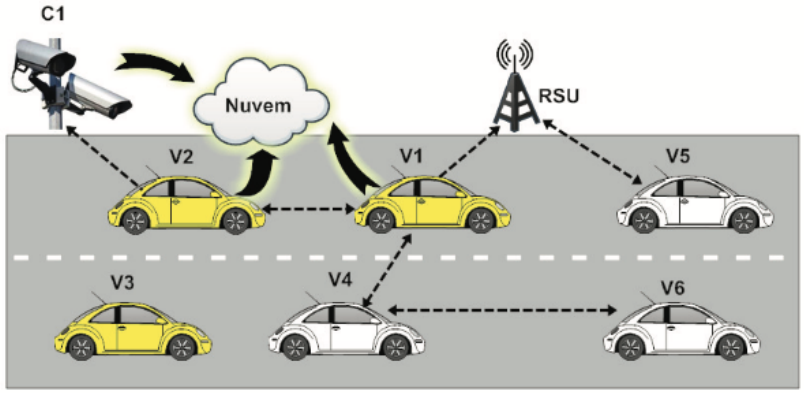


Figura 2 – Modelo do VCN (LEE et al., 2014)

.

Gerla (2012) introduziu um novo modelo computacional, a computação veicular em nuvem (VCC). O VCC é uma variação da computação móvel em nuvem (MCC), que começa a partir de um modelo convencional de computação em nuvem. Sendo que a maioria das consultas são sobre o mundo que nos rodeia, ou seja possui relevância local, e os veículos são os melhores exemplos deste ambiente.

Sookhak, Yu e Tang (2017) apresentou um novo metodo para abordar o problema de compartilhamento de dados em redes veiculares, utilizando a técnica de emparelhamento bilinear. Neste caso, foi utilizada computação em nuvem para armazenar a grande quantidade de dados e executar o processo de re-criptografia.

Foi apresentado por Comi et al. (2015), uma abordagem evolutiva baseada na clonagem de agentes, ou seja, um mecanismo de reprodução de agentes que permite que os fornecedores substituam um agente "insatisfatório" agindo em um "contexto de nuvem" por um clone de um agente existente com conhecimento adequado e boa reputação no contexto multi-nuvem. Por esta abordagem, os desempenhos dos agentes de nuvem podem ser melhorados porque são substituídos por clones de agentes que têm mostrado um comportamento melhor.

1.1.2 Problemática

Como criar uma plataforma em nuvem capaz de permitir a construção de aplicações voltadas para VANETs, considerando o desenvolvimento simplificado e extensível de maneira que o cerne principal da plataforma seja abstraído das novas funcionalidades, mantendo o foco nos principais desafios atualmente encontrados em redes veiculares e permita a escalabilidade.

1.1.3 Hipótese

É possível criar uma plataforma aberta, flexível e extensível capaz de permitir o gerenciamento de redes veiculares como serviço (VaaS) por meio de uma solução em nuvem, sendo capaz de atender aos requisitos mínimos de tempo para a maioria das aplicações voltadas para redes veiculares?

1.2 Objetivos da Dissertação

1.2.1 Objetivo Geral

Propor uma arquitetura de *software* flexível e extensível, com capacidade de gerenciar nós de uma rede VANET por meio da computação em nuvem, realizando a comunicação entre os elementos de forma virtual na tentativa de corroborar com a solução de alguns dos principais desafios relacionados às redes veiculares.

Tabela 1 – Comparação dos trabalhos relacionados

Propostas	Foco em Segurança	Algo. de Roteamentos	Comp. em Nuvem	Ad-Hoc	V2V	V2I	I2I	Sist. Distribuídos
Liu et al.Liu, Chen e Chakraborty (2015)			X					
Hajji e Bargaoui (HAJJI; BARGAOUI, 2015)		X	X					
Eltoweissy et al. Eltoweissy, Olariu e Younis (2010)			X	X				
Yan et al Yan et al. (2013)	X		X	X				
Hussain et al. Hussain et al. (2012)			X					
Qin et al. Qin, Huang e Zhang (2012)			X					
Falchetti et al. Falchetti, Azurdia-Meza e Cespedes (2015)			X		X			
Lee et al. Lee et al. (2014)			X					
Gerla Gerla (2012)			X					
Sookhak et al. Sookhak, Yu e Tang (2017)	X		X					
Comi et al. Comi et al. (2015)			X					X
Dorri et al. **??**)	X		X					
I9VANET	X	X	X		X	X	X	X

1.2.2 Objetivos Específicos

- Elaborar uma arquitetura de *software* de maneira que permita a extensibilidade, flexibilidade e escalabilidade para suportar o gerenciamento de redes veiculares em nuvem;
- Construir uma plataforma seguindo os requisitos da arquitetura definida;

- Implememtar cada camada da arquitetura de acordo com os trabalhos encontrados na literatura;

1.3 Metodologia

No desenvolvimento desta dissertação, foram empregadas a pesquisa bibliográfica, pesquisa quantitativa e a pesquisa experimental. Na pesquisa bibliográfica foi verificado o estado da arte sobre redes veiculares e seus principais desafios. A pesquisa quantitativa foi utilizada para poder quantificar dentre as aplicações relacionadas à VANETs, quais os tempos de comunicação necessários para viabilidade da solução. E por fim, a pesquisa experimental, pois foi criada uma plataforma para que pudessem ser realizados experimentos e gerados dados em um ambiente simulado para responder a hipótese levantada na problemática.

Na elaboração do escopo e das metas desta pesquisa, foi definida uma arquitetura de software que permitisse construir uma plataforma modular de maneira que cada módulo funcione de forma independente, abstraindo detalhes de implementação dos demais, possibilitando usufruir de "contratos" conceituais da orientação a objetos e da arquitetura *Publish-Subscribe* na comunicação entre os módulos.

Os cenários para os experimentos foram definidos utilizando movimentações reais, extraídas de um software de monitoramento de uma empresa de taxi, com 120 veículos, na cidade de Aracaju-Sergipe Brasil, os quais foram extraídos 12 milhões de registros corresponente ao período de 10 de janeiro a 30 de junho de 2016. Cada registro contém dados referente à posição geográfica, latitude e longitude e velocidade.

Por meio do uso do simulador de rede Mininet, foram criados *hosts* virtuais, os quais representavam os veículos, onde cada um enviava uma mensagem com a localização geográfica, velocidade, direção e horário para o servidor em nuvem. Sendo estabelecido como métrica e técnica de avaliação desta pesquisa, o tempo de envio da mensagem pelo veículo e o retorno da confirmação de entrega e o tempo de processamento de cada mensagem enviada ao servidor. As velocidades de comunicação utilizadas no experimento deveriam respeitar as velocidades do 2G, 3G, 4G e 5G, permitindo avaliar se as velocidades atuais da telefonia móvel viabilizam a solução e um teste de carga para medir a capacidade operacional da plataforma.

1.4 Contribuições

A computação veicular em nuvem é um novo campo de pesquisa com o potencial de mudar a vida das pessoas. Ele traz a eficiência, segurança e conforto para motoristas e passageiros (FALCHETTI; AZURDIA-MEZA; CESPEDES, 2015). Sendo a principal contribuição deste trabalho, a construção de uma arquitetura para criação de redes veiculares em nuvem, como também, a análise dos dados coletados a partir dos experimentos realizado sobre a plataforma desenvolvida seguindo a especificação da arquitetura. Nesse sentido, este estudo mostra que também é válido o uso de VCC no auxílio das soluções para os principais desafios relacionados à VANETs.

Outra contribuição importante foi poder instalar e analisar aspectos de segurança nas redes veiculares, podendo apenas assinar a mensagem ou criptografá-la, seguindo com análise de viabilidade de ambos os casos. Conclui-se que, o desempenho e a eficiência da arquitetura atendem aos requisitos relacionados à VANETs. E assim, demonstrar que o tempo de conexão e processamento em uma rede VCC, em ambiente de simulação, atendem às necessidades de vários tipos de aplicações relacionadas a redes veiculares.

Dentro desse contexto, este trabalho apresenta como principal contribuição, um modelo de arquitetura de software flexível e extensível, que utiliza rede veicular ad-hoc (VANET) e Computação em Nuvem, com a finalidade de proporcionar uma alternativa para os principais desafios relacionados a VANETs e permitir construir ambientes de gerenciamento de rede veicular como serviço (VaaS).

1.5 Organização do Trabalho

Os demais capítulos da dissertação estão organizados da seguinte forma: O capítulo 2 aborda de maneira geral, conceitos de internet das coisas e mostrando a sua importância para soluções de ITS. No capítulo 3, mostram-se definições de sistemas distribuídos bem como os tipos e os protocolos de comunicação. É apresentado no capitulo 4, as redes veiculares, seus os principais desafios, seus protocolos e suas aplicações. No capítulo 5 é apresentada a arquitetura I9VANET, seus módulos e detalhes da tecnologia utilizada na construção. Já o capítulo 6, apresenta o processo de avaliação realizado na plataforma I9VANET, bem como, a análise dos resultados. O capítulo 7 apresenta as conclusões,

destacando as principais contribuições, o resultado da análise dos dados e sugestão para os possíveis trabalhos futuros.

2 Rede de Sensores sem Fio

As Redes de Sensores sem Fio ou WSN (*Wireless Sensor Network*) estão emergindo como uma nova ferramenta para aplicações importantes em diversos campos como vigilância militar, monitoramento de ambiente, coleta de informações em ambiente hostil, vigilância de edifícios, entre outros.

2.1 Definição

Uma rede de sensores sem fio é uma tecnologia específica que ajuda na criação de aplicações com foco em cidades inteligentes. O seu objetivo consiste em criar uma rede que contenha muitos nós de sensores "inteligentes" que possam detectar múltiplos parâmetros de interesse para uma melhor gestão da cidade.

Segundo Losilla et al. (2011), para realizar tarefas de detecção, a maioria dos Sistemas de Transporte Inteligentes atuais contam com sensores caros, que oferecem apenas funcionalidade limitada. Uma tendência mais recente, consiste em utilizar WSN para tal finalidade, o que reduz o investimento necessário e permite o desenvolvimento de novas aplicações colaborativas e inteligentes, que contribuem ainda mais para melhorar tanto a segurança de condução como a eficiência do tráfego (LOSILLA et al., 2011).

Da mesma forma que em redes de sensores sem fio, os nós de uma rede VANET necessitam trocar informações com outros nós próximos, o que tornam as técnicas de WSN aplicáveis às redes veiculares. Entretanto, há o desafio da alta mobilidade dos nós em uma rede VANET o que torna o processo de comunicação complexo.

2.2 Arquitetura de Rede e Topologia

Uma aplicação ITS baseada em WSN distribuída, realiza quatro tarefas principais distintas: (i) aquisição de informação, (ii) distribuição de dados, (iii) processamento de dados para planejar as ações necessárias e, finalmente, (iv) execução das ações apropriadas (LOSILLA et al., 2011). Uma vez que estas tarefas possam ser realizadas de forma independente, pode-se considerar que definem correspondentemente quatro subsistemas diferenciados que estão presentes nos sistemas ITS, como mostrado na Figura 3. Nomeada-

mente o subsistema «Sensores», o subsistema «Distribuição», o subsistema «Tomada de decisões» e o subsistema «Execução».

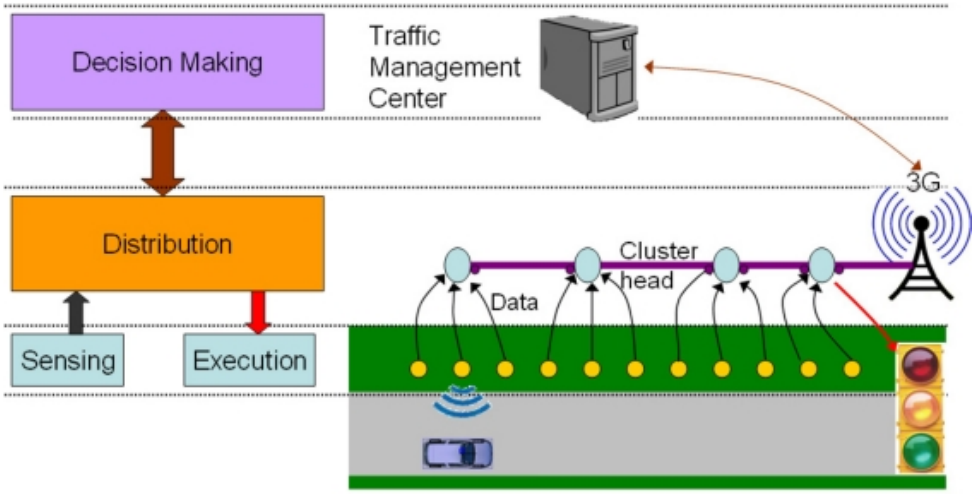


Figura 3 – Arquitetura de referência para aplicações ITS baseadas em WSN. (LOSILLA et al., 2011)

.

2.2.1 Subsistema de Sensores

Este subsistema é composto pela aquisição das informações relevantes, principalmente em relação ao tráfego e estados da rodovia. Numa aplicação ITS baseada em WSN, não necessariamente todos os dispositivos utilizam a tecnologia WSN, permitindo uma distribuição de tarefas entre dispositivos que utilizam tecnologias diferentes.

A implantação do subsistema de sensores consiste na construção de um ou mais WSNs em toda a área de observação (estradas ou parques de estacionamento), que detectam veículos através dos seus sensores e, opcionalmente, comunicam-se por meio da tecnologia sem fio.

Os nós WSN são divididos em grupos seguindo um esquema semelhante, e a implantação consiste em uma composição, tipicamente homogênea, desses grupos de nós. Portanto, eles podem ser considerados como os blocos de construção do subsistema de detecção. Deve-se notar que, além disso, os nós que formam esses blocos podem propagar informações dentro deles, mas isso não deve ser confundido com o subsistema de distribuição. A propagação de dados no subsistema de sensores é restrita a áreas locais, visando extrair informações do bloco / grupo (para um nó coletor) ou ativar o processamento colaborativo

com nós próximos. Se for preciso a disseminação de dados através de áreas maiores, será necessário o uso do subsistema de distribuição.

2.2.2 Subsistema de Distribuição

O subsistema de distribuição é responsável pela troca de informações entre os diferentes subsistemas de uma aplicação ITS. Numa arquitetura em camadas, tal como a da Figura 3, ela é colocada numa posição central, recebendo pedidos de comunicação de todos os outros subsistemas e servindo-os em conformidade. Também é responsável pela transmissão dos dados detectados para o subsistema «tomada de decisões» e, por outro lado, pela transmissão de comandos do subsistema «tomada de decisões» para os subsistemas «detecção e execução» (SUNG; YOO; KIM, 2007). Da mesma forma, interconecta os diferentes grupos de sensores de uma rede. Isso resulta em uma rede escalável criada pela composição desses grupos. A este respeito, o subsistema de distribuição pode igualmente interligar grupos fisicamente isolados de nós ou veículos, permitindo assim a interconexão de ilhas WSN desdobradas em diferentes partes ao longo da estrada (WEINGÄRTNER; KARGL, 2007) ou de agrupamentos de veículos que de outra maneira formariam VANETs não ligados (BARBA; AGUIRRE; IGARTUA, 2010) .

O subsistema de distribuição consome uma quantidade significativa de energia devido às exigências impostas aos seus dispositivos. Eles são encarregados de encaminhar cada evento relatado por cada nó do subsistema de sensores, que pode ocorrer em uma alta taxa de frequência. Além disso, deve ser feito sob as restrições de tempo definidas pela aplicação. Isto requer nós muito ativos para garantir a entrega de informações em tempo hábil, comprometendo a economia de energia. Portanto, os dispositivos usados neste subsistema precisam de um orçamento de energia mais generoso do que os nós de sensores, tornando necessárias fontes de energia adicionais (LOSILLA et al., 2011).

Existem diferentes formas de implementar o subsistema de distribuição. Uma delas consiste em empregar redes veiculares para disseminar informações. Dispositivos utilizados nos veículos não têm restrições de energia, uma vez que podem ser alimentados por instalações do próprio veículo. Além disso, a mobilidade de veículos, que é um fator limitante para outros tipos de aplicações, ajuda a espalhar dados em veículos e nós estáticos ao longo da rodovia, e alivia os nós estáticos das tarefas de distribuição. O esquema mais simples envolve a utilização de transmissão direta entre veículos, isto é,

uma rede de um salto que permite a partilha de dados de um veículo de origem para cada veículo que se aproxima (LI; LIU; TANG, 2007) (MIURA; ZHAN; KURODA, 2006). A segunda alternativa baseia-se na utilização de uma rede de distribuição VANET multi-salto (QIN et al., 2010). Esta opção requer mais recursos do sistema, mas também facilita a disseminação mais rápida de dados e escalas muito melhor à medida que o número de veículos tecnologicamente equipados cresce.

De acordo com Losilla et al. (2011), um subsistema de distribuição baseado em redes celulares é uma opção atraente, não apenas por sua alta disponibilidade, mas também pelo custo de implantação. Mesmo sabendo que *gateways* são dispositivos caros que requerem interfaces de rede celular, apenas alguns deles são necessários para serem colocados perto de estações de base celulares (BS) e alguns nós mais baratos que conectam os nós de detecção com os *gateways* devem ser implantados. Por outro lado, os custos de exploração também devem ser levados em conta, uma vez que os operadores de celulares cobram a utilização da BS. Isso pode levar a uma outra escolha, quando não está planejado pagar o uso de BS ou não estão disponíveis, que consiste em usar IMS (Internet Multimedia Subsystems) (BIRK; OSIPOV; ELIASSON, 2009) que fornecem acesso ao Sistema de Distribuição através de diferentes tecnologias (2G, 2.5G, 3G, 4G, 5G, WLAN) independentemente do operador.

2.2.3 Subsistema de Tomada de Decisão

Subsistema de tomada de decisão é também conhecido por DMS (*Decision Making subsystem*), é responsável pelo planejamento das ações necessárias para alcançar os objetivos do sistema. As tarefas atribuídas a este subsistema podem ser divididas em três grupos diferentes. O primeiro deles inclui tarefas destinadas ao armazenamento e ao pré-processamento de dados. Trata-se da enorme quantidade de dados que chega ao subsistema, filtrando e armazenando apenas informações relevantes e subseqüentemente, acessando-as. O segundo grupo, trata as informações de tráfego de diferentes fontes e as processa de acordo com o objetivo da aplicação. Finalmente, o terceiro grupo de tarefas é responsável por endereçar comandos de controle, bem como para gerenciamento da rede.

O DMS pode ser executado em diferentes níveis. Em um nível superior, ele pode ser implementado em centrais de monitoramento de tráfego. Isto implica que todos os dados recolhidos pelo subsistema de sensores são enviados, através do subsistema de distribuição, para o DMS, que deve suportar um fluxo de dados assimétrico. A principal vantagem desta

abordagem é a possibilidade de realizar cálculos complexos sobre uma grande quantidade de informação. Inversamente, se apenas o processamento simples for aplicado, o DMS pode ser distribuído entre os nós . Isso permite executar algoritmos colaborativos simples entre nós vizinhos que permitem a execução de aplicações de segurança de tráfego em tempo real. Finalmente, uma outra solução é a utilização de dispositivos inteligentes (smartphones, etc.) em veículos, que podem receber dados brutos de redes rodoviárias e usá-los, por exemplo, para planejar rotas (LOSILLA et al., 2011).

2.2.4 Subsistema de Execução

Este subsistema é responsável por executar ações necessárias que promovam mudanças no fluxo de tráfego de acordo com o objetivo da aplicação ITS. É composto principalmente por dispositivos que fornecem estímulos visuais e acústicos aos condutores, embora outros, destinados à automação de veículos, também pertençam a este subsistema. Os semáforos ou os sinais de mensagem variável instalados ao longo das estradas são opções atrativas que proporcionam um controle rigoroso e adaptabilidade a diferentes situações, respectivamente. Eles oferecem a vantagem de serem infra-estruturas rodoviárias amplamente adotadas, adequadas para reutilização em aplicações para ITS, o que ajudaria a reduzir os custos de implantação (LOSILLA et al., 2011).

O emprego de sistemas no veículo, oferece, por um lado, a possibilidade de apresentar informação personalizada para cada veículo e, por outro lado, a possibilidade de utilizar sinais acústicos e mensagens que diminuem as distrações durante a condução. Além disso, as informações provenientes dos sistemas rodoviários podem ser integradas nos sistemas de informações no veículo, o IVI (*In-Vehicle Infotainment*), por exemplo, para a sua fusão com mapas digitais ou outros serviços de informação (horários de transporte, previsões meteorológicas, etc.). No entanto, há um interesse crescente dos fabricantes de veículos em incorporar sistemas IVI em seus produtos como um diferencial competitivo. A este respeito, novos modelos de automóveis de empresas importantes estão equipados com sistemas como o iDrive da BMW (NIEDERMAIER et al., 2009), o Audi MMI , a Ford SYNC ou GENIVI Apollo da aliança GENIVI apud Losilla et al. (2011), que visam facilitar o desenvolvimento das aplicações IVI.

2.3 Considerações Finais do Capítulo

Neste capítulo foi abordada a importância das redes de sensores sem fio em soluções voltadas para cidades inteligentes e principalmente a arquitetura de referência aplicada à ITS, sendo dividido em quatro tarefas principais, são elas: aquisição de informação, distribuição das informações, processamento de dados e execução das ações apropriadas. Redes veiculares também são consideradas uma rede de sensores sem fio, visto que pode fazer uso de sensores do veículo tais como, velocidade, rotação do motor, direção, posicionamento geográfico entre outros. Tais informações podem ser transmitidas para outros veículos de forma que seja montada uma malha de informações ajudando os motoristas na tomada de decisão.

3 Sistemas Distribuídos

Com o surgimento das redes locais (LAN), o homem tenta aproveitar melhor o poder computacional dividindo tarefas em vários computadores para realizar processamento paralelo, de modo a aumentar o poder computacional sem utilizar super computadores. Surgindo assim, o conceito de sistemas distribuídos (SD).

3.1 Definição

De acordo com Tanenbaum e Steen (2007), um sistema distribuído é um conjunto de computadores independentes entre si, que se apresenta a seus usuários como um sistema único e coerente. Já Coulouris et al. (2013), define sistemas distribuídos como sendo uma coleção de computadores autônomos interligados através de uma rede de computadores e equipados com software que permita o compartilhamento de hardware, software e dados.

De acordo com as definições, os sistemas distribuídos apresentam algumas características: concorência de recursos, falta de um relógio global e falhas de componentes independentes. Sendo o compartilhamento de recursos um dos principais motivos para se utilizar sistemas distribuídos. Contudo, a construção de sistemas distribuídos apresentam alguns desafios como: a heterogeneidade de seus componentes, ser um sistema aberto, segurança e escalabilidade, tolerância a falhas, a concorrência aos recursos e a transparêcia (COULOURIS et al., 2013).

A miniaturização dos dispositivos e a interligação em redes sem fio, tem ocasionado uma convergência de equipamentos cada vez menores com sistemas distribuídos. A portabilidade desses dispositivos, torna a computação móvel possível e fundamental para que a computação ubíqua ou pervasiva seja utilizada cada vez mais pelo usuário e de maneira transparente e natural.

3.2 Características e Desafios

Os sistemas distribuídos são encontrados em toda parte, contudo, há desafios que todo sistema SD precisa se preocupar tais como: heterogeneidade, sistemas abertos, segurança, escalabilidade e tolerância a falhas.

3.2.1 Heterogeneidade

Segundo Coulouris et al. (2013), os aspectos que devem ser levados em consideração pela heterogeneidade, estão relacionados à rede, ao hawdare do computador, sistemas operacionais, linguagens de programação e implementações de diferentes desenvolvedores. Em uma rede veicular, tais aspectos estão bem ligados visto que há vários fabricantes de veículos e cada um possui tecnologia própria.

3.2.2 Sistemas Abertos

Um sistema é considerado aberto quando é possível estendê-lo de várias maneiras. O fato de um SD ser ou não um sistema aberto é determinado pelo grau com que os novos serviços podem ser inseridos para ser utilizado por uma variedade de aplicações clientes (COULOURIS et al., 2013). Sendo assim, é necessário publicar as principais interfaces dos componentes de software. Entretanto, a publicação do padrão de comunicação com o software é apenas o início para adicionar novos serviços a um sistema distribuído.

3.2.3 Segurança

A segurança atrelada a sistemas distribuídos possui três componentes que são eles: confidencialidade, integridade e disponibilidade. O primeiro item, deve proteger o acesso à informação de pessoas não autorizadas. Quanto à integridade, os sistemas distribuídos devem garantir a não adulteração dos dados e por último, a disponibilidade, o qual deve proteger contra interferência de acesso ao recurso.

3.2.4 Escalabilidade

Em se tratando de escalabilidade, há dois tipos: a vertical e a horizontal, sendo a primeira, a forma de escalonar um sistema centralizado, realizando um *upgrade* na memória, nos processadores e/ou nos discos, quando possível. Já no escalonamento horizontal, consiste em acoplar mais máquinas em um ambiente distribuído, não havendo limites.

Um dos grandes objetivos dos sistemas distribuídos, é permitir o aumento de capacidade à medida que cresce a demanda pelo recurso. E um dos principais desafios

é balancear os custos dos recursos físicos com a perda de desempenho, impedindo que os recursos de software se esgotem evitando gargalos na performance. Sendo assim, a escalabilidade é considerada um tema central em sistemas distribuídos.

3.2.5 Tolerância a Falhas

Os sistemas de computador falham, seja por motivo de hardware ou software, os programas podem produzir resultados incorretos. Em um ambiente com sistema distribuído, alguns componentes podem falhar enquanto outros continuam funcionando, sendo considerado algo complexo tratar corretamente cada problema que pode ocorrer. A tolerância a falha é fundamental para o bom funcionamento das aplicações de segurança crítica (GORENDER; MACÊDO, 2002) . Dentre as técnicas para tolerância a falhas temos:

- **Detecção de Falhas**: algumas falhas podem ser detectadas antes que mesmo que afete o sistema. Para isso, somas de verificação podem ser utilizadas para verificar a integridade da informação por exemplo, mas outras falhas não há como prevê, como, por exemplo, a parada de um servidor.
- **Mascaramento de Falhas:** algumas falhas podem ser mascaradas, por exemplo, a lentidão momentânea de um *link* pode ser resolvido retransmitindo novamente a mensagem, sem precisar que o usuário tome conhecimento.
- **Recuperação de Falhas**: envolve desenvolver softwares capazes de recuperar ou retroceder o estado dos dados para um momento consistente.
- **Redundância:** alguns serviços podem melhorar sua tolerancia a falhas com o uso de discos espelhados ou *links* de Internet duplicados, entre outras redundancias. Assim, caso algum destes recursos falhem, terá uma alternativa para continuar funcionando.

3.2.6 Transparência

A transparência tem como objetivo “esconder’, do usuário ou programador, detalhes da separação dos componentes em um sistema distribuído, de maneira que seja percebido como um todo ao invés de uma coleção de componentes independentes. Segundo (COULOURIS et al., 2013), há dois critérios de transparência que são considerados importantes,

a transparência de acesso e de localização, sendo também conhecida como transparência de rede. Sua presença ou falta, afeta substancialmente a utilização de recursos distribuídos.

3.3 Comunicação em Sistemas Distribuídos

As restrições temporais determinam a variação dos modelos de sistemas distribuídos existentes. Podendo ser síncronos ou assíncronos. O modelo síncrono deve ser utilizado quando se conhece os limites temporais, permitindo que haja uma estimativa com relação ao tempo de execução dos protocolos do sistema e funções da própria aplicação distribuída (LAMPORT; SHOSTAK; PEASE, 1982) (LYNCH, 1996) (CRISTIAN, 1991). A tolerância a falhas é mais eficiente com a comunicação síncrona, pois permite o uso de *timeout* mais facilmente. Entretanto, nos sistemas assíncronos (*time free*), não há restrições temporais, sendo impossível obter o consenso na presença de falhas (FISCHER; LYNCH; PATERSON, 1985) (MACÊDO, 2000), devido à dificuldade inerente desse tipo de sistema em diferenciar entre uma operação com falha ou extremamente lenta (GORENDER; MACÊDO, 2002).

Os modelos de comunicação existentes para sistemas distribuídos baseiam-se ou nas características síncronas de redes locais ou em ambientes de melhor esforço como os da Internet (GORENDER; MACÊDO, 2002). Nenhum dos modelos consideram as arquiteturas IntServ e DiffServ para prover controle de QoS (Qualidade de serviço).

As arquiteturas IntServ e DiffSer foram propostas pela IETF (*Internet Engineering Task Force*) cujo objetivo é permirtir que sistemas de comunicação possam fornecer serviços com diferentes níveis de qualidade, levando em consideração aspectos como: fixação de limite para transferência de dados e diferentes prioridades com relação à possibilidade de perda de pacotes.

3.3.1 Comunicação Cliente-Servidor

A comunicação baseada no Cliente-Servidor foi projetada para suportar a troca de mensagens em interações tipicamente síncronas, pois o processo no cliente bloqueia a execução até a chegada da resposta. Também é considerada confiável, já que a resposta é a confirmação da chegada da requisição no servidor (COULOURIS et al., 2013). E o protocolo é baseado em três primitivas de comunicação: *doOperation*, *getRequest* e

sendReply. A maioria dos sistemas RMI (*Remote Mehthod Invocation*) e RPC (*Remote Procedure Call*).

O método *doOperation* é utilizado para invocar processos remotos. Seus parâmetros especificam o objeto remoto e a operação a ser executada. A segunda primitiva, o *getRequest*, é executado por um processo no servidor, para ler as requisições do serviço. Após execução do processo, pelo servidor, será executado o *sendReply* para enviar a mensagem de resposta para o cliente, fazendo com que o método *doOperation* seja desbloqueado liberando a aplicação cliente.

3.3.2 Comunicação em Grupo

Para fornecer tolerância a falhas à disponibilidade, é necessário o uso de difusão seletiva (*multicast*) em situações que precise realizar uma comunicação de um processo com um grupo de processos. Sendo que a troca de mensagens *multicast* fornecem uma infra-estrutura importante para construção de sistemas distribuídos com as seguintes características:

- Tolerância a falhas baseada em serviços replicados: as requisições do cliente são solicitadas para todos os membros do grupo onde cada um é responsável por executar a mesma operação. Mesmo se alguma requisição falhar, o cliente pode ser atendido.
- Localizar servidores na interligação em rede espontânea: mensagens *multicast* podem ser utilizadas para localizar os serviços de descobertas disponíveis, podendo ser usadas por servidores e clientes.
- Melhor desempenho através da replicação de dados: em alguns casos, os dados são replicados nas máquinas dos clientes, e quando houver mudança, a atualização é feita por *multicast*.
- Propagação de notificações de um evento: pode ser utilizado para notificar um grupo, de servidores ou clientes, quando ocorrer mudanças em um determinado evento.

3.3.3 Protocolos de Comunicação

Há diversos estilos de empacotamento para troca de mensagens em sistemas distribuídos. O CORBA, opta por empacotar os dados para uso pelos destinatários que já

conhecem seus tipos anteriormente. Ao contrário do Java, que serializa os dados incluindo informações completas sobre os tipos de seu conteúdo, possibilitando que o destinatário possa reconstruí-los. A linguagem XML segue o modelo semelhante à do Java, fornecendo os tipos de seus conteúdos.

A Figura 4 mostra os mecanismos usados para criação de clientes e servidores para *WebService*, CORBA e Java-RMI. Quando são utilizados stubs gerados automaticamente no lado do cliente para Web Services, os processos de desenvolvimento e a complexidade do código para o cliente e do servidor, são praticamente os mesmos para soluções Web Service, Java-RMI e CORBA. Os fatores que determinarão a escolha serão: a interoperabilidade e o desempenho (GRAY, 2004). Sendo que os Serviços Web permite uma maior integração entre plataformas distintas e dispositivos, enquanto que o desempenho favorecerá ao CORBA e Java-RMI.

Gray (2004) realizou um experimento que enviava uma única solicitação de dados e encerrava. Em qualquer dos três casos, *Web Services*, CORBA e Java-RMI, as tecnologias de serviços da Web funcionam bem em comparação com as outras duas, como mostra a Tabela 2.

Tabela 2 – Análise de custos de solicitação de uma única requisição utilizando várias tecnologias(GRAY, 2004)

Tecnologia	Latência Total	Total Pacotes	Total Dados Transferidos (B)
WS	0,11s	16	3338
CORBA	0,48s	8	1111
Java-RMI	0,32s	48	7670

3.3.3.1 WebSocket

A tecnologia *websocket* foi introduzida pelo HTML5 (*HyperText Markup Language* quinta geração) através da RFC 6455: *The WebSocket Protocol*, o qual permite utilizar um canal de comunicação bidirecional entre um cliente e um servidor remoto, de forma persistente e dedicado usando um único *socket* TCP. Isto é, a comunicação pode ocorrer nos dois sentidos simultâneo e assíncrono, e a conexão permanece aberta até que em uma das partes realize o fechamento (MELNIKOV; FETTE, 2011).

O *WebSocket* é um protocolo que é definido na camada de aplicação e pode ser utilizado para superar restrições existentes na camada de transporte. Utiliza o modelo de

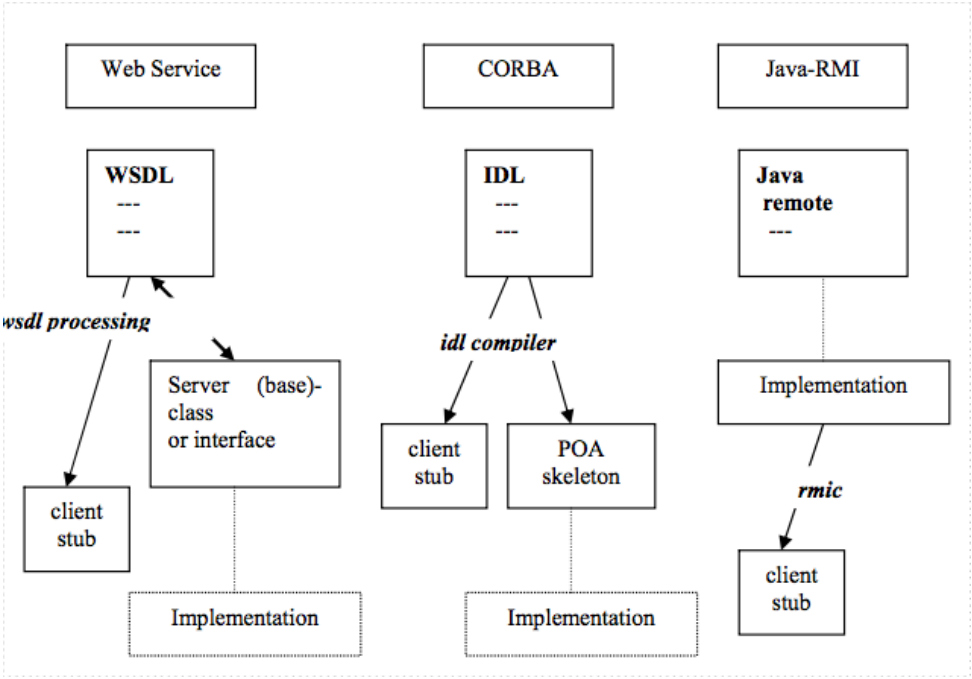


Figura 4 – Geração de componentes de cliente e servidor, a partir da interface para Web Services, CORBA e Java-RMI (GRAY, 2004)

.

mensagens cliente-servidor como base e suporta tanto dados binários quanto texto simples (THEMUDO, 2014).

As fases de conexão *WebSocet* estão representadas na Figura 5, que vai da criação do canal TCP, depois, troca de mensagens e por último, o fechamento. Na fase de **Ligação *WebSocket*** é realizada a troca de mensagens indefinidamente.

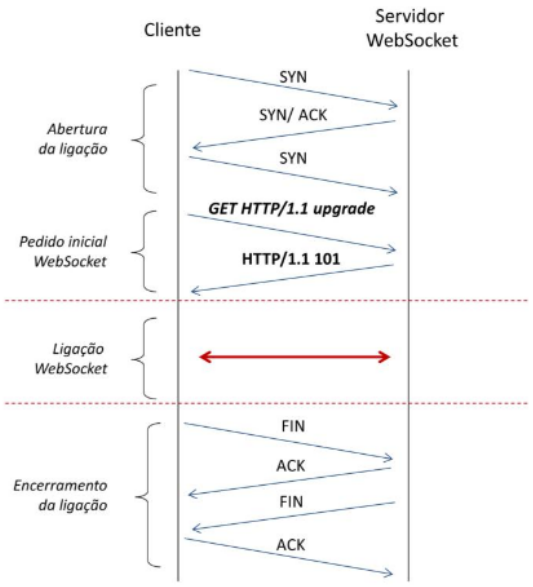


Figura 5 – Etapas de uma conexão *WebSocket* (THEMUDO, 2014)

.

As implementações do *WebSocket* são baseadas no modelo de eventos, isso quer dizer que as aplicações apenas têm que esperar o evento ser chamado. Há quatro tipos de eventos nas implementações do *WebSocket*:

- **Open**: indica que a conexão foi realizada com o servidor e que a comunicação pode começar.
- **Message**: é acionado sempre que recebe uma nova mensagem.
- **Error**: sempre que ocorre uma falha, normalmente a conexão é finalizada após esse evento.
- **Close**: informa que a conexão foi finalizada.

Sobre a segurança, o *WebSocket* apresenta as mesmas vulnerabilidades em relação a outras aplicações web com o protocolo HTTP (Hypertext Transfer Protocol). Sendo assim, é interessante que a comunicação do protocolo WS seja criptografada com o SSL/TLS. O protocolo SSL/TLS tem como principal objetivo oferecer privacidade e integridade dos dados no momento da comunicação entre dois terminais (THEMUDO, 2014).

3.4 Considerações Finais do Capítulo

Neste capítulo, foram mencionados as caracteísticas e desafios relacionados à construção de sistemas distribuídos, os tipos de comunicação cliente-servidor e em grupo, como também as tecnologias para se criar sistemas distribuídos tais como: CORBA, JAVA-RMI, WebServices e WebSocket. Em um ambiente VANET, a heterogeneidade está presente, visto que é uma das grandes preocupações no momento do desenvolvimento de uma solução. E em uma rede veicular com gerenciamento em nuvem, a comunicação entre os veículos e os servidores devem considerar o menor tempo possível, sendo necessário utilizar tecnologias de comunicação que permitam um melhor desempenho tal qual WebSocket.

4 Redes Veiculares

4.1 Definição

As VANETs que usam veículos como nós móveis são uma subclasse de rede móveis ad hoc chamadas de MANETs. Elas fornecem comunicação entre os veículos próximos e entre veículos e equipamentos à beira da rodovia. Os nós numa rede VANET são muito mais dinâmicos, pois os veículos possuem velocidade e direção variável. A alta mobilidade dos nós conduz a uma topologia de rede dinâmica caracterizada pela constante perda de comunicação (BUBENIKOVA; DURECH; FRANEKOVA, 2014) (JAKUBIAK; KOUCHERYAVY, 2008) e podem ser categorizadas segundo o tipo de ligações existentes. Dessa maneira, podemos considerar três arquiteturas de redes veiculares (LUÍS, 2009), como mostra a Figura 6:

- **Arquitetura WLAN ou celular:** baseada na utilização de antenas fixas, colocadas ao longo da rodovia, funcionando como pontos de acesso à rede. Não existe qualquer ligação direta entre os veículos. Em um cenário de auto-estrada, a implantação dos equipamentos à beira da rodovia de maneira suficiente para permitir a cobertura necessária pode tornar uma solução bastante dispendiosa.
- **Arquitetura ad-hoc:** é considerada uma arquitetura ad-hoc quando não há quaisquer uso de infra-estruturas para realizar a comunicação, sendo as ligações é feita diretamente entre os nós envolvidos. Fatores como velocidade ou densidade dos nós podem pôr em *check* o desempenho deste tipo de rede.
- **Arquitetura híbrida:** tem objetivo de retificar as falhas existentes nas duas arquiteturas anteriores, utilizando concomitantemente às arquiteturas ad-hoc e WLAN

As comunicações em VANETs são categorizadas em 4 tipos:

- **Em veículos:** pode ser utilizado para detectar a fadiga e/ou sonolência de um motorista que representa risco na segurança;
- **Entre veículos**: a comunicação V2V (veículo para veiculo) pode fornecer uma plataforma de intercâmbio de dados para compartilhamento de informações de advertência de modo a alertar o motorista;

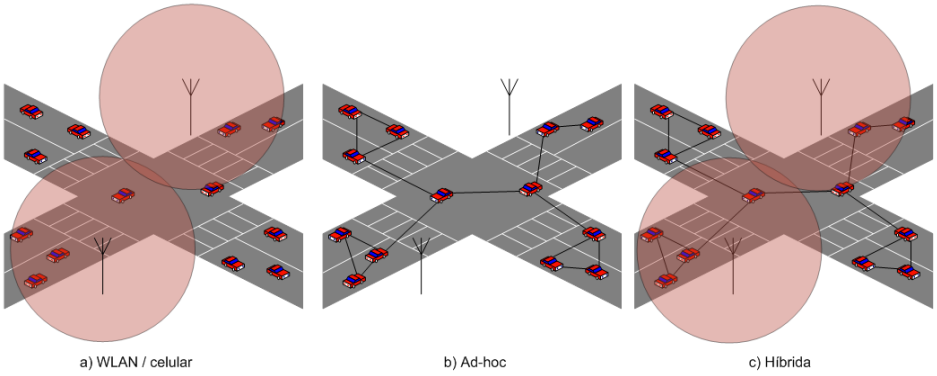


Figura 6 – Arquiteturas de redes veiculares(LUÍS, 2009)

.

- **Entre veículos e rodovia (V2I)**: permite atualização em tempo real do tráfego e fornece detecção e monitoramento do ambiente;
- **Entre veículo e nuvem**: os veículos podem comunicar-se através da banda larga sem fio tais como 3G, 4G ou WIMAX podendo enviar dados para uma central, o que permitiria um controle mais abrangente do tráfego e assistência ao motorista

As aplicações VANETs podem ser divididas em 3 classes: segurança, entretenimento e assistência ao motorista. O principal desafio relacionado à segurança, está ligado à velocidade de alerta ao condutor, para que possa ter tempo hábil de reação. Nas aplicações de entretenimento, destacam-se os sistemas de compartilhamento de conteúdo e jogos. Já sobre a assistência ao motorista, são aquelas que auxiliam o condutor como por exemplo, através do uso de informações sobre as condições do trânsito (SOUZA RONIEL DE; SOARES, 2014).

4.2 Características e Desafios

As redes veiculares ad-hoc caracterizam-se por serem redes sem fio em que todos os nós são efetivamente ativos na comunicação. Entretanto, as redes veiculares apresentam como principal caracteristica a sua alta mobilidade, e é justamente por isso que surge uma série de desafios a serem tratados antes mesmo de uma solução ser posta em prática. Dentre os maiores desafios encontram-se:

- **meio físico**: interferências devido a prédios, árvores e outros obstáculos;
- **alta mobilidade:** dificulta a troca de informações mais completas;

- **topologia**:VANET possui uma característica dinâmica, devido à velocidade que os veículos se movimentam;
- **baixa densidade**: quando a densidade de tráfego é baixa e os veículos estão distantes uns dos outros;
- **alta densidade**: muitos veículos em uma pequena área fazem com que a quantidade de mensagens trocadas torne-se um problema;
- **segurança**: como as VANETs suportam aplicações de emergência em tempo real e lidam com informações críticas de segurança no trânsito, estas devem satisfazer os seguintes requisitos de segurança: confidencialidade, integridade, disponibilidade, autenticidade e não repudiação para prover segurança na comunicação dos dados (SAMARA; AL-SALIHY; SURES, 2010) (MATOS et al., 2013).

4.3 Segurança

Dito por Wangham et al. (2014), a segurança em redes veiculares é um fator crucial que precisa ser levado em consideração, pois a falta desta pode afetar a vida das pessoas. Como quaisquer redes de computadores sem fio e redes *ad-hoc*, estas estão sensíveis a ataques, tais como: negação de serviço e alteração de mensagens (RAYA; PAPADIMITRATOS; HUBAUX, 2006).

Para construir uma arquitetura de segurança robusta para redes veiculares, é necessário estudar as peculiaridades dos ataques que podem ocorrer. Do mesmo modo que as redes clássicas, as VANETs são vulneráveis a muitos ataques. Alguns destes, são encontrados e soluções são concebidas, considerando que um dia estes ataques podem ser lançados sobre a rede (ENGOULOU et al., 2014). De acordo com Wangham et al. (2014), os principais ataques analisados na literatura são: ataques contra a disponibilidade; ataques contra a autenticidade e a identificação; ataques contra a integridade e confiança dos dados; ataques contra a confidencialidade; entre outros.

4.3.1 Ataques contra a Disponibilidade

- **Negação de serviço (DoS)**: este ataque tem como objetivo evitar que veículos autênticos acessem aos recursos da rede não permitindo a troca de informação. Um ataque pode proceder de 3 maneiras: sobrecarregando um nó específico da rede com

informações deixando-o extremamente ocupado; atacando o canal de comunicação gerando altas frequências adicionando ruído e impossibiltando a troca de informações entre os veículos; o não repasse de pacotes para outros veículos da rede, fazendo com que a informação não seja propagada para os outros nós.

- **Negação de serviço distribuída (DDoS)**: possui o mesmo objetivo do DoS, ou seja, indisponibilidade de recursos, porém o ataque parte de diferentes localizações e em horários distintos como mostra a Figura 7, onde veículos (B, C e D) enviam uma grande quantidade de pacotes contra uma unidade RSU, ocasionando sua indisponibilidade.

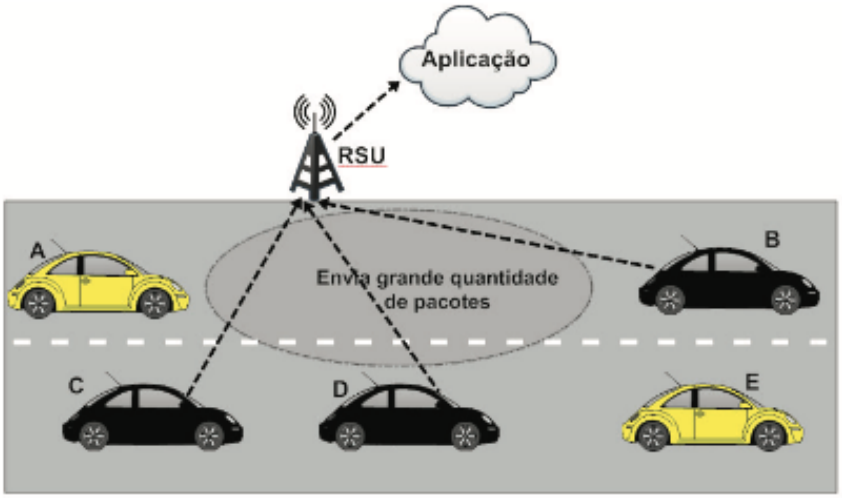


Figura 7 – Ataque DDOS (WANGHAM et al., 2014).

- **Supressão de mensagem**: o atacante recebe e não repassa pacotes da rede com objetivo de impedir que veículos seguintes possam saber de alguma ocorrência, por exemplo, um aviso de congestionamento ou acidentes (TANGADE; MANVI, 2013).
- **Buraco Negro (*black hole*)**: é uma área onde o tráfego de rede é redirecionado, contudo, ou não há veículos neste local ou nós maliciosos que estão nesta área se recusam a participar, fazendo com que os pacotes da rede não se propaguem. A Figura 8 ilustra o ataque onde veículos se recusam a transmitir a mensagem recebida pelo veículo C.
- **Jamming**: é um ataque de negação de serviço através do meio físico, onde o atacante transmite um sinal para pertubar o canal de comunicação, o que reduz a relação sinal ruído SNR (Signal to Noise Ratio) para o receptor. Segundo Avelar et al. (2015), o impacto com esse tipo de ataque é devastador.

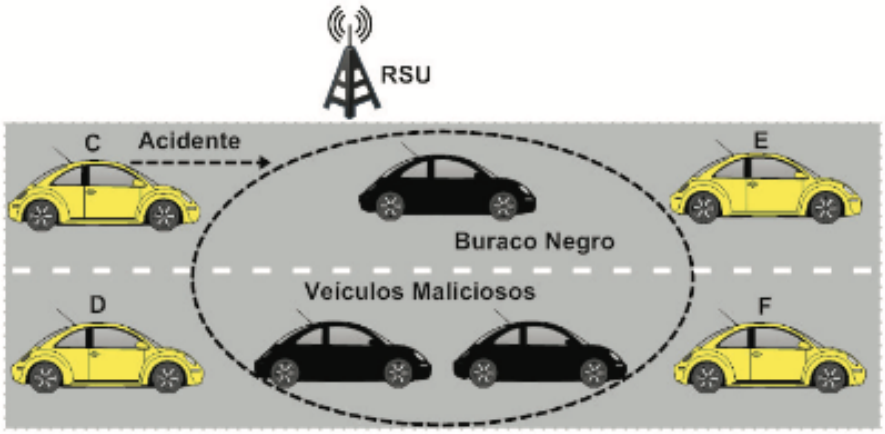


Figura 8 – Ataque Buraco Negro (WANGHAM et al., 2014).

4.3.2 Ataques contra a Autenticidade e a Identificação

- **Falsificação de endereço (*address spoofing*)**: o atacante cria um endereço falso de origem, fazendo com que o nó atacado confie no remetente achando que o mesmo possui permissão para conectar-se à rede (AL-KAHTANI, 2012).
- **Mascaramento**: ocorre quando um veículo malicioso falsifica sua identidade para se passar por outro, com intuito de ter acesso a recursos restritos. Por exemplo, se passar por uma ambulância e ter vantagem no trânsito.
- **Replicação do certificado ou chave**: consiste em utilizar certificados ou chaves duplicados, que são usados como prova de identificação com objetivo de criar ambiguidade dificultando assim a identificação de um veículo pelas autoridades (MEJRI; BEN-OTHMAN; HAMDI, 2014).
- **Sybil**: é gerado múltiplas identidades por um atacante para simular vários veículos e cada nó transmite mensagens com múltiplas identidades e assim sucessivamente. Desta maneira, um único veículo pode parecer centenas fazendo com que veículos reais mudem a rota achando que ali há um congestionamento (TANGADE; MANVI, 2013).

4.3.3 Ataques contra a Integridade e Confiança dos Dados

- **Falsificação nos dados de GPS (*GPS spoofing*)**: o atacante utiliza um simulador de satélite GPS para gerar sinais mais fortes que o sinal originado de um satélite

real, de maneira a enganar os sensores de GPS, introduzindo uma localização falsa (RAWAT; SHARMA; SUSHIL, 2012).

- **Ilusão (ataque contra os sensores do veículo)**: Isaac, Zeadally e Camara (2010) e Al-Kahtani (2012) disseram que esse é um novo tipo de ameaça em aplicações VANETs, onde o atacante interfere intencionalmente nos sensores do seu próprio veículo gerando valores errados, com objetivo de criar mensagens de aviso de tráfego incorretas na rede. Assim, é criado uma condição de ilusão em VANET. Os métodos tradicionais de autenticação e integridade utilizados em redes sem fio são inadequados contra este tipo de ataque.
- **Injeção de informação falsa (*bogus information*)**: neste tipo de ataque o atacante pode ser um usuário real ou um intruso que transmite informações falsas na rede para obter vantagens ou afetar decisão de outros veículos. Pode ser chamado também de ataque social, onde o atacante procura confundir e distrair a vítima com envio de mensagens antiéticas, para o motorista ficar confuso e distraído, podendo causar um acidente, como mostra a Figura 9.

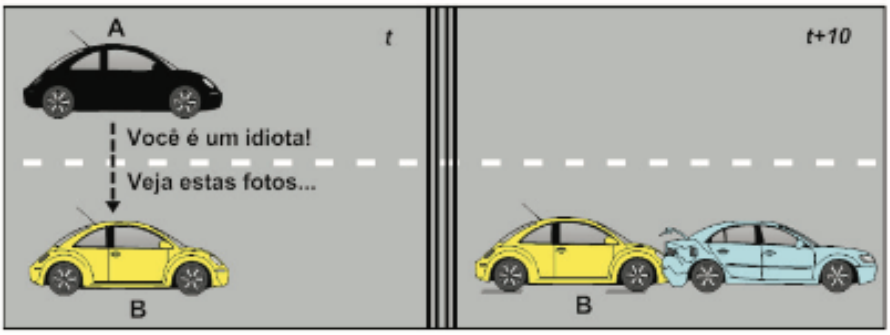


Figura 9 – Ataque social (WANGHAM et al., 2014).

- **Modificação de mensagem (*man in the middle*)**: de acordo com Mejri, Ben-Othman e Hamdi (2014), este ataca a autenticidade do remetente e a integridade das mensagens onde o veículo atacante fica inserido entre dois veículos reais que se comunicam. Então, o atacante faz o intermédio entre a comunicação interceptando as mensagens enquanto estes, acreditam estar se comunicando diretamente. A Figura 10 mostra o veículo malicioso M escutando a comunicação entre os veículos A e B, modifica um alerta recebido e propaga uma informação falsa, para os veíulos B e C)

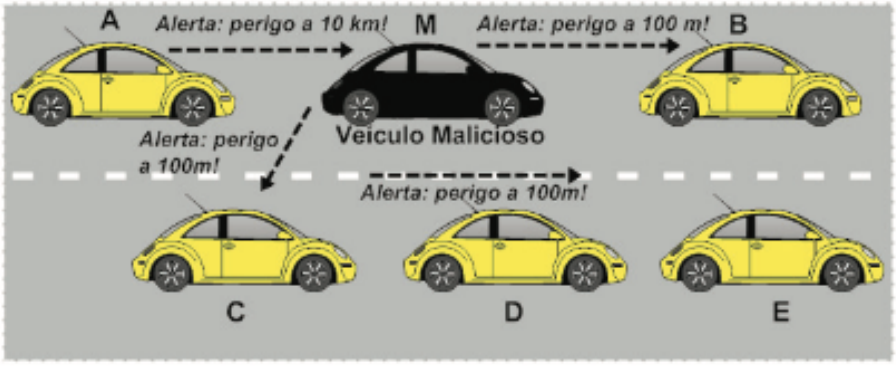


Figura 10 – Ataque demodificação de mensagem (WANGHAM et al., 2014).

4.3.4 Ataques contra a Confidencialidade

- **Análise de tráfego**: segundo Isaac, Zeadally e Camara (2010), esse é um ataque passivo utilizado em redes *ad-hoc*, onde o atacante que tenha acesso à rede pode interceptar o tráfego de pacotes, coletando dados que estejam sendo transmitidos sem o uso de criptografia.
- **Força bruta**: tem como foco capturar mensagens trocadas ou ainda, o processo de identificação e autenticação. Esse ataque não é trivial já que o tempo de comunicação entre os veículos é relativamente curto e esse tipo de ataque consome muito tempo (PATHRE, 2013).
- **Revelação de identidade**: normalmente, o motorista é o dono do veículo, então, o atacante pode obter a identidade do proprietário, violando sua privacidade (TANGADE; MANVI, 2013) e utilizá-lo para compor informações para um ataque social.

4.3.5 Outros Ataques

- **Ataques contra não repúdio (*accountability*)**: consiste em tomar medidas para permitir ao atacante, negar a realização de uma ou mais ação (WANGHAM et al., 2014). Apesar disso, dito por Mejri, Ben-Othman e Hamdi (2014), não foi encontrando na literatura um documento afirmando a possibilidade desse ataque.
- **Ataques contra a privacidade**: estes tipos de ataques violam a privacidade dos usuários e condutores em redes veiculares. Segundo Wangham et al. (2014), estudos na literatura classificam esses ataques como uma categoria separada para VANETs e citam dois exemplos: rastreamento de veículo durante sua viagem e engenharia

social, verificando se o veículos encontra-se em deslocamento ou parado (MEJRI; BEN-OTHMAN; HAMDI, 2014).

- **Conluio**: é o ataque que se utiliza de vários nós da rede com um objetivo comum, de realizar o vandalismo ou terrorismo, tendo como resultado a indisponibilidade da rede ou de algum serviço ou denegrir a reputação de confiança de um veículo (ZHANG, 2011).

As aplicações suportadas por VANETs, com foco em segurança, permitem a tomada de decisão por um condutor com base em informações enviadas por outros veículos. Entretanto, se um veículo malicioso cria ou altera mensagens, este pode colocar o motorista em situação de risco (WASEF et al., 2010). Dessa forma, a autenticidade das mensagens é fundamental para segurança das redes veiculares e dependendo da aplicabilidade, é necessário ainda fazer uso da criptografia (WANGHAM et al., 2014). Pensando nisso, vários autores (RAYA; PAPADIMITRATOS; HUBAUX, 2006) (MEJRI; BEN-OTHMAN; HAMDI, 2014) escolheram a assinatura digital como solução para autenticação das mensagens contidas nas redes veiculares.

O método mais simples e eficiente para assinatura digital, é atribuir, a cada veículo, um par chaves possibilitando assinar digitalmente as mensagens. Assim, é necessário utilizar uma autoridade certificadora confiável implicando no uso de uma Infraestrutura de Chaves Públicas (ICP) veicular (WASEF et al., 2010).

4.4 Algoritmos de Roteamento

Os algoritmos de roteamento em redes veiculares são um dos desafios reconhecidos na literatura, já que não existem rotas pré-definidas e nem estimativa da quantidade de nós de uma rede e o fator primordial é a alta mobilidade dos nós. Então, como calcular as rotas necessárias para encaminhar os pacotes com sucesso desde a origem até ao destino em uma rede veicular ad-hoc? Pensando nisso, há diversos estudos com finalidade de comparar o desempenho dos algoritmos de roteamento aplicados à redes móveis, de modo que alguns estão sendo adaptados para a corrigir suas limitações e poderem ser aplicados a este tipo de rede, visto que alguns sofreram adaptações, enquanto outros foram criados (LUÍS, 2009). Dentre os principais algoritmos de roteamento, com foco nas redes veiculares,

temos os protocolos: ad-hoc, baseados em localização, baseado em *clusters*, por *broadcast* e geocast.

4.4.1 Protocolos Ad-hoc

As redes ad-hoc são caracterizadas por ser um tipo de rede que não utiliza infra-estruturas e permite a mobilidade dos nós. Ao contrário dos outros protocolos de roteamento infra-estruturados, os quais não apresentam um desempenho aceitável para redes VANETs.

São tradicionalmente divididos em duas grandes categorias: reativos e pró-ativos. Os Reativos caracterizam-se pelo fato de nem sempre possuírem as rotas para todos os nós da rede. Quando um nó precisa de uma rota para um certo destino na rede, inicia o processo de descoberta de rota e é finalizado quando a rota é calculada com êxito ou não exista uma rota disponível. As rotas já decobertas, são utilizadas enquanto o nó de destino permanecer alcancável ou não ser mais necessário. Os protocolos reativos mais conhecidos são AODV (*Ad hoc On Demand Distance Vector*) e o DSR (*Dynamic Source Routing*). Em contra partida, Boukerche (2004), afirmou que o desempenho dos protocolos AODV e DSR é bastante prejudicado pela constante alteração topológica da rede.

Os protocolos pró-ativos (*table-driven*) caracterizam-se por ter conhecimento das rotas para todos os nós existentes. Esse protocolo possui a vantagem de ter um reduzido atraso incial, pois as rotas já estão estabelecidas. Não obstante, a metodologia dos protocolos pró-ativos implica na existência de um tráfego adicional para controle da topologia, uma vez que é preciso manter as rotas existentes sempre atualizadas, consumindo uma maior largura de banda. São exemplos de protocolos pró-ativos: DSDV ('*Destination Sequenced Distance Vector*) e o WRP (Wireless Routing Protocol).

Os protocolos PRAODV (*PReemptive AODV*) e PRAODV-M (*PReemptive AODV - Maximum*) foram criados em cima do protocolo AODV, oferecendo um componente baseado na velocidade, na localização e na predição. A difererença é que no PRAODV é estabelecida uma ligação alternativa entre dois nós, antes mesmo da principal expirar, e a PRAODV-M, escolhe uma rota que prevê ficar ativa por mais tempo, ao contrário da AODV, que escolhe a rota mais curta (LUÍS, 2009).

O protocolo OLSR (Optimzed Link State Routing), realiza uma otimização da topologia, sendo este, o algoritmo mais utilizado em redes ad-hoc (JACQUET et al., 2001). Possui um modo de funcionamento bastante característico, pois cada nó, seleciona dentre

seus vizinhos, uma quantidade de nós suficientes de maneira a cobrir toda a vizinhança a dois saltos do nó, com objetivo de realizar o roteamento e retransmitir as mensagens.

4.4.2 Protocolos Baseados em Localização

Um dos primeiros protocolos baseados em localização foi o GPSR (*Greedy Perimeter Stateless Routing*), no qual baseia-se na informação geográfica em relação aos vizinhos. A vantagem deste protocolo está em manter a informação apenas sobre a topologia local, permitindo uma maior escalabilidade e um menor tempo para criação de novas rotas. Entretanto, o desempenho do GPSR é comprometido em ambiente reais, apresentando obstáculos e distribuição aleatória dos veículos.

O protocolo GSR (*Global State Routing*) foi criado com objetivo de corrigir a limitação do GPSR de modo a utilizar o *link-state* de forma que cada nó mantém uma tabela de conectividade, contendo todas as ligações existentes entre os diversos nós da rede, otimizando as decisões a nível de roteamento local. Avaliação feita por Füssler et al. (2003) indica que o GSR apresentou melhor desempenho em relação ao GPSR e melhora no atraso comparado ao DSR, do mesmo modo que uma melhor taxa de sucesso de entrega e menor ocupação da largura de banda se comparado com o AODV (LI; WANG, 2007).

O protocolo A-STAR é baseado no GSP e GPSR e incorpora um sistema de sensibilização de tráfego (*traffic awareness*) fazendo uso de mapas das estradas ordenados por utilização, de forma a poder definir suas rotas pelas estradas com maior conectividade, na tentativa de aumentar a probabilidade de sucesso na entrega dos pacotes. Devido à sensibilização de tráfego aplicado ao A-STAR, ele apresenta um melhor desempenho, entorno de 40%, na entrega de pacotes em relação ao protocolo GSR.

Foi proposto por Leontiadis e Mascolo (2007) o protocolo de roteamento denominado GeOpps (*Geographical Opportunistic routing for vehicular networks*), ele assume que todos os nós estão munidos com sistemas de posicionamento de maneira a encaminhar um pacote para um nó que está, teoricamente, em melhores condições de poder entregar ao seu destino final. Segundo Karp e Kung (2000), os resultados mostraram que o protocolo GeOpps tem um melhor comportamento do que o GPSR.

4.4.3 Protocolos Baseado em *Clusters*

Os protocolos de roteamento baseados em *Clusters* representam uma rede virtual criada por meio de nós de uma rede física, onde o grupo criado por um conjunto de nós interligados de maneira lógica, tem tendência a alterar rapidamente a sua composição.

Cada *cluster* pode apresentar um nó como líder, denomicado *cluster-head*, que é incubido pela coordenação da comunicação dos nós da rede. Os nós de um cluster podem se comunicar diretamente, porém a comunicação extra grupo deve ser feita somente pelo líder. A criação dos *clusters* é de suma importância para aumentar a escalabilidade dos protocolos de roteamento e está na estabilidade a chave para o desempenho destes algoritmos (LUÍS, 2009).

Em redes veiculares, a aleatoriedade da movimentação dos veículos faz com que os protocolos de roteamento baseados em cluster aplicados em redes móveis sejam frustrados, tais como: *Adaptative Clustering* e o MCDS *Minimum Connected Dominating Set* (DAS; BHARGHAVAN, 1997).

Como dito anteriormente, o sucesso de um cluster está na estabilidade, consequentemente, muitas pesquisas são realizadas com objetivo de tornar o grupo o mais estável possível a exemplo do protocolo COIN *Clustering for Open IVC Networks* que utiliza informações sobre mobilidade para formação do *cluster*. Entretanto, são adicionados ao algoritmo, as intenções do motorista do veículo como também a dinâmica veicular. De acordo com Blum, Eskandarian e Hoffman (2003), resultados demonstram que as otimizações feitas melhoram o desempenho do protocolo, identificando um aumento de 192% no tempo médio de vida de um *cluster* e uma redução de 42% no número de alterações dos integrantes do grupo.

O protocolo CBLR (*Cluster-Based Location Routing algorithm*) apresentado por Santos, Edwards e Edwards (2004), utiliza conceitos de *cluster* juntamente com informações de localização. Este protocolo, se comparado ao AODV e DSR, desmostrou um desempenho superior ao atraso *end-to-end* e à taxa de sucesso de entrega.

4.4.4 Protocolos por *Broadcast*

Os protocolos de roteamento baseados em broadcast consistem em transmitir informações por todos os nós que façam parte da rede. Em VANETs, este tipo de difusão

é muito utilizado para compartilhar informações sobre o tráfego, condições da estrada, condição do clima, entre outros (LUÍS, 2009). Assim, fica garantido que todos receberam a informação. Entretanto, o funcionamento destes protocolos não é indicado para redes consideravelmente grandes, podendo gerar um efeito de tempestade de broadcast (*broadcast storm*), aumentando a probalilidade de colisões de pacotes e o consumo da largura de banda, comprometendo o desempenho.

O protocolo BROADCOMM (*BROADcast COMMunications*) apresenta algumas semelhanças com os protocolos baseados em clusters ao dividir a auto-estrada em células e utilizar o conceito de *cluster-head*, aqui chamando de *cell-reflectors* (DURRESI; DURRESI; BAROLLI, 2005). Neste caso, a diferença é que as *cell-reflectors* devem ficar geometricamente no centro da célula. A função do *cell-reflectors* é difundir as informações de emergências entre as células. Contudo, este protocolo funciona apenas para ambiente de auto-estrada.

4.4.5 Protocolos *Geocast*

O protocolo de *Geocast* (*Geocast Routing*) leva em consideração a posição/localização, e seu objetivo é entregar um pacote aos nós que estão em uma determinada região denominada ZOR (*Zone of Relevance*). A implementação deste protocolo deve levar em consideração a integração de um serviço de multidifusão em conjunto com o agrupamento dos nós conforme seu posicionamento geográfico, criando assim as ZORs.

Uma implementacão do algoritmo *Geocast* foi utilizado na construção do protocolo *Message Dissermination Process* proposto por Briesemeister, Schafers e Hommel (2000), cujo objetivo é evitar colisão de pacotes e diminuir o número de retransmissões. No momento que um nó recebe um pacote, ele não o encaminha imediatamente esperando um tempo de modo a poder tomar uma decisão referente à retransmissão. O tempo de espera é baseado na distância do nó que lhe enviou o pacote, ou seja, quanto maior a distância menor é o tempo de espera. Quando o período de tempo expira, o pacote somente é retransmitido se a informação não tiver sido recebida novamente. Esse controle no envio dos pacotes faz com que seja menos provável a existêcia de *broadcast stoms* e a propagação de pacotes seja mais eficiente.

Os protocolos DRG (*Distributed Robust Geocast*) e ROVER (*RObust Vehicular Routing*) projetados por Kihl, Sichitiu e Joshi (2008), sendo o DRG um protocolo com

foco em grandes cenários, adaptável às constantes mudanças da topologia das redes veiculares e fornece um sistema de encaminhamento rápido e confiável. Em contra-partida, o protocolo ROVER oferece uma difusão *multicast* confiável, tendo como base, um processo de descoberta de rotas dentro da ZOR.

4.4.6 Comparação dos Protocolos de Roteamentos

Ainda não foi encontrada a melhor forma de se criar um protocolo de roteamento para VANETs, porém há autores que consideram um roteamento baseado em cluster mais viável em relação aos outros modelos (LUÍS, 2009). A aplicabilidade do protocolo faz aumentar o número de pesquisas existentes. Sendo que se há alguns protocolos que possuem desempenho favorável em cenários urbanos, isso já não acontece em ambientes de alta mobilidade, e vice-versa. A Tabela 3 faz um apanhado geral sobre os protocolos de roteamento e os cenários mais indicados.

Protocolo de Roteamento	Comunicaçãc	Informação Sobre Posição	Estrutura Hierárquica	Cenário de Mobilidade
AODV	Unicast	Não	Não	—
DSR	Unicast	Não	Não	—
OLSR	Unicast	Não	Não	—
PRAODV-M	Unicast	Seleção de rotas	Não	Auto-estrada
GPSR	Unicast	Encaminhamento de Pacotes	Não	—
GSR	Unicast	Encaminhamento de Pacotes	Não	Urbano
A-STAR	Unicast	Encaminhamento de Pacotes	Não	Urbano
GeOpps	Unicast	Encaminhamento de Pacotes	Não	Urbano
COIN	Unicast	Formação de Cluster	Sim	Auto-estrada

CBLR	Unicast	Encaminhamento de Pacotes	Sim	Circuito circular e quadrangular
BROADCOMM	Broadcast	Formação de células	Sim	Auto-estrada
Msg. Diss. Proc.	Geocast	Encaminhamento de Pacotes	Não	Auto-estrada
DRG	Geocast	Encaminhamento de Pacotes	Não	Auto-estrada
ROVER	Geocast	Encaminhamento de Pacotes	Sim	Auto-estrada

Tabela 3 – Protocolos de roteamento aplicados a redes veiculares (LUÍS, 2009)

4.5 Aplicações

Há uma série de aplicações que podem ser desenvolvidas para redes veiculares com foco em cidades inteligentes, entretanto para cada aplicação, é necessário conhecer as características que as cercam, tais como: segurança da informação; tempo de resposta; garantia na entrega; entre outros.

Segundo DANTAS (2011), há uma relação entre velocidade dos veículos e a distância entre eles. Esta relação é diretamente proporcional, indicando que quanto maior a velocidade, maior deve ser a distância entre os veículos. Isto ocorre naturalmente, pois o motorista precisa considerar-se seguro no trânsito. Porém, esta condição é dinâmica variando a cada instante, e para agravar, a relação entre velocidade do fluxo e a distância dos veículos, define um fator de realimentação positiva, e sistemas com esta característica tendem à instabilidade, gerando transtornos no trânsito. Então, um sistema capaz de alertar o motorista caso o mesmo esteja a uma distância "não segura" do veículo da frente, pode ser alcançado com sistemas baseados em VANETs.

Outra aplicação, com foco em redes veiculares, pode ser produzida por meio de alertas de colisão em uma via de fluxo intenso. O condutor possui um tempo de percepção mais um tempo de reação e somente após a soma dos tempos poderá executar uma reação. Esta situação mostra a importância de agregar aos veículos uma rede de comunicação no

qual os mesmos e seus motoristas poderão trocar informações de ocorrências emergenciais e preventivas no trânsito. Esta situação fica claramente evidenciada na Figura 11. Há também a possibilidade de aplicações voltadas para o entretenimento com o uso de *chats*, *downloads* de vídeos ou até mesmo propagandas comerciais nas proximidades do veículos.

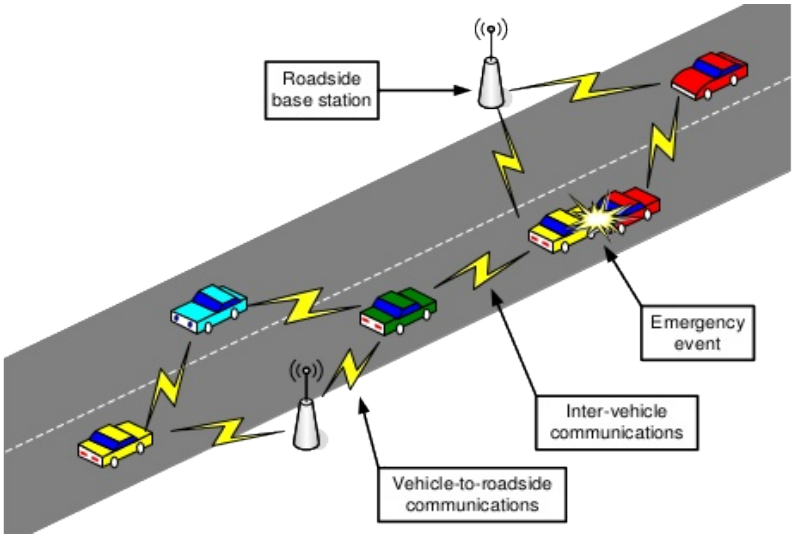


Figura 11 – Exemplo de aplicação VANET (QIAN; LU; MOAYERI, 2008).

Por ser uma área crescente, as redes veiculares tem atraído empresas automobilísticas e órgãos controladores a exemplo do órgão americano de pesquisa de inovação tecnológica, RITA (*Research and Innovative Technology Administration*) o qual é coordenado pelo departamento de transporte dos EUA. O projeto Car 2 Car (*Communication Consortion*), criado por cinco fabricantes de automóveis europeus (BMW, Volvo, Volkswagen, Honda, e Audi) e apoiada por fornecedores de equipamentos, organizações de pesquisa e outros parceiros, com objetivo de aumentar ainda mais a segurança e a eficiência do tráfego rodoviário através de Sistemas de Transporte Inteligente Cooperativos (C-ITS) com a comunicação veículo para veículo (V2V) suportada pela comunicação veículo infraestrutura (V2I) .

4.6 Considerações Finais do Capítulo

Neste capítulo foram apresentadas as peculiaridades e os principais desafios relacionados às redes veiculares tais como: alta mobilidade; alta e baixa densidade; segurança e privacidade e escalabilidade, com as quais a presente pesquisa se debruça na tentativa de criar alternativas utilizando em conjunto a computação em nuvem para gerenciamento

virtualizado de uma VANET. Explana também sobre as arquiteturas aplicadas às VANETs, a importância da comunicação segura para proteção dos motoristas e passageiros, citando alguns tipos de ataques possíveis e os principais algoritmos de roteamentos aplicados à redes veiculares.

5 Arquitetura I9VANET

5.1 Visão Geral

A arquitetura proposta tem a finalidade de criar redes veiculares virtuais em nuvem com foco no auxílio das soluções para os principais desafios relacionados à VANETs tais como: alta densidade, baixa densidade, alta mobilidade, segurança e privacidade, entre outros.

A arquitetura consiste em um modelo aberto, flexível e dividido em módulos com funções bem definidas. Cada módulo possui funcionalidades específicas e comportamentos bem definidos. Sendo possível estender suas operações ou até mesmo substituí-las de maneira que atenda às novas necessidades.

5.2 Módulos da Arquitetura I9VANET

A Figura 12 mostra os módulos necessários da arquitetura I9VANET: *Applications, Server Management Cloud, Routing between Nodes, Secutity OBU/RSU, Infra-Cloud Communication* e *Vehicle-Cloud Communication*. Sendo que cada módulo segue um modelo de arquitetura com objetivos bem definidos, visto que suas interfaces de comunicação são normatizadas por métodos específicos, permitindo substituir um módulo por outro, com a mesma característica, sem que interfira no funcionamento da plataforma.

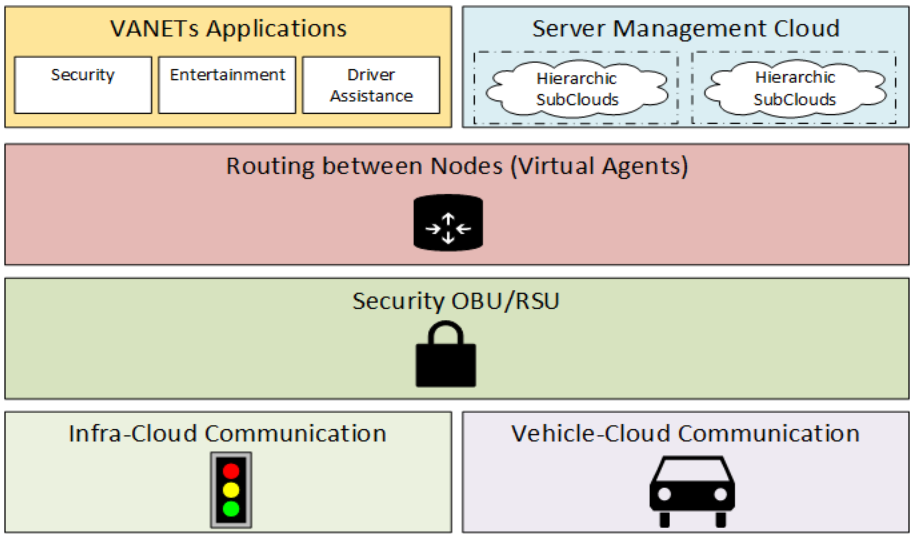


Figura 12 – Módulos definidos para a arquitetura I9VANET.

Todos os recursos que podem ser customizados, tais como: a tecnologia de comunicação; algoritmo de roteamento; modelo de segurança; implementação de eventos, devem estar definidas no arquivo de configuração **idclassload.properties** como os itens seguintes.

- **ICommunication**:
  (br.com.virtualVanets.infraVehicle.communication.CommunicationWebSocketImpl)
  Indica a classe responsável por realizar a comunicação dos servidores com os dispositivos.
- **CommunicationV2A**:
  (br.com.virtualVanets.infracloud.communication.impl.CommunicationI2AImpl)
  Classe que deve enviar e receber os comandos na comunicação entre o Veículo (OBU) e o seu agente virtual.
- **CommunicationI2A**:
  (br.com.virtualVanets.infracloud.communication.impl.CommunicationI2AImpl)
  Classe que deve enviar e receber os comandos na comunicação entre o dispositivo à beira da rodovia (RSU) e o seu agente virtual.
- **ListenerInfraEquipament**:
  (br.com.virtualVanets.applicationModel.listener.DefaultListenerInfraEquipament)
  Responsável por tratar os eventos necessários da aplicação com foco na comunicação I2A.
- **ListenerVehicle**:
  (br.com.virtualVanets.applicationModel.listener.DefaultListenerVehicle)
  Trata os eventos necessários da aplicação com foco na comunicação V2A.
- **EventInfraEquipament**:
  (br.com.virtualVanets.infracloud.listener.SVVEventInfraEquipament)
  Define a lista dos possíveis eventos de um RSU.
- **EventVehicle**:
  (br.com.virtualVanets.vehiclecloud.listener.SVVEventVehicle)
  Define a lista dos possíveis eventos de um OBU.
- **SecurityModel**:
  (br.com.virtualVanets.security.I9Security)

Responsável por realizar a assinatura e criptografia de todo conteúto trocado entre os agentes móveis e os dispositivos.

- **RouterNetwork**:
(br.com.virtualVanets.infraVehicle.communication.CommunicationWebSocketImpl)
Implementa o algoritmo de roteamento a ser utilizado na arquitetura.

5.2.1 Módulo de Comunicação

O módulo de comunicação é responsável por definir a troca de mensagens entre os servidores e os dispositivos OBU e RSU. Dessa forma, é possível mudar o protocolo de comunicação sem interferir na arquitetura. Basicamente é enviado, dos dispositivos para os servidores em nuvem, informações dos veículos a cada 10 segundos, podendo ser ajustado à medida que uma aplicação necessite de um menor tempo. Este tempo foi definido devido à alguns modelos de aparelhos de monitoramento GPS veicular utilizarem como menor tempo de envio possível. Exemplos de alguns aparelhos de monitoramento veicular: TK103, AVL05, ST 340, entre outros.

5.2.1.1 Comunicação Infra-Cloud

A forma como a comunicação será efetivamente implementada, não é preocupação da arquitetura, cabe ao desenvolvedor utilizar a tecnologia (socket, webservices, websocket, push, etc) que mais se adequa à sua necessidade e/ou região. As informações trocadas são de entrada e saída, portanto, os equipamentos RSU podem enviar como também receber informações dos servidores. Em VANETs pode existir a comunicação entre veículos e dispositivos à beira da rodovia (V2I), todavia, não é o propósito desta arquitetura realizar, de forma direta, a comunicação entre os equipamentos, contudo será realizada entre os respectivos agentes virtuais em nuvem.

Todos os equipamentos RSU serão virtualizados e representados por agentes virtuais. Sendo assim, pode haver a comunicação entre agente RSU e o agente do veículo (AV2AI), como também entre agentes RSUs (AI2AI), como mostrado na Figura 13. Os dispositivos RSUs disponibilizam as seguintes informações: identificação, latitude, longitude, altitude, temperatura, pressão, umidade relativa do ar e um campo textual para informações extras.

Tais informações podem ser úteis para informar aos motoristas o estado atual de cada região. A plataforma prevê 3 operações:

- **Conectar**: realiza a conexão com o servidor, informando identificador do dispositivo, a latitude e longitude;
- **Enviar Mensagem**: envia mensagens para rede. O conteúdo das mensagens variam de acordo com a aplicação;
- **Desconectar**: fecha a conexão com o servidor.

Para implementar a comunicação entre o agente e o veículo (I2AI) é necessário implementar a classe abstrata **CommunicationI2A** e sobrescrever o método **sendMsg**. Esse método é executado pela arquitetura de maneira transparente e deve enviar o comando para o dispositivo.

5.2.1.2 Comunicação Veículo-Cloud

Define a comunicação entre os servidores e os equipamentos instalados nos veículos (OBU). Os detalhes da implementação podem ser substituídos permitindo utilizar a melhor tecnologia do momento. Assim como no módulo de Comunicação Infra-Cloud, a comunicação V2V não ocorrerá diretamente, como ocorre nos modelos tradicionais, ela será realizada através dos agentes virtuais, que nada mais são do que a representação virtual dos veículos. Desta forma, a comunicação se dará entre os agentes (AV2AV) e entre o agente e o veículo físico (V2A) este processo é mostrado na Figura 13.

O veículo deve disponibilizar as seguintes informações: identificação, latitude, longitude, altitude, velocidade, direção, tipo operação e um campo textual. A plataforma prevê 4 operações que podem ser extendidas, são elas:

- **Conectar** (CONNECT_CODE): realiza a conexão com o servidor, informando identificador do dispositivo, a latitude e longitude;
- **Movimentar** (MOVIMENTATION_CODE): este comando é executado a cada 10 segundos, com o intuito de informar a geolocalização, velocidade e direção do mesmo;
- **Enviar Mensagem** (SEND_BROADCAST_CODE): envia mensagens para os veículos da rede, sendo que o conteúdo das informações variam de acordo com a aplicação;

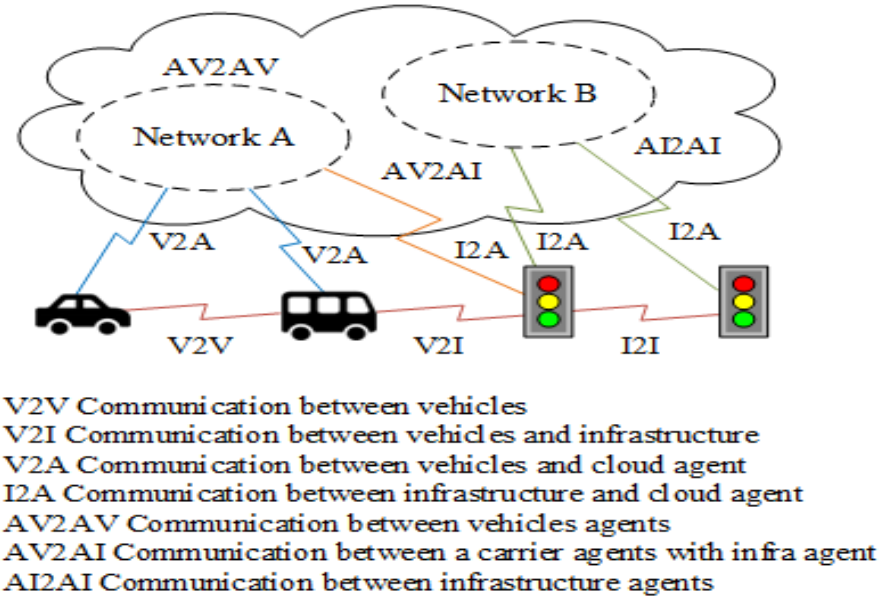


Figura 13 – Modelo de comunicação V2A, I2A, AI2AI e AV2AI (Próprio autor).

- **Desconectar** (DISCONNECT_CODE): fecha a conexão com o servidor removendo-o da rede.

Para implementar a comunicação entre o agente e o veículo (V2AV), é necessário implementar a classe abstrata **CommunicationV2A** e sobrescrever o método **sendMsg**. Este método executa a classe responsável pela conexão efetivamente definida.

5.2.2 Módulo de Segurança

A segurança na troca de informações em redes veiculares é um dos desafios que devem ser solucionados antes mesmo da implantação de uma solução VANET ser posta em prática. Então, como garantir os princípios da autenticidade, integridade, confidencialidade e não repúdio na troca de mensagens entre os veículos, já que é inviável fornecer as chaves de segurança de todos os veículos durante a comunicação?

Na arquitetura I9VANET, foi pensado em um processo de comunicação entre os equipamentos, OBU e RSU, e os servidores em nuvem, de forma que deve garantir os princípios da autenticidade, integridade, confidencialidade e não repúdio, através do uso de assinatura digital baseado em algoritmos assimétricos e criptografia dos dados. Toda via, este último foi definido de forma parametrizada para que fosse feita análise comparativa entre a comunicação aberta e criptografada.

A geração das chaves para cada equipamento deve ser controlada pelo órgão responsável pelos veículos, a exemplo do Brasil, o DENATRAN (Departamento Nacional de Trânsito). Quando um novo veículo for cadastrado em sua base, "veículo zero quilômetro", devem ser geradas as chaves pública e privada, sendo que a pública deve ser enviada para os servidores em nuvem responsáveis pelo gerenciamento da plataforma I9VANETs e a privada, inserida no equipamento embarcado do veículo. A cada renovação de licenciamento, um novo token deve ser gerado e fornecido ao equipamento e publicado nos servidores. Como também, em caso de venda de um veículo, ao passar para o nome do novo proprietário, deve ser gerado um novo token. Para a geração das chaves assincronas, foi utilizado o algoritmo RSA com um tamanho de 1024 bits.

Se o parâmetro de criptografia estiver habilitado, o primeiro passo, no momento da conexão, é a troca de chave secreta para que toda comunicação seja segura. Para isso, assim que o comando *onConnection* for executado, o servidor enviará o comando *createSecretKey* com a chave secreta, criptografada com a chave pública do veículo. Assim o mesmo poderá descriptografar com sua chave privada e utilizá-la na criptografia de cada envio e recebimento de mensagens de agora em diante. Para criação da chave secreta, foi utilizado o algoritmo AES com chave de 128 bits.

No momento que um equipamento desejar enviar uma mensagem para seu agente em nuvem, ela deve ser assinada com sua chave privada, assim através da verificação da assinatura com a chave pública, o agente poderá confiar no remetente. Porém, a mensagem será assinada com a chave privada do servidor para que o equipamento, OBU ou RSU, possa validar através da chave pública do servidor.

Alterar o modelo de segurança adicionando novas regras é possível através da implementação da classe **ASecurityModel** e dos métodos **verifySign, sign, encrypt e decrypt**. A chamada dos métodos é feita pela arquitetura de maneira transparente, não sendo necessário intervenção do usuário.

Como foi dito anteriormente, as redes veiculares são vulneráveis a muitos ataques e a arquitetura procurou se proteger aos principais como mostrado na Tabela 4.

Tipos de Ataques	Opções	Observação

DoS	X	Este ataque não seria com foco em um OBU ou RSU, apenas fará sentido ocorrendo nos servidores da plataforma I9VANET. Para proteção os servidores devem se proteger com *firewall*.
DDoS	X	O mesmo do DoS.
Supressão de mensagem	X	o repasse das mensagens não depende do veículo físico e sim do agente virtual em nuvem.
Buraco Negro	X	quem repassa as mensagens é o agente em núvem, não sendo possível acontecer este tipo de ataque.
Jamming	–	como a arquitetura depende da telefonia celular, ou redes wifi, entre outras já mencionadas, então este sinal pode ser afetado impossibilitando o acesso dos veículos à nuvem.
Falsificação de endereço	X	todo nó deve ser reconhecido pelo órgão controlador de veículos por meio de um certificado.
Mascaramento	X	Todo certificado emitido deve ser único e intransferível e renovado anualmente ou a cada operação de compra e venda.
Replicação do certificado	–	Este tipo de ataque envolve protege o certificado do veículo para que não possa ser utilizado por outro, fugindo do escopo da arquitetura.
Sybil	X	Todo veículo deve possuir um certificado reconhecido por alguma entidade certificadora, não sendo possível simular mais veículos.

Falsificação nos dados de GPS	–	É difícil se proteger deste ataque, pois o sinal de GPS emitido por satélites são abertos.
Ilusão	–	A proteção para esse tipo de ataque consiste em criar uma comunicação inviolável entre os sensores e o equipamento que se conecta na aplicação em nuvem.
Injeção de informação falsa	X	Não há comunicação direta entre os veículos e a aplicação desenvolvida para realizar a comunicação virtual pode realizar uma análise de toda informação.
Modificação de mensagem	X	Não havendo comunicação direta entre os veículos e todas as mensagens sendo assinadas e criptografadas, não há como aldulterar uma mensagem.
Análise de tráfego	X	É uma taque em redes ad-hoc, não sendo aplicado à arquitetura.
Força bruta	X	A comunicação é ponto a ponto e o conteúdo é criptografado, dificultando ainda mais este tipo de ataque.
Revelação de identidade	X	A identidade do veículo não é informada durante as comunicações.
Ataques contra não repúdio	X	As mensagens da rede veicular não depende de um veículo para se propagar.
Ataques contra a privacidade	X	Todo conteúdo pode ser criptografado, assim a privacidade do veículo é garantido.
Conluio	X	A segurança é garantida pela arquitetura e não por critérios de confiança que pode ser denegrido por este tipo de ataque.

Tabela 4 – Quadro de análise sobre os tipos de ataques e a arquitetura I9VANETs.

5.2.3 Módulo de Gerenciamento dos Servidores

Partindo do princípio que seria complexo prever a quantidade de veículos conectatos a um servidor, isto criou a necessidade de construir uma estrutura de gerenciamento com objetivo de aumentar a capacidade de dispositivos conectados à plataforma. Então, pensando nisso foi construída uma forma de gerenciamento o qual pode ser modificada, onde os servidores são distribuídos hierarquicamente de forma que permitam controlar regiões, denominadas domínios, separadamente e independentemente conforme Figura 14. O servidor nível 0 é a raiz da árvore e responsável por gerenciar os nós abaixo dele. Os servidores "Pais" são os responsáveis por realizar a localização de um domínio no momento que um veículo mudar de área, então ele deve possuir todos os domínios de seus filhos para que possa informar ao veículo, o novo endereço de conexão.

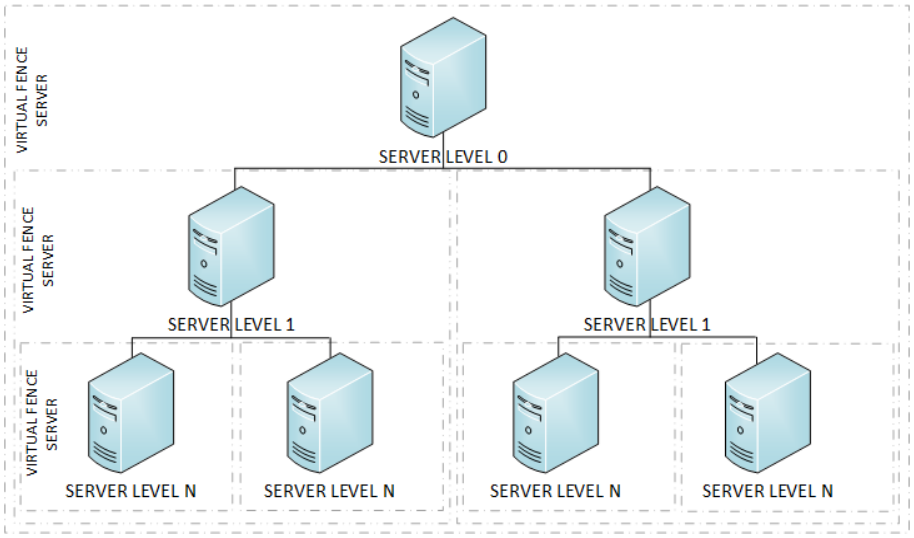


Figura 14 – Organização dos servidores na nuvem.

A estrutura não é uma árvore binária e sim uma formação em árvore com N nós filhos. Quando um veículo que está conectado a um servidor precisar conectar-se a outro, a mudança deve ser solicitada ao servidor pai imediato, caso este não seja o de destino, deve perguntar ao pai deste e assim sucessivamente, até encontrar o servidor responsável pelo domínio de destino ou o servidor nivel 0 será responsável por este gerenciamento. A Figura 15 mostra como exemplos de domínios, onde cada cerca virtual representa um servidor.

Quando um veículo iniciar a transmissão será conectado automaticamente ao servidor Nível 0, informando o comando *StartConnection*, o qual realizará uma busca pela

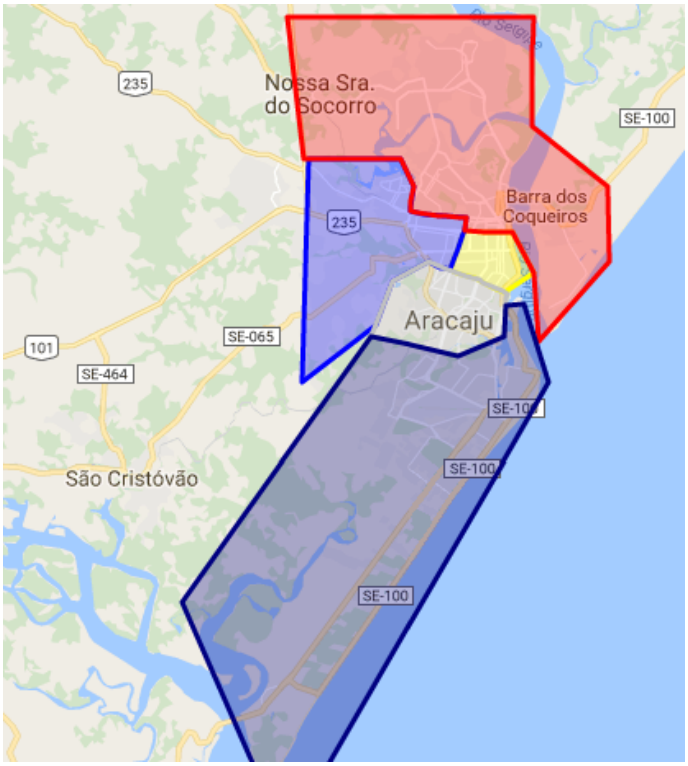


Figura 15 – Exemplo de domínios.

coordenada GPS enviada pelo veículo para localizar o servidor responsável pelo domínio onde encontra-se o veículo, podendo ocorrer duas situações:

1. O veículo apresenta-se em uma área sem cobertura de cerca virtual, assim o próprio servidor raiz será responsável por gerenciar a rede VANET deste veículo;
2. O veículo encontra-se em um domínio válido que é de responsabilidade de algum servidor filho. Então o servidor nível 0 irá passar o comando, *ChangeServer*, informando o endereço do servidor que deve ser responsável pela rede desse domínio, e assim o veículo faz uma nova conexão para o novo servidor.

Cada movimentação do veículo deve ser transmitida para o servidor do domínio, pelo comando *SendMoviment*, o qual verificará se o veículo ainda está em sua região, de acordo com a cerca virtual. Caso não esteja mais sob seu domínio, o servidor irá enviar o comando *ChangeServer* com o endereço do servidor pai. Assim, o veículo poderá refazer a conexão com este novo servidor, através do comando *StartConnection*, o qual deverá verificar se a coordenada pertence a este servidor ou a algum de seus filhos. Este processo é realizado para todos os níveis da hierarquia.

Em todos os servidores deve existir uma base de dados para definir a hierarquia, os domínios geográficos como também o cadastro dos dispositivos. As tabelas 5 e 6 mostram, respectivamente, o dicionário de dados da tabela *server* e *device* utilizadas no sistema.

Tabela *Server*		
Coluna	Tipo do Dado	Descrição
server_id	*integer*	Identificador único no sistema
address	*varchar*	Endereço IP do servidor
server_idsuper	*integer*	Identificador do servidor pai
dominio	*polygon*	Cerca virtual de responsabilidade do servidor

Tabela 5 – Colunas das tabelas *Server*.

Tabela *Device*		
Coluna	Tipo do Dado	Descrição
device_id	*integer*	Identificador único no sistema
identification	*varchar*	Identificador do dispositivo (Imei, Mac Address, etc.)
type	*char*	Tipo do dispositivo (OBU ou RSU)
public_key	*byte[]*	Chave pública do dispositivo
	integer	

Tabela 6 – Colunas da tabela *Device*.

A cada envio do comando *SendMoviment* pelos veículos, o servidor deve verificar se o mesmo ainda encontra-se no mesmo domínio. Como este processo é executado a cada 10 segundos por cada veículo, foi necessário realizar essa consulta em memória ao invés de ir no banco. Para isso, foi utilizada a API GEOTools (HALL; LEAHY, 2008).

O GeoTools é uma biblioteca feita em Java *open source* que provê ferramentas para manipulação de dados geoespacial em memória. O objetivo da biblioteca é permitir que programadores que estejam desenvolvendo aplicações geoespaciais se concentrem na construção do negócio fim, enquanto reutilizam ferramentas genéricas para funções básicas (HALL; LEAHY, 2008).

A mesma funcionalidade do GeoTools poderia ser atingida com as funções do PostGIS (OBE; HSU, 2015), porém ao realizar pequenos testes comparativos, foi percebido que o tempo da consulta é 10 vezes maior que a do GeoTools, o que poderia comprometer o processo de avaliação da arquitetura. Entretanto, o PostGIS foi utilizado apenas para armazenar o polígono que representa as cercas virtuais dos domínios.

5.2.4 Módulo de Roteamento

O módulo de roteamento em uma rede veicular é um dos grandes desafios, e por isto é foco de estudos na busca de soluções que viabilizem a rede. Na arquitetura I9VANET proposta, essa preocupação está presente, porém a plataforma permite usar novos

algoritmos sem a necessidade de preocupar-se com detalhes, como segurança, protocolo de comunicação ou quantidade de veículos.

Toda comunicação entre os nós é realizada entre os agentes virtuais da rede (AV2AV e AV2AI e AI2AI). As regras de roteamento podem ser substituídas ou expandidas, basta implementar novos algoritmos seguindo o modelo da arquitetura do módulo. Quando houver a necessidade de enviar algo para algum nó físico da rede, esta mensagem deve ser enviada através do Módulo Comunicação Veículo-Cloud ou Infra-Cloud e eles se responsabilizarão pela entrega.

O módulo de roteamento pode ser definido de acordo com a necessidade, bastando implementar a classe **RouterNetwork**, porém para fins de avaliação da arquitetura, foi implementado o algoritmo conhecido como *geocast routing*, no qual o objetivo é entregar um pacote aos nós que pertencam a uma certa região, denominada *Zone of Relevance* que utiliza um OBU como nó cabeça de uma rede veicular.

Os veículos quando iniciam o processo de conexão com o servidor, buscam uma rede já existente para se conectar. O veículo "cabeça", ou seja o responsável por ter criado a rede, deve estar a uma distância máxima de 1 KM, parametrizável de acordo com a necessidade da aplicação, e seguir na mesma direção na via, caso contrário, este veículo criará uma nova rede esperando que novos veículos solicitem acesso.

A cada transmissão de um veículo, através da operação **sendMoviment**, o algoritmo de roteamento seguirá uma dentre três opções, são elas: permanecer na rede atual; entrar em uma nova; criar uma nova rede veicular.

5.2.5 Módulo de Aplicações

Permite adicionar novas aplicações tendo como base os recursos disponibilizados pelos outros módulos. Porém, para que as aplicações possam enviar e receber informações da arquitetura, é necessário definir um contrato com as classes dos módulos, através do uso de interfaces e classes abstratas. Sendo assim, quando um veículo enviar uma informação para seu agente virtual, a arquitetura irá executar um método padronizado de uma classe da aplicação, fazendo com que os dados sejam entregues corretamente.

Podem ser implementados váriados tipos de aplicações com os dados que a arquitetura já disponibiliza, porém caso seja necessário que uma aplicação possa modificar

os dados de entrada e saída da arquiteura, é possível utilizar o campo mensagem para adicionar novas informações e assim atender às novas demandas.

5.3 Processo de Negócio

As Figuras 16 e 17 mostram os diagramas de processos de negócio da plataforma I9VANET com e sem criptografia, respectivamente. No primeiro processo, o início, Figura 16(1), se dá quando o veículo é ligado e abre a primeira conexão com o servidor raiz, Figura 16(2). Em seguida, o processo de negociação da chave a ser utilizada na criptografia Figura 16(3), a chave é criptografada, pelo servidor com a chave pública do veículo e enviada para o nó da rede, Figura 16(4). A partir deste momento, toda comunicação será criptografada e em ambos os modelos, toda mensagem será assinada.

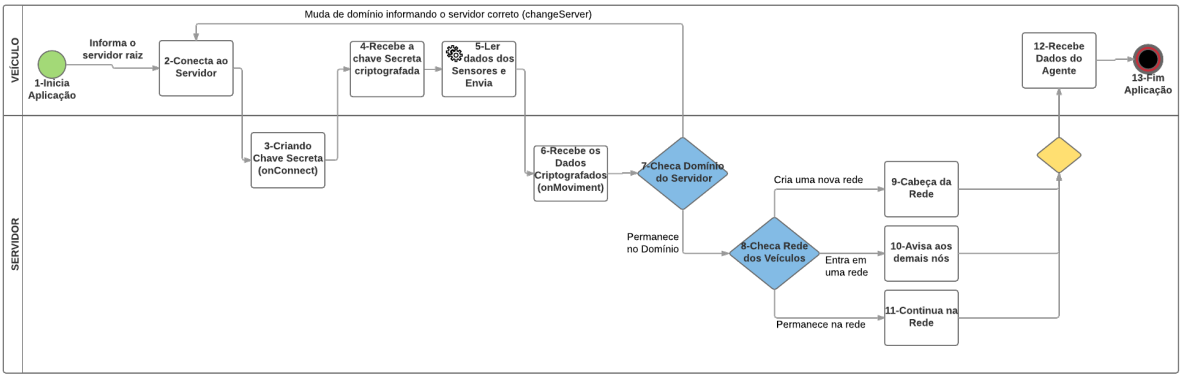


Figura 16 – Modelo de processo de negócio da plataforma I9VANET com criptografia.

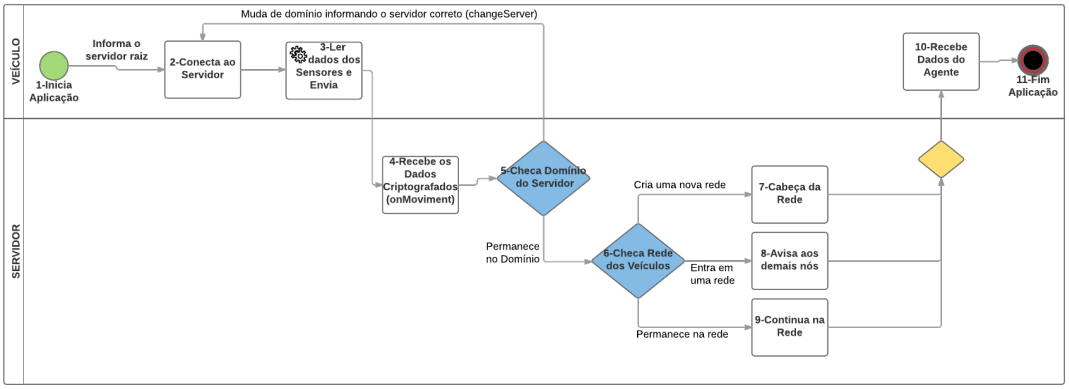


Figura 17 – Modelo de processo de negócio da plataforma I9VANET sem criptografia.

Após estabelecido o canal seguro, é iniciada uma *thread*, Figura 16(5), responsável por obter os dados dos sensores do veículo e enviá-los para o servidor, Figura 16(6), dentre os dados estão a latitude, longitude, velocidade e direção. O recebimento dos dados é

feito pelo método **onMoviment** que irá verificar em qual domínio o servidor deve ficar conectado, Figura 16(7). Se a coordenada geográfica indicar que o veículo deve mudar de servidor, então, o comando **changeServer** é acionado informando o endereço do servidor "pai", voltando para o processo, Figura 16(2), para recriar a conexão e segurança do canal. Caso permaneça no mesmo domínio, deve realizar uma análise sobre a rede que o veículo pertence, verificando se deve permanecer na rede atual, Figura 16(11), cria uma nova rede, Figura 16(9), ou entrar em uma outra, 16(10).

O processo de execução da rede em si não está mapeado nesse fluxo, já que as mensagens a serem trocadas entre os nós da rede dependem da camada da aplicação que está sendo desenvolvida. Por exemplo, quais mensagens devem ser enviadas para os veículos da rede.

5.4 Considerações Finais do Capítulo

Neste capítulo é proposta a arquitetura I9VANET, o qual consiste em gerenciar redes veiculares com o auxílio da computação em nuvem, cujo objetivo é corroborar com a solução para os principais desafios relacinados à VANET. Também é apresentada a estrutura modular da arquitetura, a qual está dividida em seis partes, são elas: Comunicação Infra-Cloud e Comunicação Veículo-Cloud, são responsáveis pela comunicação do agente em nuvem e o dispositivo; o módulo de Segurança deve garantir a privacidade do remetente e proteção da informação; Gerenciamento de Servidores, tem o papel de proporcionar escalabilidade à infraestrutura; A Camada de Roteamento permite que o desenvolvedor gerencie os nós da rede de maneira virtual, facilitando a comunicação entre os agentes virtuais; Camada de Aplicação tem a função de definir a contextualização que o desenvolvedor quer dar aos dados coletados pelos veículos.

6 Avaliação da Plataforma

Este capítulo visa avaliar a plataforma desenvolvida sobre a arquitetura I9VANET por meio do paradigma GQM (*Goal-Question-Metric*) muito usado para planejar medições em projetos de *softwares* (SOLINGEN et al., 2002). As seguintes fases foram analisadas durante o processo de avaliação: fase de planejamento, de coleta de dados e de interpretação dos resultados.

O processo de avaliação iniciou-se na fase de planejamento com o intuito de analisar a plataforma com relação à eficácia e eficiência da solução. O experimento teve como público alvo os desenvolvedores de software interessados na arquitetura como gerenciamento de uma rede VANET em nuvem para ITS no contexto de cidades inteligentes.

Com base na proposta da plataforma I9VANET, foram realizados dois tipos de experimentos. O primeiro, com o intuito de avaliar a sua aplicabilidade em ambientes de redes veiculares reais, teve a função de avaliar o fluxo de dados referentes às velocidades das tecnologias atualmente utilizados na telefonia móvel, sendo 2G, 3G, 4G e 5G. O segundo experimento objetivou dimensionar a capacidade de operação e o comportamento da plataforma a partir de excessivas requisições, sem limite de velocidade. Ambos os testes extraíram dados estatísticos a partir de métricas pré-definidas e planejadas.

6.1 Definição

Analisar a plataforma I9VANET sob a ótica da eficácia e eficiência. Sendo que para a eficácia será verificado se as mensagens são entregues independente do tempo decorrido e se houve perda de informação, e a eficiência utilizará o tempo como critério para entrega das mensagens, do ponto de vista de múltiplos acessos concorrentes no contexto de redes veiculares com gerenciamento em nuvem.

6.2 Planejamento

O experimento tem como alvo, os desenvolvedores de soluções que visam melhorar a mobilidade urbana com o uso de VANETs. Onde o objetivo foi avaliar a capacidade de processamento do servidor e medir o tempo de latência das comunicações da telefonia móvel, através dos seguintes questionamentos:

- Qual a taxa de processamento por minuto?
- Qual a latência média de cada requisição para as velocidades de 2G, 3G, 4G e 5G?
- A latência média com a mensagem criptografada irá aumentar em relação à mensagen aberta?
- Qual a capacidade de processamento da plataforma I9VANET?

Para isso, foram utilizadas as seguintes métricas:

- número Total de requisições por min (TR/min);
- tempo de latência da comunicação (Lat);
- tempo de processamento de cada requisição no servidor (PT).

O tempo de latência representa o tempo que uma mensagem sai do ponto A para o ponto B, sendo que o tempo calculado pelo experimento foi o RTT (*round trip time*), tempo que a mensagem foi enviada e recebida de volta pelo veículo, sendo considerado como latência, a metade do RTT.

De acordo com Papadimitratos et al. (2008), as aplicações voltadas para as redes VANETs possuem alguns requisitos que devem ser respeitados e estão relacionados ao tipo de comunicação, ao tipo de mensagem, ao tempo de entrega, à latência (tempo de atraso máximo requerido) e a outros requisitos como mostra a Tabela 7. Então, o objetivo do experimento é avaliar se o tempo obtido se adequa aos requisitos mostrados.

Tabela 7 – Características das aplicações veiculares (PAPADIMITRATOS et al., 2008).

Aplicações	Tempo	Latência	Outros
Alerta de Veículo Lento	500ms	100ms	Alcance: 300m, alta prioridade
Alerta de Colisão em cruzamento	100ms	100ms	Posicionamento preciso em um mapa digital, alta prioridade
Pré Colisão	100ms	50ms	Alcance 50m, prioridade alta/média
Gerenciamento de Cruzamento	1000ms	50ms	Precisão de posicionamento menor que 5m
Download de Mídia	–	500ms	Acesso a internet e Gerência dos direitos
Assitência para direção ecológica	1000ms	500ms	Acesso a internet e disponibilidade do serviço

6.3 Cenário Proposto

O ambiente proposto para avaliação consiste em simular movimentações de veículos na cidade de Aracaju-SE Brasil, representando uma área de 174 quilômetros quadrados, transmitindo e recebendo informações dos servidores que compõem a plataforma I9VANETs. Foram definidos dois grupos de testes, o primeiro consistiu em avaliar o comportamento da arquitetura referente à quantidade de veículos em conjunto com a limitação das velocidades utilizadas pela telefonia móvel. O segundo, avaliou a capacidade de processamento dos servidores sem limitação da velocidade de acesso.

Cada teste realizado, do primeiro grupo, levou em consideração quantidades diferentes de dispositivos representando os veículos, sendo da seguinte forma: 50, 100, 200 e 400. Para cada faixa, também foi ajustada a largura de banda para definir a velocidade máxima de comunicação de cada dispositivo, seguindo os modelos 2G, 3G, 4G e 5G, cuja velocidades definidas foram respectivamente: 400 Kbps, 2 Mbps, 100 Mbps e 1 Gbps.

Todo o ambiente, tanto o cliente quanto os servidores, foi montado em máquinas virtuais em 12 estações físicas. No caso dos servidores, foram criadas 4 máquinas com Linux Ubuntu Server 16.04 com o postgresql 9.5 e plugin postgis 2.0 e Glassfish 4.1.1 como servidor de aplicação. Todos os servidores possuíam a mesma configuração, sendo 1 GB de RAM e 2 processadores virtuais, e foram executados em uma única máquina física cuja configuração está presente na Tabela 18. A disposição dos servidores virtuais foi definida em dois níveis hierárquicos, sendo o servidor nível 0, a raiz da hierarquia, e como filhos, 3 servidores no nível 1.

Para simulação dos clientes, os OBUs e RSUs, foram utilizadas máquinas virtuais com linux Ubuntu Server 14.04 LTS 32 bits com o Mininet 2.2.1 configurado. Cada rede virtual Mininet instanciou entre 25 e 50 hosts, para que seu desempenho não fosse comprometido. Todas as máquinas com Mininet utilizaram a mesma configuração: 2GB de RAM e 2 processadores virtuais. O modelo do ambiente é mostrado na Figura 18.

Foi criado um script em Python, conforme Código 6.1, com intuito de automatizar a criação dos hosts, limitar largura de banda entre 2G, 3G, 4G e 5G, iniciar a execução da aplicação responsável por simular as funcionalidades dos OBUs e RSUs, realizar a comunicação com a plataforma I9VANET e registrar em *log* os dados referentes à comunicação de cada *host*, em arquivos individuais.

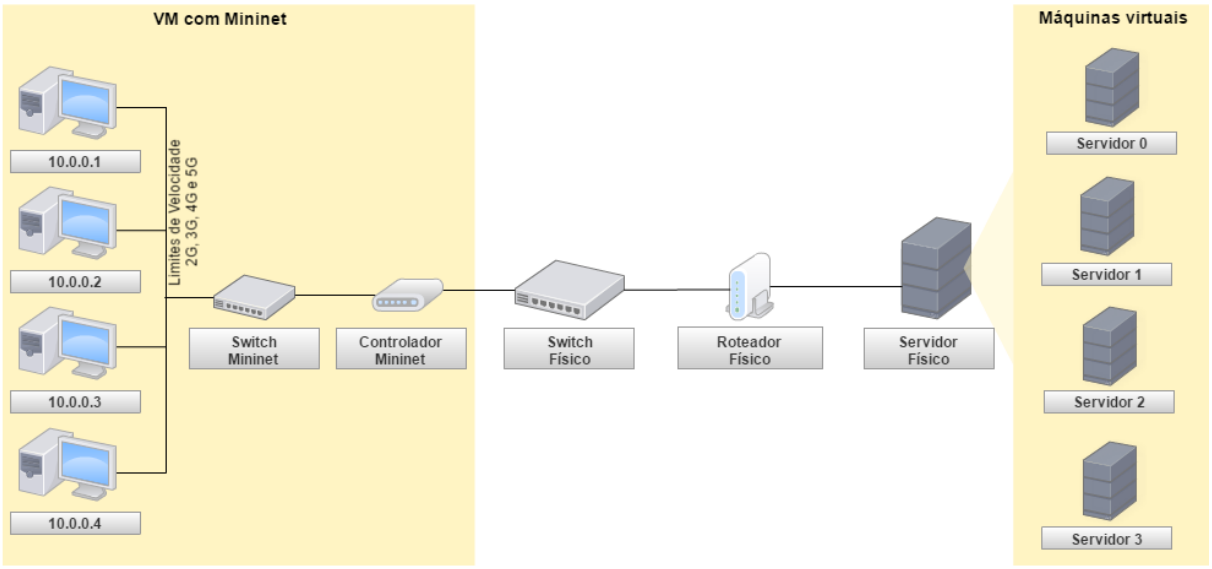


Figura 18 – Configuração do ambiente utilizado para a realização dos experimentos. Fonte: criada pelo autor.

Um *host* virtual tem a função de representar um veículo real, inclusive com uso de movimentações reais, coletadas por 12 meses, de um sistema de monitoramento de veículos de uma empresa de taxi com 102 carros, totalizando mais de 12 milhões de movimentações. A Figura 19 mostra a tela do sistema de monitoramento de uma empresa de taxi.

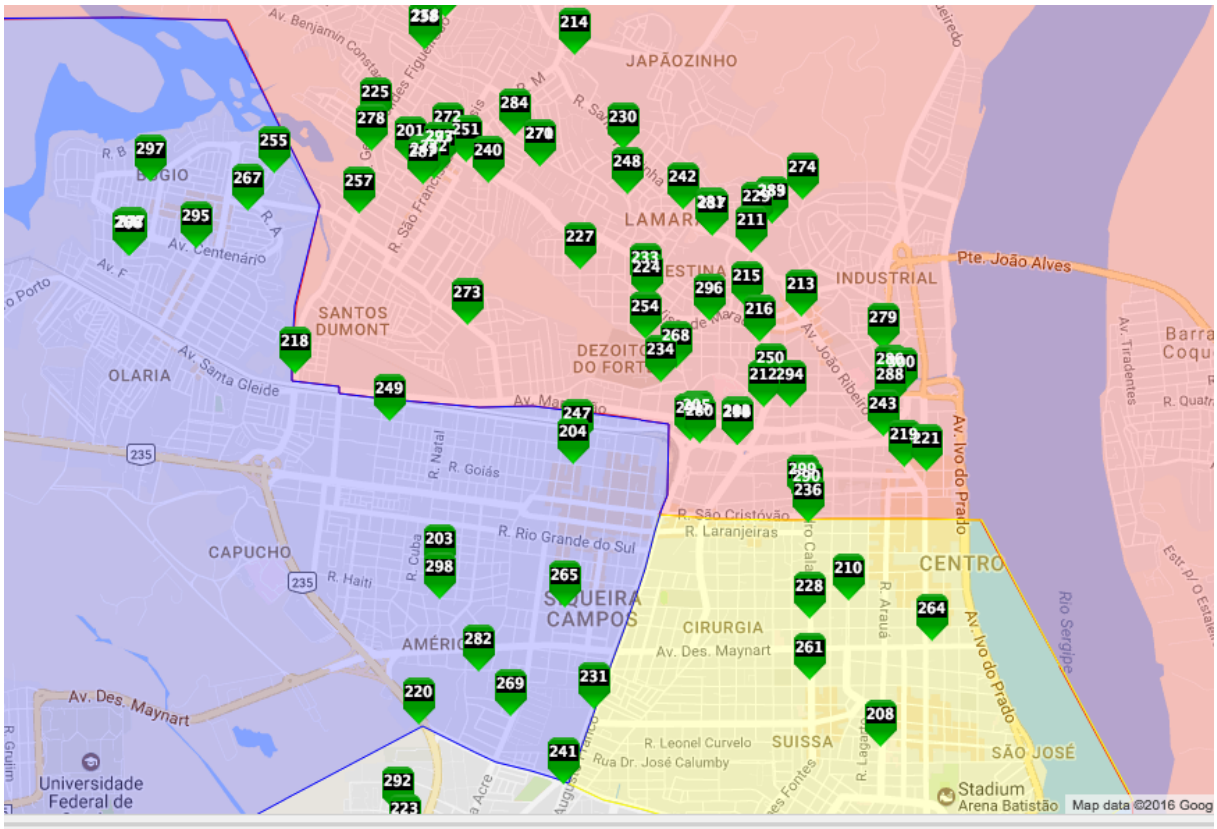


Figura 19 – Tela de monitoramento do Sistema TaxiFast (GMSOLUTIONS, 2017).

A base de movimentação foi extraída, agrupada por veículo e em seguida, armazenada em arquivo e colocada em cada máquina Mininet, para que cada host pudesse simular uma movimentação real. Cada arquivo de movimentação recebeu um número, por exemplo: 1.csv, 2.csv e assim por diante. Então, os *hosts* virtuais realizaram um sorteio entre os arquivos disponíveis para efetuar a leitura das informações de movimentação e transmití-la ao servidor. A Tabela 8 mostra exemplo do conteúdo de um arquivo de movimentação.

Latitude	Longitude	Velocidade
-10.9062183	-37.0619264	0
-10.906164143173685	-37.061813870099655	10.7
-10.904381347197752	-37.0671766923126	21.121
-10.904279005332516	-37.067784265675165	26.259
-10.904165425504841	-37.06846398328098	29.678

Tabela 8 – Linhas do arquivo de movimentação.

Para o segundo grupo dos testes, utilizaram-se as mesmas máquinas virtuais, com o uso de uma outra aplicação desenvolvida pelo autor. Esta aplicação consistiu em criar inúmeras *threads*, para simular os veículos e realizar a conexão com os servidores sem o controle da largura de banda, com objetivo de sobrecarregar a infraestrutura na tentativa de encontrar o limite suportado pela mesma. Foram executadas em 4 testes, sendo com 800 e 1600 veículos utilizando mensagens abertas e, posteriormente, com informações criptografadas.

6.4 Experimentos

Ao iniciar cada experimento, os *hosts* virtuais devem estabelecer a conexão com o servidor nível 0, contudo se todos os "veículos" iniciarem a conexão ao "mesmo tempo", ocasionará uma sobrecarga tanto na máquina cliente quanto no servidor. Pensando nisso, foi implementado na aplicação cliente um *delay* forçado antes de iniciar o processo de conexão pela primeira vez. Cada *host* obtém um número pseudo-randômico entre 0 e 120.000 milisegundos (2 minutos), o qual será usado para dar uma pausa na aplicação para depois iniciar o processo de conexão, gerando assim, uma aleatoriedade inicial. Se por algum

motivo a conexão do *host* cliente cair ou der *timeout*, ela será refeita automaticamente, não considerando mais o tempo de pausa inicial.

```
#!/usr/bin/python
...
class SingleSwitchTopo(Topo):
def build(self, n=2):
    switch = self.addSwitch('s1')
    #Cria todos os hosts e conecta ao switch
    for h in range(n):
        host = self.addHost('h%s' % (h + 1))
        self.addLink(host, switch)
def limit( bw=0.2, cpu=1 ):
    intf = custom( TCIntf, bw=bw )
    myTopo = SingleSwitchTopo(n=2)
    for sched in 'rt', 'cfs':
    if sched == 'rt':
    release = quietRun( 'uname -r' ).strip('\r\n')
    output = quietRun( 'grep CONFIG_RT_GROUP_SCHED /boot/config-%
       s'
    % release )
    if output == '# CONFIG_RT_GROUP_SCHED is not set\n':
        continue
    host = custom( CPULimitedHost, sched=sched, cpu=cpu )
    net = Mininet( topo=myTopo, intf=intf, host=host)
    net.addNAT().configDefault()
    net.start()
    # Para cada host criado, executa a aplicacao de simulacao
    for host in net.hosts:
        if 'h' in host.name:
           host.cmd('java -jar SVVMainTest.jar ' + str(host) + '.
              csv /home/mininet/vanet &')
    CLI( net )
if __name__ == '__main__':
setLogLevel( 'info' )
```

```
31 limit()
```

Listing 6.1 – Script Python para criação dos hosts no mininet.

As máquinas utilizadas no experimento pertenciam a um laboratórios do Instituto Federal de Sergipe - Campus Lagarto e dois notebooks pertencentes ao autor, cuja configurações estão presentes na Tabela 9.

Maquina Cliente			
Sist Operacional.	Qnt.	Processador	Memória
Windows 7	12	AMD Phenom II X2 B57 3.20 GHz	4 GB
Mac OS Sierra	1	2,4 GHz Intel Core i5	8 GB
Maquina Servidora			
Sist Operacional.		Processador	Memória
Ubuntu 16.4	1	2,4 GHz Intel Core i7 GHz	8 GB

Tabela 9 – Configuração dos computadores utilizados no experimento.

Para cada cenário, 50, 100, 200 e 400 veículos, os *hosts* emulados usaram a seguinte distribuição entre as máquinas clientes, como mostra a Tabela 10, tendo como preocupação o poder computacional das máquinas físicas.

Cenário 50		
Máquian Física.	Qnt.	Nr *Hosts* Cada
Windows	2	25
Cenário 100		
Máquina Física.	Qnt.	Nr *Hosts* Cada
Windows	4	25
Cenário 200		
Máquina Física.	Qnt.	Nr *Hosts* Cada
Windows	6	25
Mac OS	1	50
Cenário 400		
Máquina Física.	Qnt.	Nr *Hosts* Cada
Windows	12	25
Mac OS	2	50

Tabela 10 – Configuração dos computadores utilizados no experimento em cada cenário.

Conforme dito anteriormente, foram definidos 36 cenários de testes, sendo 32 levando em consideração quantidades de veículos, largura de banda da comunicação e a criptografia ou não dos dados e 4 utilizando a velocidade total disponível com e sem criptografia dos dados. Os detalhes de cada cenário está apresentado a seguir:

- **Experimento 1(2G), 2 (3G), 3(4G) e 4(5G)**: rodando o teste com 50 veículos sem a utilização da criptografia e foram utilizadas duas máquinas físicas com windows rodando 25 *hosts* cada uma delas;
- **Experimento 5(2G), 6(3G), 7(4G) e 8(5G)**: rodando o teste com 50 veículos com o uso da criptografia e foram utilizadas duas máquinas físicas com windows rodando 25 *hosts* cada uma delas;
- **Experimento 9(2G), 10(3G), 11(4G) e 12(5G)**: rodando o teste com 100 veículos sem o uso da criptografia e foram utilizadas quatro máquinas físicas com windows rodando 25 *hosts* cada uma delas;
- **Experimento 13(2G), 14(3G), 15(4G) e 16(5G)**: rodando o teste com 100 veículos com o uso da criptografia, sendo utilizadas quatro máquinas físicas com windows rodando 25 *hosts* cada uma delas;
- **Experimento 17(2G), 18(3G), 19(4G) e 20(5G)**: rodando o teste com 200 veículos sem o uso da criptografia, sendo utilizadas oito máquinas físicas com windows rodando 25 *hosts* cada uma delas;
- **Experimento 21(2G), 22(3G), 23(4G) e 24(5G)**: rodando o teste com 200 veículos com o uso da criptografia, sendo utilizadas oito máquinas físicas com windows rodando 25 *hosts* cada uma delas;
- **Experimento 25(2G), 26(3G), 27(4G) e 28(5G)**: rodando o teste com 400 veículos sem o uso da criptografia, sendo utilizadas dez máquinas físicas com windows, rodando 30 *hosts* e 2 VMs com mininet com 50 veículos cada, na máquina Mac;
- **Experimento 29(2G), 30(3G), 31(4G) e 32(5G)**: rodando o teste com 400 veículos com o uso da criptografia, sendo utilizadas dez máquinas físicas com windows, rodando 30 *hosts* e 2 VMs com mininet com 50 veículos cada, na máquina Mac;

- **Experimento 33 (Sem limite de velocidade)**: rodando o teste com 800 veículos sem o uso da criptografia, sendo utilizadas dez máquinas físicas com windows rodando 80 *threads* ;
- **Experimento 34 (Sem limite de velocidade)**:rodando o teste com 800 veículos com o uso da criptografia, sendo utilizadas dez máquinas físicas com windows rodando 80 *threads*;
- **Experimento 35 (Sem limite de velocidade)**: rodando o teste com 1600 veículos sem o uso da criptografia, sendo utilizadas dez máquinas físicas com windows rodando 130 *threads* e mais 300 na máquina mac;
- **Experimento 36 (Sem limite de velocidade)**: rodando o teste com 1600 veículos com o uso da criptografia, sendo utilizadas dez máquinas físicas com windows rodando 130 *threads* e mais 300 na máquina mac.

6.5 Resultados

Inicialmente, os dados colhidos referentes ao RTT, foram avaliados quanto à sua normalidade, para verificar se a distribuição de probabilidade associada às amostras podem ser aproximadas em entre os experimentos. Para tal, foi utilizado o *Kolmogorov-Smirnov test* (KS) tendo as seguintes hipóteses:

H0: Os dados seguem uma distribuição normal.

Ha: Os dados não seguem uma distribuição normal.

O resultado do teste KS aplicado sobre as médias dos tempos RTT das requisições de cada *host* indica que nenhuma das amostras apresentaram uma distribuição normal a um nível de significância, comumente utilizado, de 0,05. Rejeitando a hipótese H0 para todos os testes com velocidade limitada, como também com 800 veículos sem limite de velocidade. Já o teste com 1600 veículos, apresentou um *P-Value* acima de 0,05, indicando que a distribuição de probabilidade associada às médias das requisições podem ser aproximadas não rejeitando a hipótese nula, como mostrado na Tabela 11.

Sob a ótica das variações entre os RTT das requisições, podemos observar que nos cenários com 50 e 100 veículos e utilizando a comunicação aberta, não segura, o coeficiente de variação foi baixo e para os testes acima de 200 veículos houve um crecimento no coeficiente. Isso aconteceu devido à maior quantidade de veículos presentes nos testes

Tabela 11 – Resultado do teste de análise de distribuição normal dos dados para cada quantidade de veículos e velocidades de acesso.

Qnt. Veículos	2G	3G	4G	5G	Sem Limite
50	0,0	0,0	0,0	0,0	-
50 encpriptado	0,0	0,0	0,0	0,0	-
100	0,0	0,0	0,0	0,0	-
100 encpriptado	0,0	0,0	0,0	0,0	-
200	0,0	0,0	0,0	0,0	-
200 encpriptado	0,0	0,0	0,0	0,0	-
400	0,0	0,0	0,0	0,0	-
400 encpriptado	0,0	0,0	0,0	0,0	-
800	-	-	-	-	0,195
800 encpriptado	-	-	-	-	0,310
1600	-	-	-	-	0,801
1600 encpriptado	-	-	-	-	0,755

e consequentemente, houve mais mudança de domínios. Em média o tempo gasto para realizar a operação *changeServer* é de 2,5 segundos, conforme mostra a Tabela 12.

Tabela 12 – Resultado do teste de análise dos coeficientes de variação com mensagens abertas.

Qnt. Veículos	2G	3G	4G	5G
50	0.52360	0.5504	0.5751	0.5435
100	0.5340	0.5490	0.5630	0.5560
200	2.1970	2.5930	2.4360	2.3560
400	1.9220	2.1680	2.4270	2.5360

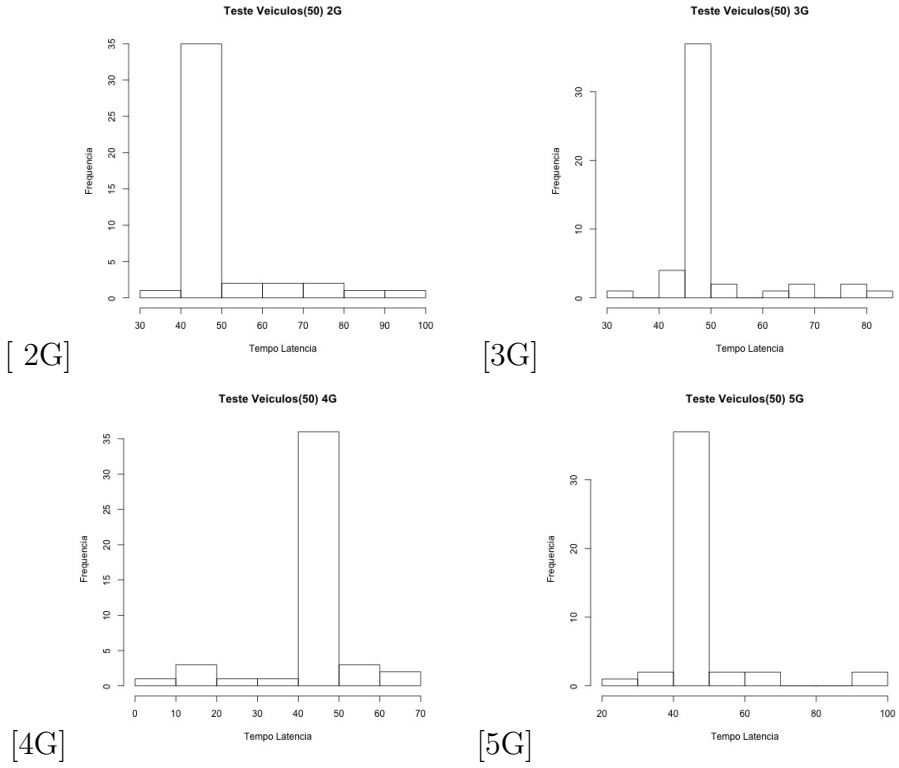
Todavia, o mesmo experimento utilizando mensagens criptografadas, apresentou coeficientes de variação alto, para todos os cenários, nas velocidades de 2G, 3G e 4G, apenas reduzindo no 5G, com exceção do teste com 400 veículos que apresentou uma diminuição a cada aumento de velocidade, porém continuou alto mesmo com o 5G, como mostra a Tabela 13. E vale ressaltar que o tempo medido no envio de mensagens criptografadas, envolve a encriptação na saida do dispositivo, a decriptação na chegada no servidor, a encriptação na resposta do servidor e a decriptação na chegada no dispositivo.

As imagens, contidas na Figura 20 mostram o histógrama de frequência das médias das requisições para cada velocidade definidas para 2G, 3G, 4G e 5G. Para o teste com 50 veículos com mensagens abertas, observou-se que não há uma relação de melhora nos tempos médios à medida que aumenta a velocidade de 2G para 5G. A concentração dos tempos permanecem na mesma referência, entre 40 e 50 milisegundos, sendo considerado ótimo se comparado com as especificações apresentadas por Papadimitratos et al. (2008).

Tabela 13 – Resultado do teste de análise dos coeficientes de variação com mensagens criptografadas.

Qnt. Veículos	2G	3G	4G	5G
50	2.2270	2.0500	1.8631	0.6435
100	2.2520	2.1540	2.5100	0.5560
200	2.0670	1.2260	0.7901	0.6220
400	1.8570	1.6830	1.8230	1.5800

Figura 20 – Análise de Histograma de Frequência do Teste com 50 veículos com comunicação aberta.

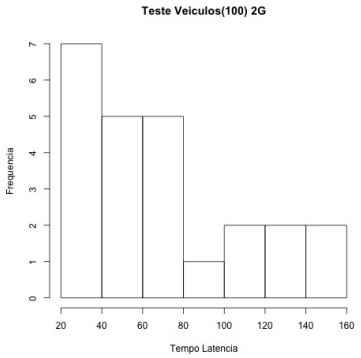


O cenário com 100 veículos apresentou gráficos semelhantes aos de 50, ou seja o tempo médio das requisições não variou de acordo com o aumento da velocidade indicando que a plataforma suporta bem estas quantidades de veículos, conforme mostra a Figura 21.

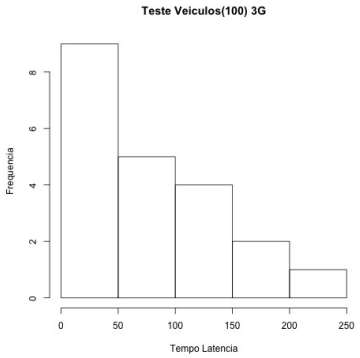
Entretanto, os testes com o cenário de 200 veículos apresentaram uma variação com a diferença de velocidades, de acordo com a Figura 22, o teste com a velocidade 2G apresentou maior concentração entre o intervalo de 0 e 50 milisegundos. Já Com a velocidade de 3G, a maior concentração ficou entre 140 e 160 milisegundos, dando sinais que o aumento na concorrência das requisições comeca a prejudicar o desempenho da plataforma na infraestrutura montada.

No experimento com 400 veículos, a velocidade 2G apresentou o pior resultado em relação às outras velocidades, contendo, de maneira significativa, médias entre 600 e

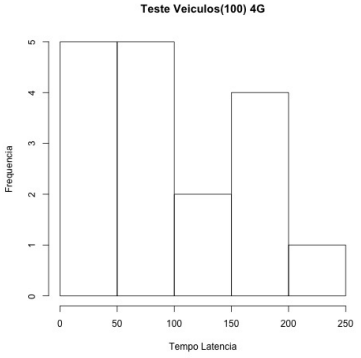
Figura 21 – Análise de Histograma de Frequência do Teste com 100 veículos com comunicação aberta.



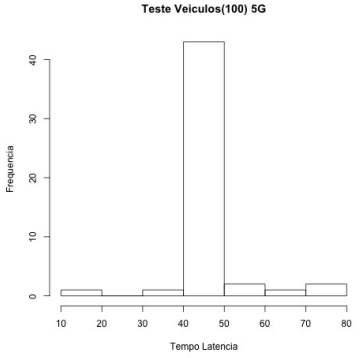
[2G]



[3G]

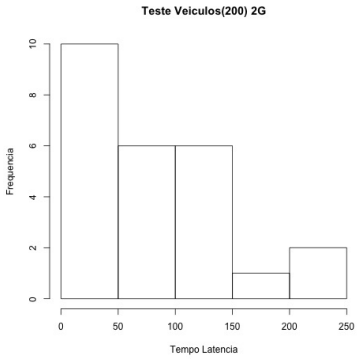


[4G]

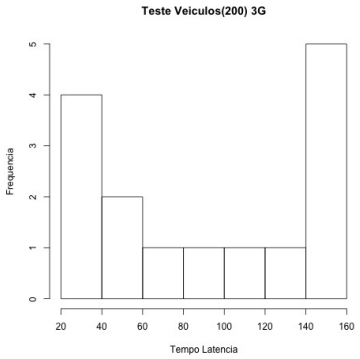


[5G]

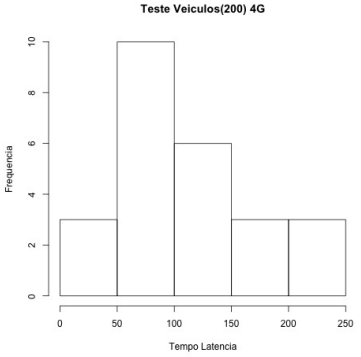
Figura 22 – Análise de Histograma de Frequência do Teste com 200 veículos com comunicação aberta.



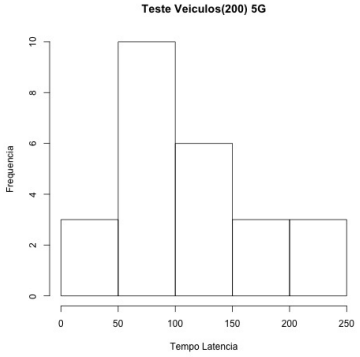
[2G]



[3G]



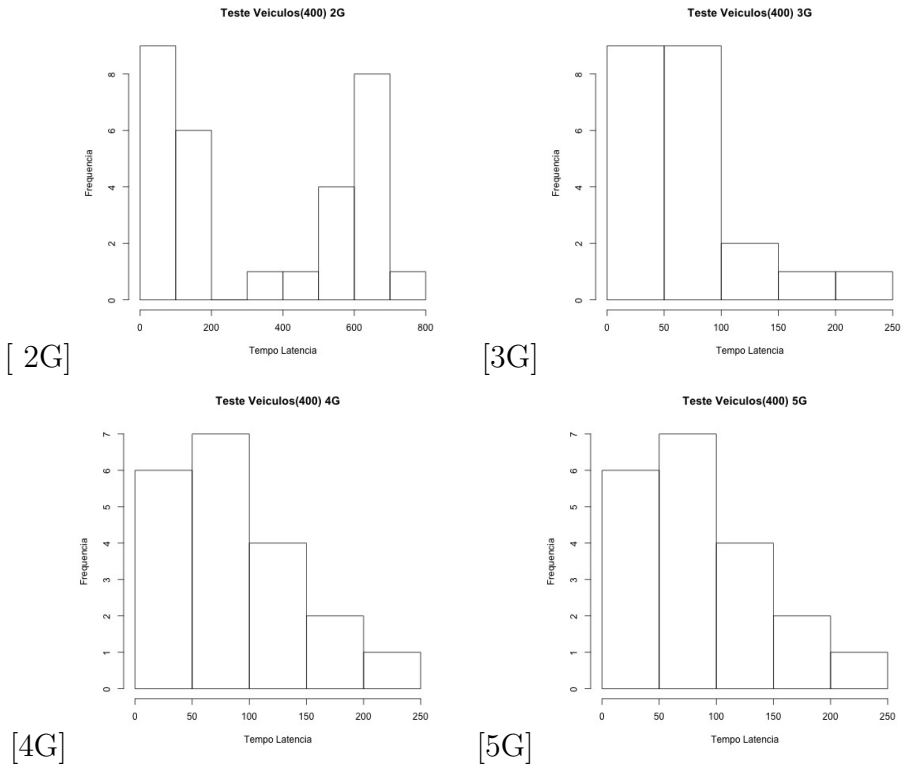
[4G]



[5G]

700 milisegundos, tornando inviável o uso da velocidade 2G com a plataforma I9VANET. O mesmo não ocorreu com as velocidades de 3G, 4G e 5G, que apresentaram grande concentração entre os tempos de 0 à 150 milisegundos como mostra a Figura 23, indicando que para estas velocidades a infraestrutura montada pode ser utilizada.

Figura 23 – Análise de Histograma de Frequência do Teste com 400 veículos com comunicação aberta.



Nas Figuras 24, 25 e 26, é feita uma análise sobre os tempos mínimos, médios e máximos das médias dos RTTs das requisições com mensagens abertas e encriptadas, extraídas de cada cenário. Fica evidente que nos cenários de 50, 100 e 200 veículos simultâneos, os tempos mínimos, médios e máximos apresentam pequena variação, entretanto os tempos médios e máximos no cenário com 400 veículos, com velocidade de 2G, há um aumento considerável em relação às outras velocidades, como mostra as Figuras 25 e 26, indicando que essa quantidade de veículos já não é indicada com a velocidade do 2G.

No segundo grupo de experimento, foram avaliadas as quantidades de 800 e 1600 veículos sem limite da conexão. Conforme Figura 27, fica evidente o aumento nos tempos médios de RTT das requisições com mensagens criptografadas, Figura 27.B, sobre as requisições com conteúdo aberto, Figura 27.A. Entretanto, o crescimento não foi significativo sendo a diferença da maior concentração dos tempos entre 10 e 18 milisegundos, para o teste

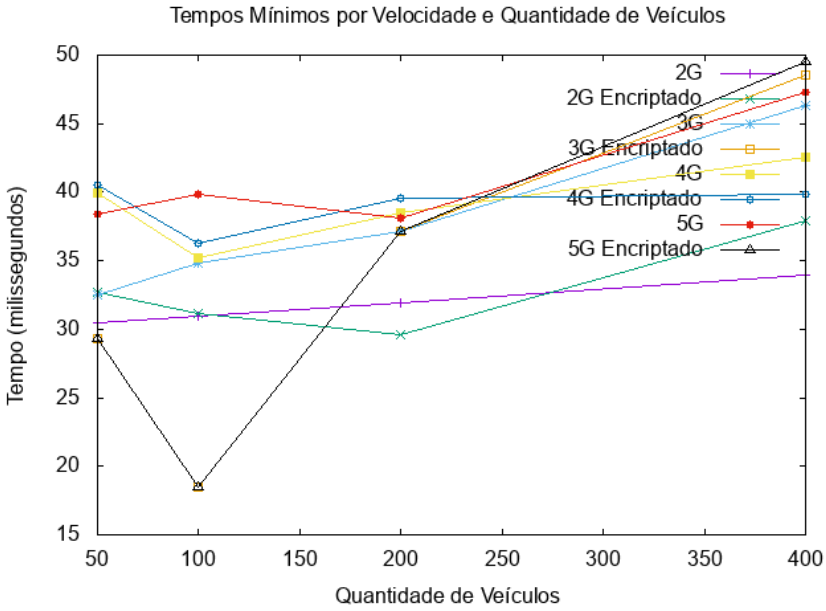


Figura 24 – Gráfico comparativo dos tempos mínimos entres os cenários.

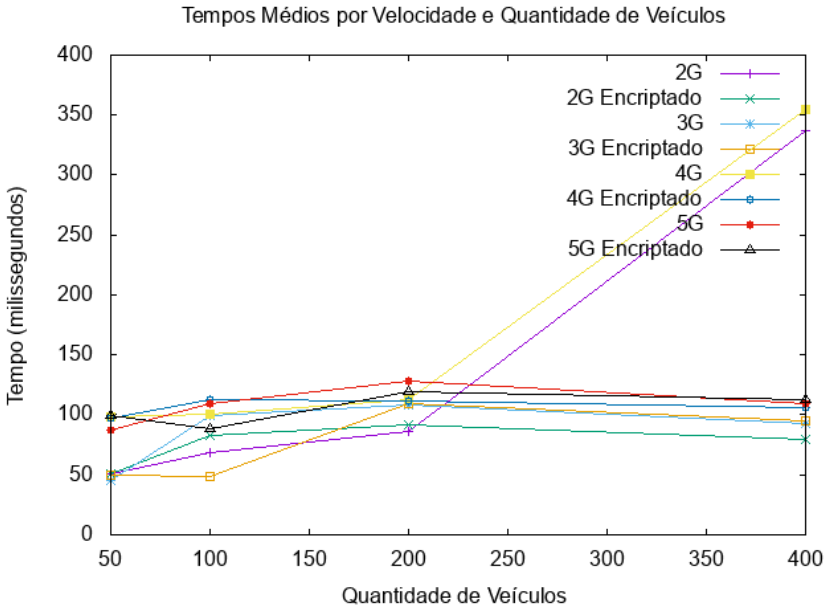


Figura 25 – Gráfico comparativo dos tempos médios entres os cenários.

com mensagens abertas, e 21 e 22 milisegundos, para o teste com conteúdo criptografado, gerando uma diferença média de 8 milisegundos indicando que em ambos os casos, o tempo obtido é ótimo para o uso em redes veiculares, todavia o tempo mais rigoroso é de 100 milisegundos de acordo com Papadimitratos et al. (2008).

Para o teste com 1600 veículos, houve um aumento considerável dos tempos médios das requisições como mostra a Figura 28. A concentração dos tempos médios na Figura 28.A, ficou entre 16 e 20 milisegundos, já no teste com mensagens criptografadas, ficou

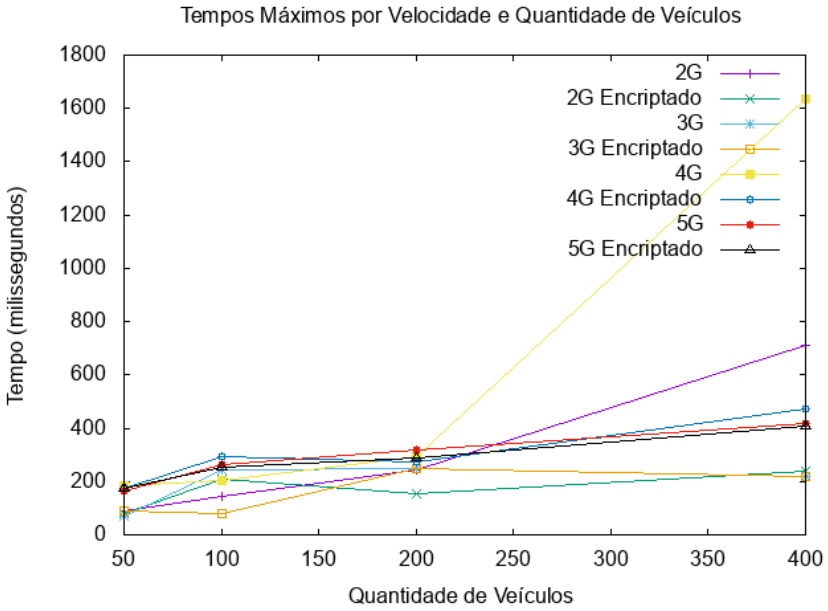
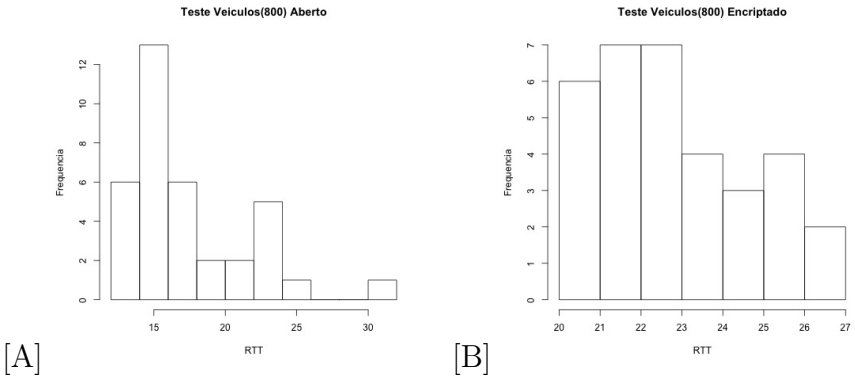


Figura 26 – Gráfico comparativo dos tempos máximos entres os cenários.

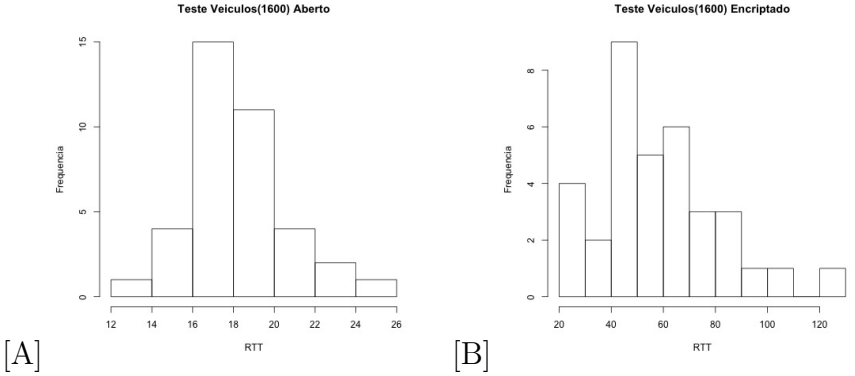
Figura 27 – Comparação dos gráficos de histograma para os RTTs com 800 veículos.



[A] [B]

em 40 e 90 milisegundos indicando que o tempo de processamento, para a o processo de criptografia, afetou os tempos de resposta em até 450%, conforme mostrado na Figura 28.B.

Figura 28 – Comparação dos gráficos de histograma para os RTTs com 1600 veículos.



[A] [B]

6.5.1 Taxa de Transferência

Os veículos enviaram, a cada 10 segundos, os dados com localização, direção e velocidade, estas infomações variaram entre 212 e 252 bytes e a Tabela 14 mostra as velocidades de transmissão utilizadas por cada veículo nos experimentos e o gráfico da Figura 29 exibe a capacidade estimada ocupada por cada teste executado com limites de conexões. Apesar de ter sido definido um tempo fixo, a depender da aplicação as requisições podem ser realizadas à medida que ocorre algum evento previso, como por exemplo ao encontrar um deformidade na rodovia, este alerta deve ser enviado no momento em que foi detectado.

Tabela 14 – Velocidade de cada link por veículo definidos de acordo com Li et al. (2009).

Link	Lagura de Banda(kbps)	Largura de Banda (Mbps)
2G	400	0,4
3G	2.000	2
4G	100.000	100
5G	10.000.000	1.000

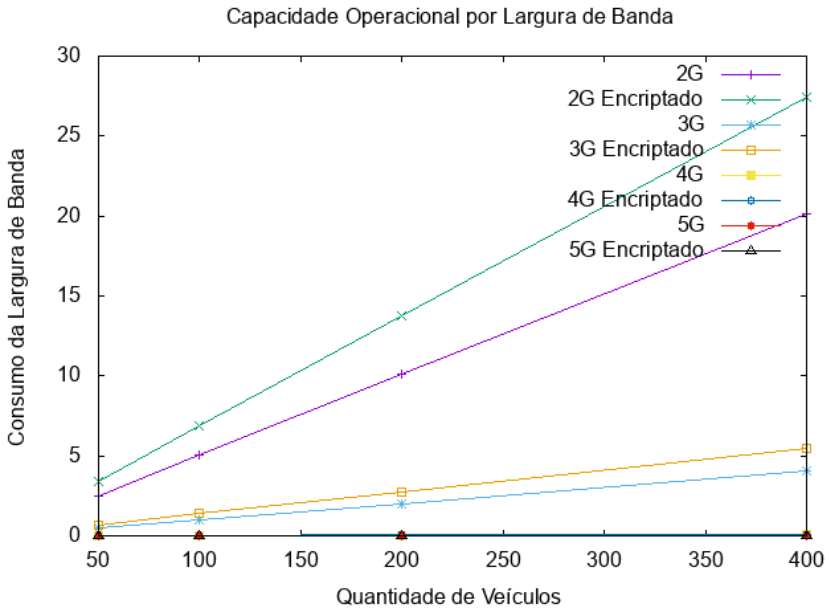


Figura 29 – Comparativo da capacidade de cada *link* pela quantidade de veículos.

6.5.2 Processamento

Foi avaliado o nível de processamento dos servidores, sendo que o servidor nível 0 chegou a 100% de processamento nos primeiros 2 minutos, no teste com 800 veículos, e depois reduziu para 20%, onde se manteve, conforme Figura 30. Isso ocorreu devido a todas as conexões iniciais serem feitas para este servidor, como também quando há necessidade de mudança de domínio, o servidor nível 0 é requisitado de acordo com a hierarquia estabelecida.

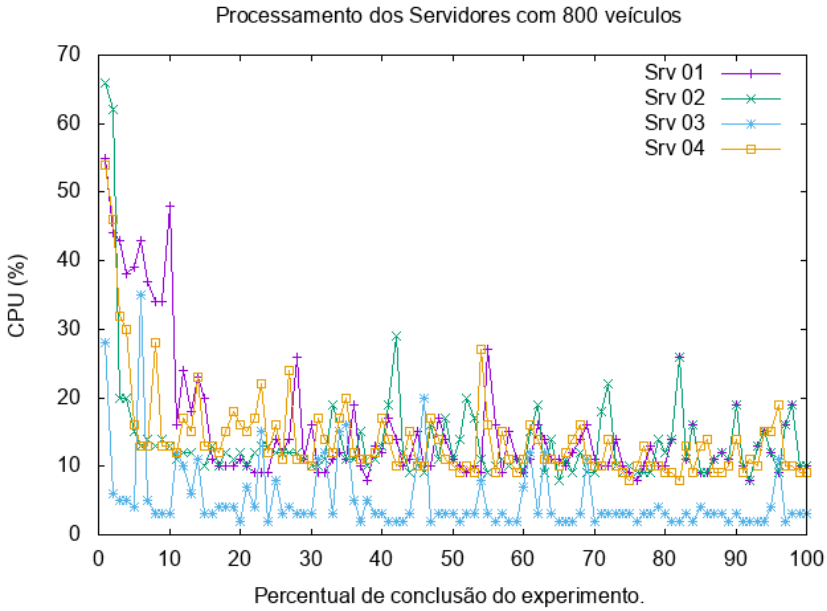


Figura 30 – Nível de processamento dos servidores para o teste com 800 veículos com conteúdo criptografado.

No teste com 1600 veículos, utilizando conteúdo criptografado, o processamento chegou a 100% no sevidor nível 0 e permanceu alto, entre 75% e 100%, durante todo o teste, como mostra a Figura 31.

Durante os experimentos foram analisadas as perdas em cada dispositivo por meio de dois contadores. O primeiro foi utilizado para contabilizar a quantidade de requisições realizadas e outro para apurar a quantidade de respostas recebidas. Como mostrado na Figura 32.A, a maior perda ocorre nas velocidades de 2G utilizando encriptação atingindo o valor a cima de 25% de perda. Contudo, para o teste com 400 veículos, a perda ficou abaixo de 15%. Em amos os casos a perda é considerada alta por estar acima de 10% como foi definido por Cechin (2008). Já as perdas para os experimentos sem limite de conexão, apresentaram valor abaixo de 1% para o teste com 800 veículos e 2.06% para o teste com

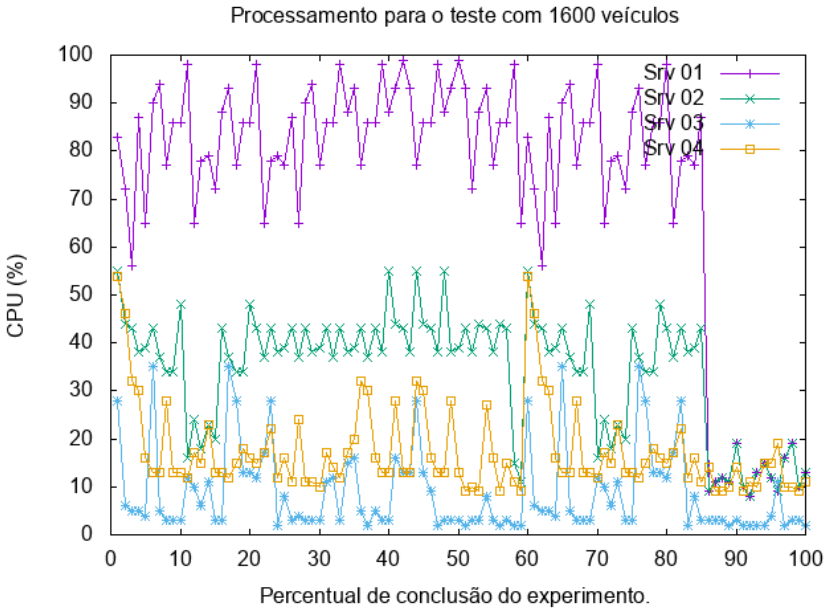
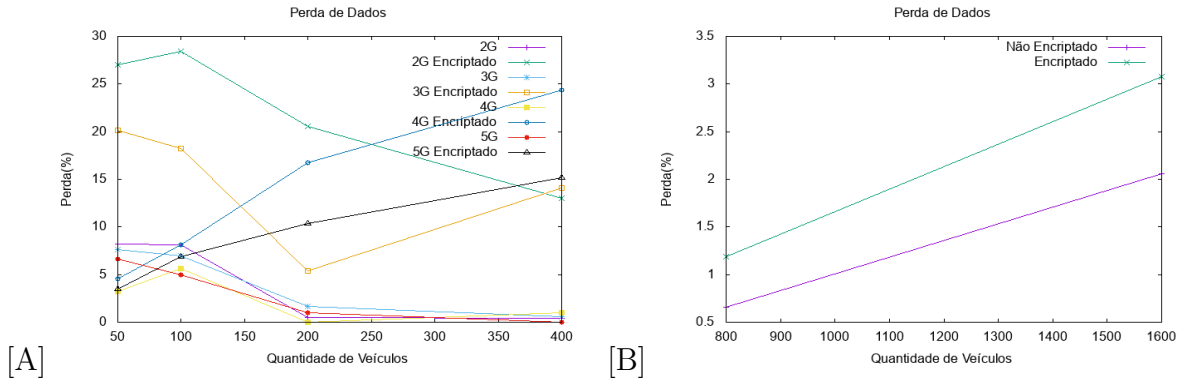


Figura 31 – Nível de processamento dos servidores para o teste com 1600 veículos com conteúdo criptografado.

Figura 32 – Percentual de perda para cada experimento realizado.



1600 veículos e utilizando mensagens criptografadas, ambos ficaram mais altos em relação à comunicação aberta, como mostra a Figura 32.B. Sendo assim, sob a ótica da eficácia, o experimento utilizando a velocidade do modelo 2G, mostrou-se não ser eficaz devido ao alto índice de perda.

7 Conclusão e Trabalhos Futuros

Nesta dissertação abordou-se um tema que nos últimos anos tem sido foco de estudo por pesquisadores em todo o mundo, as redes veiculares. Sendo assim, o presente estudo definiu um modelo de arquitetura de *software* para redes veiculares em nuvem, denominado I9VANETs, como também uma avaliação da plataforma criada sobre a arquitetura, com intuito de medir a sua eficácia e eficiência, visando atender às demandas e desafios relacionados às VANETs.

7.1 Contribuições

Como principais contribuições do trabalho temos: o modelo proposto permite montar uma rede veicular em nuvem e realizar todo o gerenciamento e comunicação de maneira virtual, permitindo criar ambientes flexíveis capazes de oferecer o gerenciamento de uma rede veicular como serviço (VaaS); Além disso, a plataforma possibilita tornar os equipamentos OBUs e RSUs mais simples e baratos e atua sobre alguns dos principais desafios relacionados à VANET tais como: interferências na comunicação, alta mobilidade, baixa e alta densidade de veículos, segurança na comunicação entre os nós; Também foi construído uma plataforma web de simulação, facilitando a utilização em ambiente reais e de maneira distribuída; Criação de um protótipo para Detecção e Alerta de Congestionamento em Cruzamentos Semaforizados utilizando Lógica Fuzzy.

A arquitetura I9VANET foi definida de maneira modular, sendo dividida em 6 módulos são eles: Applications, Server Management *Cloud, Routing between Nodes, Infra-Cloud Communication e Vehicle-Cloud Comunication*, e cada módulo foi pensado de forma que seu funcionamento interno não interfera nos outros. Assim, é possível substituir a implementação de um módulo sem afetar a plataforma.

Contudo, apesar da plataforma I9VANET realizar o gerenciamento virtual e atuar sobre alguns dos desafios relacionados às redes veiculares, é importante salientar que há peculiaridades referentes ao modelo de rede veicular ad hoc tradicional que apenas a comunicação direta, veículo a veículo, pode resolver, como por exemplo aplicações pré-colisão, alerta de ultrapassagem, entre outros. Entretanto, nada impede que os modelos em núvem e ad-hoc trabalhem em conjunto, o então chamado modo híbrido, na tentativa de aproveitar o melhor dos dois mundos, aproveitando a escalabilidade e segurança da

nuvem e a independência de infraestrutura para comunicação, o acesso direto a sensores do veiculo e processamento local realizados em redes ad-hoc tradicionais.

Um fato importante que deve ser comentado, é a dependência da plataforma I9VANET em relação ao dados de localização dos veículos. A falta de precisão do GPS é um fator limitante para o uso da plataforma por aplicações que reinvidiquem este critério. Todavia, uma pesquisa recentemente publicada, apresentou uma técnica que potencializa os dados de geolocalização por meio da utilização sensor de inércia acoplado ao dispositivo possibilitando precisão de uma polegada (CHEN; ZHAO; FARRELL, 2016). Tal descoberta abre ainda mais a possibilidade de utilizar VANET e servidores em nuvem.

O processo de avaliação aplicado à plataforma I9VANET, teve como público alvo desenvolvedores de soluções com foco em Sistemas Inteligentes de Transportes. Cujo objetivo foi avaliar a capacidade de processamento e medir o tempo de latência das comunicações. Para isso, foram realizados 36 experimentos divididos em dois grupos. O primeiro, totalizando 34 testes, utilizou como base os limites das velocidades da telefonia móvel, definidas na literatura, para os modelos 2G, 3G, 4G e 5G, com uso de transmissão de dados abertos e criptografados e utilizando quantidades pré-definidas de veículos com o seguintes valores: 50, 100, 200 e 400. O segundo grupo, consistiu em avaliar a capacidade operacional da plataforma independentemente da velocidade de acesso dos veículos, foram realizados 4 experimentos, com 800 e 1600 veículos e utilizando dados criptografados e não criptografados.

Foi definido como hipótese a seguinte sentença: é possível criar uma plataforma aberta, flexível e extensível capaz de permitir o gerenciamento de redes veiculares como serviço (VaaS) por meio de uma solução em nuvem, sendo capaz de atender aos requisitos mínimos de tempo para a maioria das aplicações voltadas para redes veiculares? Cada tipo de aplicação voltada para redes veiculares possui requisitos que devem ser considerados como mostrado na Tabela 7, e a plataforma I9VANET apresentou tempos médios adequados para todos os tipos de aplicações, com exceção da velocidade definida pelo 2G, conforme apresentado na Figura 25.

Em relação ao nível de processamento dos servidores, o servidor nível 0, raiz da hierarquia, é o responsável pelas primeiras conexões feitas pelos veículos, como também durante a mudança de domínios. Pensando nisso, para uma melhor organização e diminuição de carga em um único servidor, é interessante que os domínios vizinhos fiquem, ao máximo, no mesmo nível de hierarquia tendo o mesmo servidor pai. Assim, essa sub-árvore

responderia de maneira mais eficiente ao comando *changeServer* e diminuiria a pressão sobre o servidor raiz.

Uma outra possibilidade é que o servidor nível 0 seja utilizado apenas para coordenar a estrutura, retirando de si o papel de gerenciar os veículos. Sendo assim, ele teria o papel de receber as primeiras conexões e encaminhá-las para o servidor correspondente, o mesmo a seria aplicado à operação *changeServer*' no momento da troca de domínios.

Os testes aplicados à plataforma com 800 e 1600 veículos, indicaram que o limite para organização proposta está em 1600 veículos e que mais veículos conectados a esta estrutura poderá comprometer o tempo de resposta, sendo necessário aumentar o número de servidores para além dos 4 que foram utilizados no experimento, ou até mesmo, melhorar a capacidade de processamento dos mesmos.

Portanto, após o exposto, os objetivos definidos para este trabalho foram atingidos por completo, visto que foi proposta a criação de uma arquitetura de *software* flexível e extensível com capacidade de gerenciar nós de uma rede VANET, de maneira virtualizada e cada objetivo específico foi prontamente atendido, como se segue:

- Foi elaboradox uma arquitetura de *software* com os recursos da extensibilidade, flexibilidade e escalabilidade, permitindo que um desenvolvedor possa construí-la em qualquer linguagem de programação;
- O autor desenvolveu uma plataforma utilizando a linguagem Java, implementando uma versão de cada módulo especificado na arquitetura. Esta plataforma está disponível para futuras contribuições no GitHub [1];
- A partir da plataforma desenvolvida, foi possível realizar uma bateria de experimentos, de onde foram extraídos os dados, de acordo com as métricas estabelecidas, para avaliação de desempenho e capacidade operacional.

Analisando o aspecto da flexibilidade da plataforma I9VANET, podemos afirmar que seu uso pode ser adotado no desenvolvimento de soluções com foco ITS. Como foi visto no presente trabalho, há seis áreas avançadas de gestão em Sistema Inteligente de Transporte, Sistema Avançado de Gestão de Tráfego, Sistema Avançado de Informações para Viajantes, Sistema Avançado de Transporte Público, Sistema de Operação de Veículos Comerciais, Sistema Avançado de Controle de Veículos e Sistema de Coleta Eletrônica de

[1] https://geoleite@github.com/geoleite/Mestrado.git

Pedágio. Cada um deles podem utilizar a plataforma I9VANET como base, uma vez que a principal característica desses sistemas é a utilização de uma solução centralizada.

7.2 Publicações

Durante o desenvolvimento desta pesquisa foi possível participar de alguns eventos que contribuíram para o enriquecimento e estimularam todo o processo de pesquisa. Segue alguns eventos onde foram realizadas as publicações:

- XVI ESCOLA REGIONAL DE COMPUTAÇÃO BAHIA - ALAGOAS - SERGIPE (ERBASE) - 2016: **Modelo de uma Arquitetura de Software para Virtualização de Redes Veiculares.**
- ConectaIF - 2016: **Identificando Níveis de Congestionamento em Cruzamentos com Sinalização Semafórica, Utilizando Lógica Fuzzy e Rede Veicular.**
- ConectaIF - 2016: **Um Comparativo entre Métodos de Comunicação em Sistemas Embarcados**
- 9th Euro American Conference on Telematics and Information Systems (EATIS) 2016 - (Qualis B3): **Uma Proposta de Arquitetura Orientada a Serviços co Foco na Interoperabilidade entre sensores para ITS em Cidades Inteligentes**
- The 26th IEEE International Conference on Enabling Techologies: Infrastructure for Collaborative Enterprises (WETICE) 2017 - (Qualis B1): **A Platform for Vehicular Networks in the Cloud to Applications in Intelligent Transportation Systems**

7.3 Trabalhos Futuros

Os resultados e contribuições apresentados nesta dissertação criam oportunidades para trabalhos futuros: por meio de mudança na organização dos servidores, visando uma melhor distribuiçào dos veículos e diminuindo a carga com a operação *ChangeServer*; criação de novos algoritmos de roteamentos; novos protocolos de comunicação; novas regras de segurança; Alerta ao motorista para seguir um modelo de direção ecológica; Controle de passagem livre para veícuos de urgência e emergência, possibilitando avisar aos nós da rodovia que um veículo está pedindo passagem, como tambem informar ao semáforo que

deve ficar aberto até a passagem do veículo, permitindo diminuir o tempo de chegada ao destino; Construção de uma plataforma para gerenciamento de Nuvens Veiculares Locais Virtuais (NVLV), onde os servidores em nuvem, teriam a responsabilidade de distribuir os processos entre os veículos físicos, não importando a distância real entre os nós da rede veicular.

Referências

AL-KAHTANI, M. S. Survey on security attacks in vehicular ad hoc networks (vanets). In: IEEE. *Signal Processing and Communication Systems (ICSPCS), 2012 6th International Conference on*. [S.l.], 2012. p. 1–9. Citado 2 vezes nas páginas 47 e 48.

AVELAR, E. et al. Interoperability issues on heterogeneous wireless communication for smart cities. *Computer Communications*, Elsevier, v. 58, p. 4–15, 2015. Citado na página 46.

BALL, R.; DULAY, N. Enhancing traffic intersection control with intelligent objects. *Urban Internet of Things Towards Programmable Realtime Cities*, 2010. Citado na página 18.

BARBA, C. T.; AGUIRRE, K. O.; IGARTUA, M. A. Performance evaluation of a hybrid sensor and vehicular network to improve road safety. In: ACM. *Proceedings of the 7th ACM workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*. [S.l.], 2010. p. 71–78. Citado na página 31.

BIRK, W.; OSIPOV, E.; ELIASSON, J. iroad—cooperative road infrastructure systems for driver support. In: *Proceedings of the 16th ITS World Congress*. [S.l.: s.n.], 2009. Citado na página 32.

BLUM, J.; ESKANDARIAN, A.; HOFFMAN, L. Mobility management in ivc networks. In: IEEE. *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*. [S.l.], 2003. p. 150–155. Citado na página 53.

BOUKERCHE, A. Performance evaluation of routing protocols for ad hoc wireless networks. *Mobile Networks and Applications*, Springer-Verlag New York, Inc., v. 9, n. 4, p. 333–342, 2004. Citado na página 51.

BRIESEMEISTER, L.; SCHAFERS, L.; HOMMEL, G. Disseminating messages among highly mobile hosts based on inter-vehicle communication. In: IEEE. *Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE*. [S.l.], 2000. p. 522–527. Citado na página 54.

BUBENIKOVA, E.; DURECH, J.; FRANEKOVA, M. Security solutions of intelligent transportation system's applications with using vanet networks. In: IEEE. *Control Conference (ICCC), 2014 15th International Carpathian*. [S.l.], 2014. p. 63–68. Citado na página 43.

CARVALHO, C. H. R. d. et al. Mobilidade urbana e posse de veículos: análise da pnad 2009. Instituto de Pesquisa Econômica Aplicada (Ipea), 2010. Citado na página 19.

CAVALCANTI, S. R. *Veer: Um algoritmo de seleção de pares em redes ad hoc veiculares*. Tese (Doutorado) — UNIVERSIDADE FEDERAL DO RIO DE JANEIRO, 2008. Citado 2 vezes nas páginas 20 e 21.

CECHIN, S. L. *Métricas para Avaliação de Desempenho em Redes QoS sobre IP*. Tese (Doutorado) — UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL, 2008. Citado na página 90.

CHEN, W. et al. Dynamic local peer group organizations for vehicle communications. In: IEEE. *Mobile and Ubiquitous Systems-Workshops, 2006. 3rd Annual International Conference on*. [S.l.], 2006. p. 1–8. Citado na página 21.

CHEN, Y.; ZHAO, S.; FARRELL, J. A. Computationally efficient carrier integer ambiguity resolution in multiepoch gps/ins: a common-position-shift approach. *IEEE Transactions on Control Systems Technology*, IEEE, v. 24, n. 5, p. 1541–1556, 2016. Citado na página 93.

COMI, A. et al. An evolutionary approach for cloud learning agents in multi-cloud distributed contexts. In: IEEE. *Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2015 IEEE 24th International Conference on*. [S.l.], 2015. p. 99–104. Citado 2 vezes nas páginas 24 e 25.

COULOURIS, G. et al. *Sistemas Distribuídos-: Conceitos e Projeto*. [S.l.]: Bookman Editora, 2013. Citado 4 vezes nas páginas 35, 36, 37 e 38.

CRISTIAN, F. Reaching agreement on processor-group membrship in synchronous distributed systems. *Distributed Computing*, Springer, v. 4, n. 4, p. 175–187, 1991. Citado na página 38.

DANTAS, A. *TRÂNSITO: O JOGO DO CAOS*. 2011. Disponível em: ⟨http://www.autoentusiastasclassic.com.br/2011/05/transito-o-jogo-do-caos.html⟩. Citado 3 vezes nas páginas 9, 19 e 56.

DAS, B.; BHARGHAVAN, V. Routing in ad-hoc networks using minimum connected dominating sets. In: IEEE. *Communications, 1997. ICC'97 Montreal, Towards the Knowledge Millennium. 1997 IEEE International Conference on*. [S.l.], 1997. v. 1, p. 376–380. Citado na página 53.

DIB, R. P. E. et al. A systematic review of the interventions to promote the wearing of hearing protection. *São Paulo Medical Journal*, SciELO Brasil, v. 125, n. 6, p. 359–361, 2007. Citado na página 19.

DURRESI, M.; DURRESI, A.; BAROLLI, L. Emergency broadcast protocol for inter-vehicle communications. In: IEEE. *11th International Conference on Parallel and Distributed Systems (ICPADS'05)*. [S.l.], 2005. v. 2, p. 402–406. Citado na página 54.

ELTOWEISSY, M.; OLARIU, S.; YOUNIS, M. Towards autonomous vehicular clouds. In: SPRINGER. *International Conference on Ad Hoc Networks*. [S.l.], 2010. p. 1–16. Citado 2 vezes nas páginas 22 e 25.

ENGOULOU, R. G. et al. Vanet security surveys. *Computer Communications*, Elsevier, v. 44, p. 1–13, 2014. Citado na página 45.

FALCHETTI, A.; AZURDIA-MEZA, C.; CESPEDES, S. Vehicular cloud computing in the dawn of 5g. In: IEEE. *2015 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*. [S.l.], 2015. p. 301–305. Citado 4 vezes nas páginas 22, 23, 25 e 27.

FISCHER, M. J.; LYNCH, N. A.; PATERSON, M. S. Impossibility of distributed consensus with one faulty process. *Journal of the ACM (JACM)*, ACM, v. 32, n. 2, p. 374–382, 1985. Citado na página 38.

FÜSSLER, H. et al. Mobicom poster: location-based routing for vehicular ad-hoc networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, ACM, v. 7, n. 1, p. 47–49, 2003. Citado na página 52.

GERLA, M. Vehicular cloud computing. In: IEEE. *Ad Hoc Networking Workshop (Med-Hoc-Net), 2012 The 11th Annual Mediterranean*. [S.l.], 2012. p. 152–155. Citado 2 vezes nas páginas 23 e 25.

GMSOLUTIONS. *TaxiFast*. 2017. Disponível em: ⟨https://www.taxifast.taxi.br⟩. Citado 2 vezes nas páginas 9 e 77.

GORENDER, S.; MACÊDO, R. Um modelo para tolerância a falhas em sistemas distribuídos com qos. *Anais do Simpósio Brasileiro de Redes de Computadores, SBRC 2002*, p. 277–292, 2002. Citado 2 vezes nas páginas 37 e 38.

GRAY, N. A. Comparison of web services, java-rmi, and corba service implementations. In: *Proceedings of the 5th Australasian Workshop on Software and System Architectures at ASWEC*. [S.l.: s.n.], 2004. p. 52–63. Citado 4 vezes nas páginas 9, 11, 40 e 41.

GUPTE, S.; YOUNIS, M. Vehicular networking for intelligent and autonomous traffic management. In: IEEE. *Communications (ICC), 2012 IEEE International Conference on*. [S.l.], 2012. p. 5306–5310. Citado na página 20.

HADALLER, D. et al. Vehicular opportunistic communication under the microscope. In: ACM. *Proceedings of the 5th international conference on Mobile systems, applications and services*. [S.l.], 2007. p. 206–219. Citado na página 21.

HAJJI, T.; BARGAOUI, H. Design of a vanet testbed based on cloud computing. *European Conference on Networks and Communications*, 2015. Citado 2 vezes nas páginas 22 e 25.

HALL, G. B.; LEAHY, M. G. *Open source approaches in spatial data handling*. [S.l.]: Springer, 2008. v. 2. Citado na página 70.

HUSSAIN, R. et al. Rethinking vehicular communications: Merging vanet with cloud computing. In: IEEE. *Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on*. [S.l.], 2012. p. 606–609. Citado 3 vezes nas páginas 22, 23 e 25.

ICE, R. et al. *REGIONAL ITS ARCHITECTURE GUIDANCE: DEVELOPING, USING, AND MAINTAINING AN ITS ARCHITECTURE FOR YOUR REGION*. [S.l.], 2001. Citado na página 18.

ISAAC, J. T.; ZEADALLY, S.; CAMARA, J. S. Security attacks and solutions for vehicular ad hoc networks. *IET communications*, IET, v. 4, n. 7, p. 894–903, 2010. Citado 2 vezes nas páginas 48 e 49.

JACQUET, P. et al. Optimized link state routing protocol for ad hoc networks. In: IEEE. *Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International*. [S.l.], 2001. p. 62–68. Citado na página 51.

JAKUBIAK, J.; KOUCHERYAVY, Y. State of the art and research challenges for vanets. In: IEEE. *Consumer communications and networking conference, 2008. CCNC 2008. 5th IEEE*. [S.l.], 2008. p. 912–916. Citado na página 43.

KARP, B.; KUNG, H.-T. Gpsr: Greedy perimeter stateless routing for wireless networks. In: ACM. *Proceedings of the 6th annual international conference on Mobile computing and networking*. [S.l.], 2000. p. 243–254. Citado na página 52.

KIHL, M.; SICHITIU, M.; JOSHI, H. Design and evaluation of two geocast protocols for vehicular ad-hoc networks. *Journal of Internet Engineering*, Klidarithmos Press, 2008. Citado na página 54.

LAMPORT, L.; SHOSTAK, R.; PEASE, M. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, ACM, v. 4, n. 3, p. 382–401, 1982. Citado na página 38.

LEE, E. et al. Vehicular cloud networking: architecture and design principles. *IEEE Communications Magazine*, IEEE, v. 52, n. 2, p. 148–155, 2014. Citado 3 vezes nas páginas 9, 23 e 25.

LEONTIADIS, I.; MASCOLO, C. Geopps: Geographical opportunistic routing for vehicular networks. In: IEEE. *2007 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*. [S.l.], 2007. p. 1–6. Citado na página 52.

LI, C. Travel information service system for public travel based on soa. In: IEEE. *Service Operations and Logistics and Informatics (SOLI), 2010 IEEE International Conference on*. [S.l.], 2010. p. 321–324. Citado na página 18.

LI, F.; WANG, Y. Routing in vehicular ad hoc networks: A survey. *IEEE Vehicular technology magazine*, IEEE, v. 2, n. 2, p. 12–22, 2007. Citado na página 52.

LI, L.; LIU, Y.-A.; TANG, B.-H. Snms: an intelligent transportation system network architecture based on wsn and p2p network. *The journal of China universities of posts and telecommunications*, Elsevier, v. 14, n. 1, p. 65–70, 2007. Citado na página 32.

LI, X. et al. The future of mobile wireless communication networks. In: IEEE. *Communication Software and Networks, 2009. ICCSN'09. International Conference on*. [S.l.], 2009. p. 554–557. Citado 2 vezes nas páginas 11 e 89.

LIANG, W. et al. Vehicular ad hoc networks: architectures, research issues, methodologies, challenges, and trends. *International Journal of Distributed Sensor Networks*, Taylor & Francis, Inc., v. 2015, p. 17, 2015. Citado na página 21.

LIU, Y.-C.; CHEN, C.; CHAKRABORTY, S. A software defined network architecture for geobroadcast in vanets. In: IEEE. *2015 IEEE International Conference on Communications (ICC)*. [S.l.], 2015. p. 6559–6564. Citado 2 vezes nas páginas 22 e 25.

LOSILLA, F. et al. A comprehensive approach to wsn-based its applications: A survey. *Sensors*, Molecular Diversity Preservation International, v. 11, n. 11, p. 10220–10265, 2011. Citado 8 vezes nas páginas 9, 18, 19, 29, 30, 31, 32 e 33.

LUÍS, N. M. A. *Melhoria de protocolos de encaminhamento em VANETs de alta densidade*. Tese (Doutorado) — FCT-UNL, 2009. Citado 11 vezes nas páginas 9, 11, 18, 43, 44, 50, 51, 53, 54, 55 e 56.

LYNCH, N. A. *Distributed algorithms*. [S.l.]: Morgan Kaufmann, 1996. Citado na página 38.

MACÊDO, R. J. de A. Failure detection in asynchronous distributed systems. 2000. Citado na página 38.

MATOS, L. B. C. d. et al. Análise de desempenho de algoritmos criptográficos assimétricos em uma rede veicular (vanet). Ciência da Computação, 2013. Citado 2 vezes nas páginas 20 e 45.

MEJRI, M. N.; BEN-OTHMAN, J.; HAMDI, M. Survey on vanet security challenges and possible cryptographic solutions. *Vehicular Communications*, Elsevier, v. 1, n. 2, p. 53–66, 2014. Citado 4 vezes nas páginas 47, 48, 49 e 50.

MELNIKOV, A.; FETTE, I. The websocket protocol. *Request for Comments*, v. 6455, 2011. Citado na página 40.

MERSHAD, K.; ARTAIL, H. Finding a star in a vehicular cloud. *IEEE Intelligent transportation systems magazine*, IEEE, v. 5, n. 2, p. 55–68, 2013. Citado na página 23.

MIURA, S.; ZHAN, Y.; KURODA, T. Evaluation of parking search using sensor network. In: IEEE. *2006 1st International Symposium on Wireless Pervasive Computing*. [S.l.], 2006. p. 6–pp. Citado na página 32.

NANDAN, A. et al. Co-operative downloading in vehicular ad-hoc wireless networks. In: IEEE. *Second Annual Conference on Wireless On-demand Network Systems and Services*. [S.l.], 2005. p. 32–41. Citado na página 21.

NASIM, R.; KASSLER, A. Distributed architectures for intelligent transport systems: A survey. In: IEEE. *Network Cloud Computing and Applications (NCCA), 2012 Second Symposium on*. [S.l.], 2012. p. 130–136. Citado na página 18.

NIEDERMAIER, B. et al. The new bmw idrive–applied processes and methods to assure high usability. In: SPRINGER. *International Conference on Digital Human Modeling*. [S.l.], 2009. p. 443–452. Citado na página 33.

OBE, R. O.; HSU, L. S. *PostGIS in action*. [S.l.]: Manning Publications Co., 2015. Citado na página 70.

PAPADIMITRATOS, P. et al. Secure vehicular communication systems: design and architecture. *IEEE Communications Magazine*, IEEE, v. 46, n. 11, p. 100–109, 2008. Citado 5 vezes nas páginas 7, 11, 75, 83 e 87.

PATHRE, A. Identification of malicious vehicle in vanet environment from ddos attack. *Journal of Global Research in Computer Science*, v. 4, n. 6, p. 30–34, 2013. Citado na página 49.

PEDEN, M. et al. *World report on road traffic injury prevention*. [S.l.]: World Health Organization Geneva, 2004. Citado na página 18.

QIAN, Y.; LU, K.; MOAYERI, N. A secure vanet mac protocol for dsrc applications. In: IEEE. *IEEE GLOBECOM 2008-2008 IEEE Global Telecommunications Conference*. [S.l.], 2008. p. 1–5. Citado 2 vezes nas páginas 9 e 57.

QIN, H. et al. An integrated network of roadside sensors and vehicles for driving safety: Concept, design and experiments. In: IEEE. *Pervasive Computing and Communications (PerCom), 2010 IEEE International Conference on*. [S.l.], 2010. p. 79–87. Citado na página 32.

QIN, Y.; HUANG, D.; ZHANG, X. Vehicloud: Cloud computing facilitating routing in vehicular networks. In: IEEE. *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*. [S.l.], 2012. p. 1438–1445. Citado 2 vezes nas páginas 22 e 25.

RAWAT, A.; SHARMA, S.; SUSHIL, R. Vanet: Security attacks and its possible solutions. *Journal of Information and Operations Management*, Bioinfo Publications, v. 3, n. 1, p. 301, 2012. Citado na página 48.

RAYA, M.; PAPADIMITRATOS, P.; HUBAUX, J.-P. Securing vehicular communications. *IEEE Wireless Communications Magazine, Special Issue on Inter-Vehicular Communications*, v. 13, n. LCA-ARTICLE-2006-015, p. 8–15, 2006. Citado 2 vezes nas páginas 45 e 50.

SAMARA, G.; AL-SALIHY, W. A.; SURES, R. Security issues and challenges of vehicular ad hoc networks (vanet). In: IEEE. *New Trends in Information Science and Service Science (NISS), 2010 4th International Conference on*. [S.l.], 2010. p. 393–398. Citado 2 vezes nas páginas 20 e 45.

SANTOS, R.; EDWARDS, R.; EDWARDS, A. Cluster-based location routing algorithm for inter-vehicle communication. In: IEEE. *Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th*. [S.l.], 2004. v. 2, p. 914–918. Citado na página 53.

SOLINGEN, R. V. et al. Goal question metric (gqm) approach. *Encyclopedia of software engineering*, Wiley Online Library, 2002. Citado na página 74.

SOOKHAK, M.; YU, F. R.; TANG, H. Secure data sharing for vehicular ad-hoc networks using cloud computing. In: *Ad Hoc Networks*. [S.l.]: Springer, 2017. p. 306–315. Citado 2 vezes nas páginas 23 e 25.

SOUZA RONIEL DE; SOARES, A. C. B. Estimativa e sinalização de congestionamentos de trafego atraves de redes veiculares v2v. *Simposio Brasileiro de Redes de Computadores e Sistemas Distribuidos*, 2014. Citado na página 44.

SUMRA, I. A.; HASBULLAH, B.; MANAN, J. Comparative study of security hardware modules (edr, tpd and tpm) in vanet. In: *Third National Information Technology Symposium", NITS 2011 Symposium*. [S.l.: s.n.], 2011. p. 6–9. Citado na página 18.

SUNG, K.; YOO, J.; KIM, D. Collision warning system on a curved road using wireless sensor networks. In: IEEE. *2007 IEEE 66th Vehicular Technology Conference*. [S.l.], 2007. p. 1942–1946. Citado na página 31.

SYSTEMATICS, C.; MEYER, M. Crashes vs. congestion-what's the cost to society. *Prepared for the American Automobile Association*, 2011. Citado na página 19.

TANENBAUM, A.; STEEN, M. V. *Sistemas Distribuídos - Princípios e Paradigmas. 2ª Edição*. [S.l.]: Editora Pearson Prentice Hall, 2007. Citado na página 35.

TANGADE, S. S.; MANVI, S. S. A survey on attacks, security and trust management solutions in vanets. In: IEEE. *Computing, Communications and Networking Technologies (ICCCNT), 2013 Fourth International Conference on*. [S.l.], 2013. p. 1–6. Citado 3 vezes nas páginas 46, 47 e 49.

THEMUDO, V. M. Implementação de um servidor de negociação em bolsa baseado em websocket. 2014. Citado 3 vezes nas páginas 9, 41 e 42.

TTI, T. T. I. U. M. *Texas Transport Institute Urban Mobility Report 2014*. 2014. Disponível em: ⟨http://d2dtl5nnlpfr0r.cloudfront.net/tti.tamu.edu/documents/ums/congestion-data/national/national-table1.pdf⟩. Citado na página 19.

WANGHAM, M. S. et al. Segurança em redes veiculares: Inovações e direções futuras. 2014. Citado 7 vezes nas páginas 9, 45, 46, 47, 48, 49 e 50.

WASEF, A. et al. Complementing public key infrastructure to secure vehicular ad hoc networks [security and privacy in emerging wireless networks]. *IEEE Wireless Communications*, IEEE, v. 17, n. 5, p. 22–28, 2010. Citado na página 50.

WEINGÄRTNER, E.; KARGL, F. A prototype study on hybrid sensor-vehicular networks. *Proceedings of the 6th GI/ITG KuVS Fachgespräch "Wireless Sensor Networks*, 2007. Citado na página 31.

XU, Q. et al. Design and analysis of highway safety communication protocol in 5.9 ghz dedicated short range communication spectrum. In: IEEE. *Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual*. [S.l.], 2003. v. 4, p. 2451–2455. Citado na página 18.

YAN, G. et al. Security challenges in vehicular cloud computing. *IEEE Transactions on Intelligent Transportation Systems*, IEEE, v. 14, n. 1, p. 284–294, 2013. Citado 2 vezes nas páginas 22 e 25.

ZHANG, J. A survey on trust management for vanets. In: IEEE. *2011 IEEE International Conference on Advanced Information Networking and Applications*. [S.l.], 2011. p. 105–112. Citado na página 50.

ZHU, T.; LIU, Z. Intelligent transport systems in china: Past, present and future. In: IEEE. *Measuring Technology and Mechatronics Automation (ICMTMA), 2015 Seventh International Conference on*. [S.l.], 2015. p. 581–584. Citado na página 18.