



INSTITUTO SUPERIOR TÉCNICO  
Universidade Técnica de Lisboa

## **Redes Veiculares de Emergência**

**Heitor Filipe Costa**

Dissertação para obtenção do grau de Mestre em  
**Engenharia de Redes de Comunicações**

### **Júri**

Presidente: Rui Jorge Morais Tomaz Valadas

Orientador: Teresa Maria Sá Ferreira Vazão Vasques

Vogais: Artur Arsénio

**Novembro de 2010**



---

## Agradecimentos

---

Em primeiro lugar quero agradecer à minha orientadora Professora Teresa Vazão por todo o apoio, paciência, disponibilidade e pela confiança que depositou em mim, pelas rápidas correções assim como pelo optimismo e apoio transmitido, durante a realização da dissertação, sem os quais este trabalho não teria sido possível.

A toda a minha família, nomeadamente a minha namorada, por me apoiarem em tudo ao longo da minha vida.

Agradeço também a todos os amigos o apoio que me deram, principalmente ao meu grande amigo Tiago Pedrosa pela ajuda e paciência que teve para comigo, e na ajuda que me deu na melhoria do conteúdo, assim como na apresentação desta tese.



---

## Resumo

---

As redes veiculares são formadas por automóveis que comunicam directamente entre si, através de tecnologias de comunicação sem fios e de protocolos de encaminhamento adequados. Como requisito à comunicação, uma rota eficiente entre nós tem que ser criada, e acompanhar a rápida mudança da topologia da rede. Devido a características como a alta mobilidade dos nós, falhas de rede e requisitos de latência, muitos dos protocolos utilizados apresentam somente um desempenho minimamente satisfatório. Mais do que um problema de selecção do protocolo de encaminhamento mais adequado, a questão fundamental que se coloca é o desenho da arquitectura global do sistema de comunicação.

Apresenta-se uma arquitectura VANET que permita responder de forma eficaz às necessidades da comunicação entre veículos, nomeadamente em cenários de emergência em ambiente de auto-estradas, onde se pretende que os condutores sejam alertados atempadamente para situações de emergências.

De forma a analisar se a referida arquitectura vai de encontro aos requisitos necessários efectuaram-se simulações. As simulações foram realizadas utilizando o programa de simulação NS-2, desenvolvido especialmente para a análise de todos os tipos de redes de comunicação, o qual fornece suporte nativo à simulação de redes ad hoc como é o caso das VANETs. A análise dos resultados destas redes, concentrou-se na análise de duas métricas usadas para medir o desempenho das redes: End-To-End Delay e Packet Delivery Ratio.

Com isto foi possível verificar que à medida que as redes escalam, os valores médios das métricas demonstram uma degradação do desempenho dessas redes, não obstante a solução demonstrou responder às necessidades do tempo de alerta para que os condutores possam reagir atempadamente, de forma a evitar mais situações de emergência.

## Palavras-chave

---

VANETs, Protocolos, Políticas de escalonamento, Correlação de eventos, Ambientes, Segurança.



---

## Abstract

---

The vehicular networks are formed by vehicles, which communicate directly with each other through wireless communication technologies and routing protocols suitable. As a prerequisite to communication, an efficient route between nodes must be created, and keep up with rapid change of network topology. Due to such characteristics as high node mobility, network failures and latency requirements, many of the protocols used have only a minimally satisfactory performance. More than a problem of selecting the most appropriate routing protocol, the fundamental question that arises is the design of the global communication system.

Is presented a Vanet architecture that enable an effective communications between vehicles particularly in emergency scenarios on motorways, where it is intended that drivers will be alerted in time of the emergencies. Simulations where preformed to measure whether this architecture meets the necessary requirements. The simulations were implemented using the software NS-2, developed specifically for analyzing all types of communication networks, which provides native support for simulation of ad hoc networks, as is the case of VANETs. The results were focused on the analysis of two metrics, End-To-End Delay and Packet Delivery Ratio used to measure the networks performance.

It was observed that as the network scale, the average values of the metrics shows a degradation of performance on those networks, despite the solution has demonstrated that copes with the requirements, as it can alert drivers in time to react promptly in order to avoid more emergencies.

## Keywords

---

VANETs, Protocols, Scheduling Policies, Events Correlation, Environments, Security





---

## Índice

---

<b>AGRADECIMENTOS</b> .....	<b>I</b>
<b>RESUMO</b> .....	<b>III</b>
PALAVRAS-CHAVE .....	III
<b>ABSTRACT</b> .....	<b>V</b>
KEYWORDS .....	V
<b>ÍNDICE</b> .....	<b>VII</b>
<b>LISTA DE SIGLAS E ABREVIATURAS</b> .....	<b>IX</b>
<b>ÍNDICE DE FIGURAS</b> .....	<b>XI</b>
<b>ÍNDICE DAS TABELAS</b> .....	<b>XIII</b>
<b>1 INTRODUÇÃO</b> .....	<b>15</b>
1.1 MOTIVAÇÃO .....	15
1.2 OBJECTIVOS DO TRABALHO .....	16
1.3 ESTRUTURA DO DOCUMENTO .....	17
<b>2 TRABALHO RELACIONADO</b> .....	<b>19</b>
2.1 REDES SEM FIOS AD HOC IEEE 802.11 .....	19
2.2 MANETs VS VANETs .....	19
2.2.1 Aplicações .....	20
2.2.2 Ambientes .....	21
2.3 MODOS DE COMUNICAÇÃO EM REDES VEICULARES .....	22
2.4 POLITICAS DE ESCALONAMENTO .....	23
2.4.1 Partilha de recursos e escalonamento .....	23
2.4.2 Algoritmos de Escalonamento .....	23
2.5 CORRELAÇÃO DE EVENTOS .....	24
2.5.1 Inteligência artificial .....	25
2.5.2 Propagação de falhas .....	26
2.6 PROTOCOLOS .....	27
2.6.1 Protocolos Topológicos .....	28
2.6.2 Protocolos Geográficos .....	28
2.6.3 Protocolos Oportunísticos .....	29
2.6.4 Protocolos de Disseminação .....	29
<b>3 PROPOSTA DE SOLUÇÃO</b> .....	<b>31</b>
3.1 REQUISITOS TEMPORAIS .....	31
3.1.1 Tempo de reacção e tempo de travagem .....	31

3.1.2	<i>Distância de travagem e distância de segurança</i> .....	32
3.2	ARQUITECTURA DA SOLUÇÃO .....	34
3.2.1	<i>Protocolo</i> .....	36
3.2.2	<i>Modelo Base</i> .....	36
3.2.3	<i>Modelo com prioridades</i> .....	37
3.2.4	<i>Modelo com correlação</i> .....	37
<b>4</b>	<b>IMPLEMENTAÇÃO</b> .....	<b>39</b>
4.1	ANÁLISE DE SIMULADORES .....	39
4.1.1	<i>Exemplo de utilização do NS-2 em Redes Móveis</i> .....	42
4.1.2	<i>Mobile Node</i> .....	42
4.1.3	<i>Mecanismos de Encaminhamento</i> .....	44
4.2	EXEMPLO DA CRIAÇÃO DE UMA VANET .....	44
4.3	MODELO DE SIMULAÇÃO PARA TESTES .....	49
4.3.1	<i>Modelo Base</i> .....	49
4.3.2	<i>Modelo com prioridades</i> .....	54
4.3.3	<i>Modelo com correlação</i> .....	56
<b>5</b>	<b>TESTES DE DESEMPENHO</b> .....	<b>59</b>
5.1	RESULTADO DA EXECUÇÃO DA SIMULAÇÃO .....	59
5.2	RESULTADOS DO <i>PACKET-DELIVERY RATIO</i> .....	62
5.3	RESULTADOS DO <i>END-TO-END DELAY</i> .....	66
<b>6</b>	<b>CONCLUSÕES E TRABALHO FUTURO</b> .....	<b>69</b>
<b>7</b>	<b>REFERÊNCIAS</b> .....	<b>71</b>

---

## Lista de siglas e abreviaturas

---

**BSS/ESS:** *Basic Service Set/Extended Service Set*

**CCH:** *Control Channel*

**CMU:** *Carnegie Mellon University*

**DSRC:** *Dedicated Short Range Communications*

**FCC:** *Federal Communications Commission*

**IBSS:** *Independent Basic Service Set*

**IP:** *Internet Protocol*

**ITS:** *Intelligent Transportation Systems*

**IVC:** *Inter-Vehicle Communications*

**LLC:** *Logical Link Control*

**MANET:** *Mobile Ad-Hoc Network*

**MLME:** *Mac Layer Management Entity*

**NAM:** *Network AniMator*

**PLME:** *Physical Layer Management Entity*

**PKI:** *Public Key Infrastructure*

**RVC:** *Road-Vehicle Communications*

**RLS:** *Reactive Location Service*

**SCHs:** *Service Channels*

**TCP:** *Transmission Control Protocol*

**UDP:** *User Datagram Protocol*

**VANET:** *Vehicular Ad-Hoc Network*

**WAVE:** *Wireless Access in the Vehicular Environment*

**WME:** *WAVE Management Entity*

**WSMP:** *WAVE Short Message Protocol*

---

## Índice de figuras

---

<b>FIGURA 1</b> – ESTRUTURA DO DOCUMENTO .....	17
<b>FIGURA 2</b> – TEMPO DE REACÇÃO E TRAVAGEM .....	32
<b>FIGURA 3</b> – ARQUITECTURA DE REDE .....	35
<b>FIGURA 4</b> – CRIAÇÃO DOS NÓS DA REDE .....	43
<b>FIGURA 5</b> – POSIÇÃO E MOVIMENTO DOS NÓS DA REDE .....	43
<b>FIGURA 6</b> – CONFIGURAÇÕES DA SIMULAÇÃO EXEMPLO .....	45
<b>FIGURA 7</b> – DISPOSIÇÃO DOS NÓS NA TOPOGRAFIA .....	46
<b>FIGURA 8</b> – COMUNICAÇÃO CBR ENTRE DOIS NÓS MÓVEIS .....	46
<b>FIGURA 9</b> – USO DO NOVO FORMATO DE TRACE ( <i>NEWTRACE</i> ) .....	47
<b>FIGURA 10</b> – FICHEIRO DE TRACE EM MODO VISUAL.....	48
<b>FIGURA 11</b> – FICHEIRO DE TRACE MODO TEXTO .....	48
<b>FIGURA 12</b> – CONFIGURAÇÕES DA SIMULAÇÃO .....	49
<b>FIGURA 13</b> – POSIÇÃO INICIAL DOS <i>MOBILE NODE</i> NA TOPOGRAFIA .....	50
<b>FIGURA 14</b> – CRIAÇÃO DOS NÓS DA REDE .....	51
<b>FIGURA 15</b> – FUNÇÃO QUE PERMITE GERAR TRÁFEGO CBR .....	52
<b>FIGURA 16</b> – PROCEDIMENTO CBR .....	52
<b>FIGURA 17</b> – FUNÇÃO QUE PERMITE GERAR TRÁFEGO FTP .....	53
<b>FIGURA 18</b> – PROCEDIMENTO FTP .....	54
<b>FIGURA 19</b> – FUNÇÃO NO NS-2 QUE CRIA AS DUAS FILAS DE PRIORIDADES .....	55
<b>FIGURA 20</b> – DEFINIÇÃO DO TIPO DE FILA A USAR .....	55
<b>FIGURA 21</b> – DEFINIÇÃO DO TAMANHO DAS FILAS .....	55
<b>FIGURA 22</b> – FUNÇÃO DE ENVIO DE MENSAGEM DE ACIDENTE.....	56
<b>FIGURA 23</b> – FUNÇÃO DE CRIAÇÃO DOS DOIS NOVOS CAMPOS DO TRACE .....	57
<b>FIGURA 24</b> – OUTPUT DO NOVO FORMATO DE TRACE ( <i>NEWTRACE</i> ) .....	60
<b>FIGURA 25</b> – OUTPUT DO NAM .....	61
<b>FIGURA 26</b> – <i>PACKET-DELIVERY RATIO</i> SEM TRÁFEGO NA REDE .....	62
<b>FIGURA 27</b> – <i>PACKET-DELIVERY RATIO</i> COM TRÁFEGO NA REDE .....	63
<b>FIGURA 28</b> – <i>PACKET-DELIVERY RATIO</i> COM SITUAÇÕES DE BUFFER CHEIO .....	63
<b>FIGURA 29</b> – <i>PACKET-DELIVERY RATIO</i> RECORRENDO A PRIORIDADES .....	64



---

## Índice das Tabelas

---

<b>TABELA 1</b> – DISTÂNCIA DE SEGURANÇA EM CONDIÇÕES DE PISO SECO .....	34
<b>TABELA 2</b> – DISTÂNCIA DE SEGURANÇA EM CONDIÇÕES DE PISO MOLHADO .....	34
<b>TABELA 3</b> – PERCENTAGEM DO USO DE CADA SIMULADOR.....	39
<b>TABELA 4</b> – <i>PACKET DELIVERY RATIO</i> SEM CORRELAÇÃO.....	65
<b>TABELA 5</b> – <i>PACKET DELIVERY RATIO</i> COM CORRELAÇÃO.....	66
<b>TABELA 6</b> - TABELA DE TEMPOS MEDIOS DE ENTREGA .....	67
<b>TABELA 7</b> - TEMPO REAL DE AVISO DE ACIDENTE ATÉ AO QUILOMETRO X .....	67
<b>TABELA 8</b> - TEMPO REAL DE AVISO DE TODOS OS ACIDENTADOS ATÉ AO QUILOMETRO X.....	68





---

# 1 Introdução

---

## 1.1 Motivação

---

Hoje em dia, as tecnologias de comunicações sem fios têm tido um desenvolvimento muito rápido, dando origem a diversas áreas de aplicação [1]. No domínio das redes móveis surgiram, inicialmente, as redes móveis não estruturadas (*Mobile Ad-Hoc Network* - MANET), em que cada nó da rede podia assumir funções de sistema terminal ou de encaminhador, o que permitia que a comunicação entre nós móveis fosse possível [2]. O facto de poderem existir sistemas embebidos dentro de veículos, equipados com capacidades de comunicação, levou ao nascimento duma nova área de pesquisa dentro das MANETs, que hoje em dia se designa por rede veicular (*Vehicular Ad-Hoc Network* - VANET).

As VANETs são formadas pelos veículos que comunicam directamente entre si, através de tecnologias de comunicação sem fios e de protocolos de encaminhamento adequados. O elevado dinamismo deste tipo de redes faz com que esta seja uma área de investigação com enormes desafios tecnológicos, tendo por isso sido alvo do interesse generalizado da comunidade académica. Por outro lado, a importância que este tipo de redes poderá vir a assumir no futuro, justifica o enorme interesse na área das empresas do sector automóvel e das organizações do governo, entre outras.

Existem três grandes classes de aplicações dentro da área dos Sistemas de Transportes Inteligentes (Intelligent Transport System - ITS): **entretenimento, segurança rodoviária e assistência**. A crescente mortalidade nas estradas conduz à necessidade dum aumento da segurança rodoviária, fazendo com que a informação atempada aos condutores, sobre situações críticas, seja uma prioridade. A maioria das aplicações de entretenimento usadas em redes veiculares estão associadas ao acesso e uso da Internet. A dependência crescente que existe hoje em dia leva os utilizadores a querer aceder à Internet a qualquer hora e em qualquer lugar. As aplicações pensadas para assistência ao condutor, focam-se principalmente na automatização de tarefas. São exemplos de algumas delas: localização de postos de abastecimento, áreas de serviço, informações a nível turístico.

Centrando a análise apenas nas aplicações destinadas a aumentar a segurança rodoviária é possível identificar vários casos em que os requisitos impostos ao sistema de comunicação são muito diferentes. Por exemplo, no caso de um acidente, o aviso atempado das autoridades é um factor crítico no socorro às vítimas, pelo que a transferência da informação associada é prioritária face à comunicação com outros veículos. Numa outra perspectiva, ainda em situação de acidente, é necessário avisar os condutores que circulam numa dada área o mais rapidamente possível, de forma a evitar choques em cadeia e a oferecer a possibilidade destes seguirem por rotas alternativas. Este tipo de informação, embora de natureza menos crítica que a informação anterior tem um enorme impacto na comunicação, pelo elevado volume de tráfego que deve circular na vizinhança do

acidente, num curto espaço de tempo. Outro tipo de situação em que se pode tirar partido da comunicação veicular é o caso duma avaria, em que o condutor pretende avisar a família ou a sua seguradora, para que lhe seja enviado um reboque. Neste tipo de cenário, os problemas essenciais que se colocam prendem-se com a eventual necessidade de uso de outro tipo de redes, para além da VANET, para completar a ligação com o destinatário da chamada.

Pode-se então antever que o sucesso das VANETs está relacionado com a capacidade das soluções tecnológicas que vierem a ser utilizadas servirem para satisfazer as necessidades dos condutores de forma adequada. Neste contexto, mais do que um problema de selecção do protocolo de encaminhamento mais adequado, como muitos autores reivindicam [3] e [4] a questão fundamental que se coloca é o desenho da arquitectura global do sistema de comunicação.

## 1.2 Objectivos do Trabalho

---

O objectivo principal do trabalho que se descreve na presente proposta de tese é a concepção e implementação duma arquitectura de rede veicular de emergência destinada a suportar aplicações de segurança física.

Pretende-se assim estudar soluções na área das redes veiculares que permitam aumentar a segurança dos veículos. Desta forma dever-se-á caracterizar o cenário da utilização deste tipo de redes na circulação automóvel, nomeadamente nas cidades e nas auto-estradas, bem como as implicações destes cenários na arquitectura e nas outras escolhas que serão tomadas.

Sabendo *à priori* que existem soluções com desempenho minimamente satisfatório em cada um dos cenários em estudo, procurar-se-á maximizar o desempenho, concebendo uma arquitectura o mais standard possível, recorrendo, sempre que possível, a protocolos já existente, ao invés da procura dos protocolos mais adequados para cada situação. Com este intuito irá ser proposta um sistema que englobe:

- A definição de uma arquitectura da rede de comunicações que seja passível de uso em qualquer tipo de cenário, cidades e/ou auto-estrada, e que permita proceder ao envio da informação de acordo com o seu nível de criticidade;

- A escolha do protocolo de encaminhamento que permita efectuar a transferência de informação na VANET e desta para outra rede, se aplicável;

- A selecção da política de escalonamento da informação a transferir, como forma de assegurar que a informação é transferida de acordo com a sua prioridade;

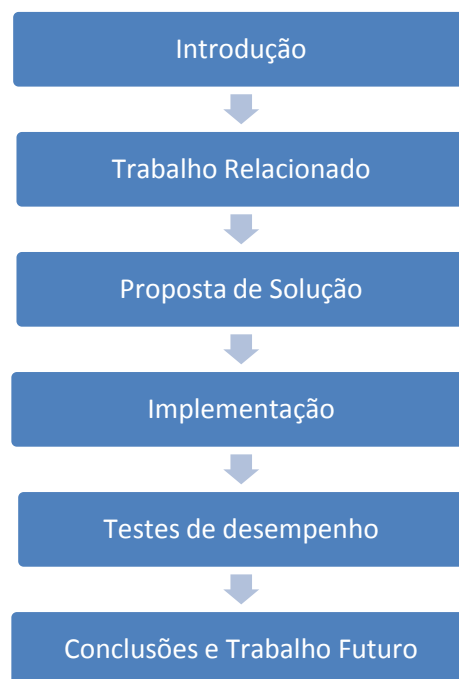
- A selecção do mecanismo de correlação de eventos, como forma de reduzir a quantidade de informação transferida

A arquitectura a propôr deverá minimizar o custo, recorrendo, por exemplo, a uma solução de comunicação que não recorra ao uso duma rede infra-estruturada, pois este é um dos investimentos mais avultados neste tipo de soluções. Como qualquer solução que tenha por objectivo a divulgação generalizada, em larga escala e para todos os tipos de utilizadores, esta deve ser a mais cómoda possível, devendo actuar de forma transparente, se possível, sem recurso a utilizadores especializados. Poderá ser equacionado o uso de redes de comunicações alternativas, sempre que a criticidade da informação a transferir o exija e as condições de comunicação na VANET o justifiquem.

### 1.3 Estrutura do documento

---

Este documento está organizado da seguinte forma (figura 1): No capítulo I, apresentam-se a motivações da realização deste trabalho, bem como os seus objectivos e termina-se com a descrição da estrutura do documento. No capítulo II é feito um estudo das tecnologias existentes, assim como os protocolos que servem como base a todo o trabalho, além de se caracterizar o cenário em que este trabalho será desenvolvido, nomeadamente refere-se superficialmente a temática das redes veiculares, assim como o ambiente e aplicações em que este trabalho se insere. É apresentada uma proposta de solução no capítulo III, explicando-se a evolução de um modelo base até ao modelo proposto. No capítulo IV descreve-se de que formas foram realizadas a simulações da implementação da arquitectura da solução, permitindo que no capítulo V seja efectuada uma análise e testes à arquitectura proposta. No capítulo VI apresentam-se as conclusões, bem como oportunidades de trabalho futuro relacionadas com o trabalho apresentado.



**Figura 1** – Estrutura do Documento



---

## 2 Trabalho Relacionado

---

Indo ao encontro dos objectivos e requisitos previamente abordados, segue-se uma análise da literatura científica sobre diversos assuntos importantes para esta tese como por exemplo, os protocolos de routing, as políticas de escalonamento e os ambientes onde estas redes se inserem. Isto permitirá conhecer abordagens de diferentes autores a problemas similares, bem como aumentar o conhecimento das tecnologias que suportam este tipo de soluções. Pretendendo servir de base para a proposta da solução desta tese para o problema previamente referido.

### 2.1 Redes Sem Fios Ad Hoc IEEE 802.11

---

O objecto de estudo deste trabalho tem foco principal na simulação de redes sem fios *ad hoc* IEEE 802.11. Este subcapítulo demonstra alguns princípios básicos da especificação IEEE 802.11.

A norma 802.11, genericamente denominado wi-fi, corresponde ao grupo de trabalho do *IEEE LAN/MAN Standards Committee* (IEEE 802) [29]. Este grupo de trabalho possui subdivisões em vários outros grupos de trabalho que são responsáveis por desenvolver áreas específicas deste padrão. Esta norma é responsável pela especificação das camadas 1 e 2 do modelo OSI (física e ligação lógica, respectivamente) da comunicação sem fios. Nesta especificação, a norma define duas topologias básicas de operação: os clientes comunicam-se directamente uns com os outros ou, no segundo caso, ligam-se inicialmente a um ponto de acesso central para depois comunicarem entre si. Estas duas topologias lógicas são definidas por: *Independent Basic Service Set* (IBSS), usualmente designada por modo *ad-hoc*, e *Basic Service Set/Extended Service Set* (BSS/ESS), conhecido como modo infra-estruturado [30].

Na comunicação por modo *ad-hoc*, os nós comunicam directamente uns com os outros. Caso um nó esteja fora do alcance de transmissão directa de outro nó, a comunicação entre eles só é possível com a transmissão *multi-hop* (múltiplos saltos), na qual nós intermédios (saltos) têm que retransmitir os pacotes da origem ao destino, agindo como routers, usando algum protocolo de routing apropriado.

Sendo o foco principal deste tese as redes IBSS, *ad-hoc*, de seguida são explicados todos os modos de comunicação possíveis nestas redes.

### 2.2 MANETs vs VANETs

---

A possibilidade de comunicar em ambientes móveis, sem recurso a qualquer tipo de infra-estrutura, tem motivado um interesse crescente na exploração das redes móveis auto-organizadas, usualmente conhecidas por MANETs. A extensão do seu uso ao ambiente rodoviário deu origem a uma nova área de estudo - as VANETs que tem como objectivos: melhorar as condições de circulação do tráfego rodoviário de forma segura e eficiente e proporcionar conforto aos passageiros,

oferecendo diversas aplicações, como por exemplo o acesso à Internet ou partilha de conteúdos multimédia, através de comunicações multi-hop entre os veículos.

Efectuando uma comparação entre as MANETs e as VANETs pode-se considerar que o grande factor diferenciador entre os dois tipos de rede está relacionado com o padrão de mobilidade dos nós [3]. Enquanto nas MANETS a mobilidade é restrita e as ligações entre os nós permanecem estáveis durante um período de tempo relativamente longo, nas VANETs a velocidade a que os nós se podem mover é bastante maior, o padrão de mobilidade mais determinístico e as ligações permanecem activas menos tempo. Esta diferença de comportamento afecta drasticamente o desempenho dos protocolos de encaminhamento, condicionando fortemente as suas características e restringindo a aplicabilidade das soluções de encaminhamento das MANETs ao ambiente das VANETs. Este problema já foi identificado por diversos autores, que propuseram soluções de encaminhamento especificamente concebidas para este tipo de redes [4], [5] e [6]. Não obstante, e conforme descrito em [3], o processo de encaminhamento numa VANET poderá ser simplificado se se considerar que os veículos circulam numa rede de estradas predefinida, com a indicação de sentido de deslocamento e das estradas por onde podem circular no futuro. Tendo por base esta informação será possível decidir quando e para que nós se deverão encaminhar as mensagens.

### 2.2.1 Aplicações

---

A comunicação entre veículos permite o fornecimento de serviços que de outra forma seria muito oneroso fornecer. Desta forma, dentro da área ITS as VANETs permitem contemplar as três classes de aplicações anteriormente referidas.

- **Segurança rodoviária** - A crescente mortalidade nas estradas conduz à necessidade dum aumento da segurança rodoviária, fazendo com que a informação atempada aos condutores, sobre situações críticas, seja uma prioridade. Este tipo de aplicações tem um carácter preventivo, sendo o seu principal objectivo a divulgação rápida de informação, como forma de possibilitar ao condutor um maior tempo de reacção.

- **Entretenimento** - A maioria das aplicações de entretenimento usadas em redes veiculares estão associadas ao acesso e uso da Internet. A dependência crescente que existe hoje em dia leva os utilizadores a querer aceder à Internet a qualquer hora e em qualquer lugar. Nas aplicações mais procuradas destacam-se os serviços de mensagens instantâneas e partilha de áudio de vídeo. Um dos maiores desafios que se coloca neste domínio está associado ao facto das aplicações de entretenimento terem todas, requisitos muito diferentes.

- **Assistência** - As aplicações pensadas para assistência ao condutor, focam-se principalmente na automatização de tarefas. São exemplos de algumas delas: localização de postos de abastecimento, áreas de serviço, informações a nível turístico. Nestas aplicações, ao contrário do que acontece nas relacionadas com a segurança, a divulgação rápida de informação não é um factor

crítico, sendo porém fundamental que a divulgação de informação deve ser o mais abrangente possível.

Com estas três classes de aplicações bastante distintas, o tipo de uso e o ambiente em que são usadas têm diferentes impactos na rede. Para além deste facto, será necessário caracterizar também os ambientes onde serão utilizadas estas aplicações.

## 2.2.2 Ambientes

---

Um dos principais desafios que se colocam às VANETs está relacionado com a possibilidade dos nós se deslocarem a alta velocidade, o que reduz substancialmente o tempo de contacto entre nós e a possibilidade de troca das mensagens entre estes. Este aspecto afecta fortemente o desempenho das aplicações, condicionando a própria arquitectura da solução a usar. Existe assim a necessidade de diferenciar dois ambientes onde as VANETs são aplicadas - as auto-estradas e cidades -, de forma a identificar os principais requisitos em termos de arquitectura que são impostos por cada um deles.

Uma auto-estrada é uma via de comunicação destinada apenas a tráfego motorizado, em que as faixas de rodagem são maioritariamente rectas, e onde não existem cruzamentos nem semáforos. Assim sendo, os nós podem circular com uma velocidade elevada, seguindo quase sempre a mesma direcção e sentido. A quase ausência de obstáculos, nomeadamente de edifícios, facilita a propagação das ondas rádio, conseguindo-se assim uma maior cobertura de rede.

Pelo facto da velocidade relativa dos nós poder ser bastante elevada, o tempo expectável de contacto é menor. Mas, por outro lado, a existência de poucas saídas permite aumentar o tempo de contacto entre nós, quando o fluxo de tráfego rodoviário é contínuo, possibilitando assim uma maior troca de mensagens. Este aspecto é particularmente relevante porque, uma vez que a velocidade relativa dos nós pode ser bastante elevada, o tempo expectável de contacto é menor. Pelo facto dos nós poderem estar incomunicáveis durante um largo período de tempo não é possível assegurar a transferência contínua de mensagens. Este é um dos problemas mais críticos deste ambiente, com um forte impacto nas aplicações de segurança rodoviária. Como forma possível de resolver este problema poderá usada uma rede alternativa para completar o processo de transferência de informação crítica, quando a comunicação pela VANET não for viável, que seja acessível em qualquer lugar, como é o caso da rede GSM.

Uma cidade caracteriza-se por ser um ambiente com edifícios, ruas e estradas com cruzamentos, sinalização de trânsito e veículos a circular a uma velocidade relativamente reduzida. Edifícios ao longo das ruas causam fenómenos de interferência, tornando a captação e recepção do sinal de rádio muito mais difícil. Os vários caminhos que os veículos podem seguir dificultam o processo de estimativa das rotas rodoviárias, fazendo ainda com que a densidade de tráfego possa variar fortemente, consoante o local e a hora do dia. Uma vez que são inúmeras as possibilidades de mudança de rota de um nó, tendo em conta o número de vias/sentidos que poderá escolher a

qualquer momento, o encaminhamento não poderá tirar partido de uma estimativa de posição futura do nó.

## 2.3 Modos de Comunicação em Redes Veiculares

---

A transferência de informação numa VANET pode ser realizada com recurso a um dos seguintes modos de comunicação:

- **Unicast**, que se caracteriza pela existência dum processo de comunicação entre um dado nó de origem e um dado nó de destino, em que a informação é transferida por um único caminho, sem ser replicada. Este modo de comunicação é útil, na medida em que permite transferir informação sem um grande overhead. Porém, para aplicações que requeiram o envio da informação para múltiplos nós não é adequado, uma vez que não tira partido da possibilidade de partilha do meio intrínseca à comunicação sem fios. Existem uma série de algoritmos que usam este tipo de comunicação e que são usados quando se pretende criar um canal origem-destino entre dois nós. Este modo de comunicação é bastante útil para aplicações na área do entretenimento, por exemplo o acesso a internet [12].

- **Broadcast**, que se caracteriza pelo envio da informação em difusão, sendo utilizado para um nó de origem enviar uma mensagem para todos os nós da rede que estão ao seu alcance. Este modo de comunicação é o mais adequado quando se pretende abordar todos os nós possíveis. Face aos exemplos de aplicação apresentados, de entre todos os modos de comunicação este é o que melhor se adequa para a situação em que se pretende fazer uma disseminação rápida da mensagem.

- **Multicast**, que se caracteriza por permitir que uma mesma mensagem seja enviada para múltiplos destinos. Os nós a que se destina a informação devem pertencer a um mesmo grupo - o grupo Multicast - ao qual se devem previamente associar. Este processo de associação a um grupo Multicast envolve o consumo duma quantidade significativa de recursos, face à capacidade expectável duma VANET. Por outro lado, muitas vezes, o envio de informação numa VANET tem um cariz geográfico que não se pode satisfazer com o recurso a uma comunicação *Multicast*.

- **Geocast**, em que uma mensagem é enviada para todos os nós que se encontrem numa determinada posição geográfica, sendo, normalmente esta posição calculada relativamente ao nó que enviou a mensagem. Nestas condições, cada nó deve estar equipado com um dispositivo de localização (GPS). Este tipo de comunicação é particularmente adequado para a transmissão de determinados tipos de informação em VANETs, como por exemplo, informação sobre acidentes.



## 2.4 Políticas de Escalonamento

---

### 2.4.1 Partilha de recursos e escalonamento

---

A partilha de recursos em redes de comunicação, neste caso VANETs, origina situações de contenção, devido à competição pela utilização dos recursos. Em redes que suportam integração de vários serviços como é o caso, torna-se necessário providenciar Qualidade de Serviço (*Quality of Service* - QoS) diferenciada por categorias, oferecendo garantias de desempenho a aplicações críticas (aplicações de segurança) e ao mesmo tempo permitindo a partilha de recursos por outro tipo de aplicações de acordo com critérios de equidade (*fairness*). Os algoritmos de escalonamento são uma componente essencial para atingir estes objectivos.

Um algoritmo de escalonamento realiza duas funções:

- Gere as filas de pedido de serviço;
- Decide a ordem de serviço de fluxos em competição.

Os algoritmos de escalonamento, desempenham um papel importante na provisão de diferentes níveis de QoS a diferentes aplicações, permitindo o controlo diferenciado de atraso, largura de banda ou taxa de perdas. Em geral pretende-se que os algoritmos de escalonamento sejam simples (fáceis de implementar), tratem de forma equitativa tráfego não prioritário (*best-effort*) e que garantam ao restante tráfego níveis de desempenho, no mínimo, satisfatórios.

### 2.4.2 Algoritmos de Escalonamento

---

Os algoritmos de Escalonamento estão divididos em *Work-conserving* e *Non-work-conserving*. A diferença entre estas duas disciplinas, está relacionada com a forma como se definem períodos de inactividade, ou seja, se se considera que o nó está inactivo apenas quando não existe nada para transmitir (*work-conserving*), ou se o nó pode estar inactivo mesmo que existam pacotes para transmitir (*non-work-conserving*).

Alguns exemplos de algoritmos são escalonamento do tipo *Work-conserving* são:

- **Round-Robin (RR)** [13], que se caracteriza por servir rotativamente um pacote de cada fila não vazia. É um protocolo justo se os pacotes tiverem comprimento fixo e se as ligações tiverem o mesmo peso;

- **Weighted Round-Robin (WRR)** [13], que se caracteriza por servir as ligações proporcionalmente aos pesos atribuídos, e pressupõe o conhecimento do tamanho médio dos pacotes gerados. É justo apenas em escalas temporais superiores à duração do ciclo, podendo ser injusto durante ciclos longos para ligações com um peso pequeno.

- **Fair Queueing (FQ)** [14], caracteriza-se por marcar os pacotes, à chegada, com a estimativa do instante em que o pacote deve voltar a sair. Desta forma os fluxos serão ordenados pelas marcas efectuadas à chegada.

- **Weighted Fair Queueing (WFQ)** [14], é um algoritmo idêntico ao FQ, a diferença é que o WFQ permite para diferentes sessões, ter diferentes partilhas de serviços.

Como algoritmos de escalonamento do tipo *Non-work-conserving* podem-se considerar:

- **Stop-and-Go** [15] e [16], caracteriza-se por utilizar uma estratégia de definição de vários níveis para alocar recursos em um único nó e por tentar garantir a fluidez do tráfego em toda a rede.

- **Rate-Controlled Static Priority (RCSP)** [15] e [16], é um algoritmo mais complexo que o Stop-and-Go. Este algoritmo separa as funções em duas componentes: (1) Tem um regulador que controla o tráfego de distorção introduzida pelos efeitos de multiplexação e variações de carga em outros nós e 2) É um regulador de prioridade para multiplexar o tráfego regulado.

O que se pretende com o uso dum algoritmo de escalonamento, é garantir que as mensagens de acidente têm prioridade face a qualquer outro tipo de tráfego que passe na rede. Deste modo, estes algoritmos têm um papel determinante nesta tese. Optou-se por utilizar uma política de escalonamento muito simples, nenhuma das referidas anteriormente porque não satisfaziam todos os requisitos necessários, que garanta que a informação de acidente é sempre prioritária face à demais. Deste modo, a política a seguir passa pela utilização de duas filas distintas, uma prioritária que tratará o tráfego referente as mensagens de acidente e outra responsável por todo o restante tráfego que circula na rede. Existe assim um escalonador que verifica o conteúdo de cada fila, e sempre que a fila das mensagens de acidente tiver algo para enviar, esta tem sempre prioridade face a outra. Assegura-se desta forma que o tráfego de emergência será atendido primeiro que qualquer outro tipo de tráfego.

## 2.5 Correlação de Eventos

---

Para que uma rede de comunicações possa ser utilizada de forma eficaz é indispensável que todos os seus recursos sejam geridos da melhor forma, reduzindo, nomeadamente, o consumo de largura de banda, que é um recurso fundamental numa VANET. Uma forma possível de efectuar esta gestão eficaz é reduzir a quantidade de informação transferida, procedendo à correlação de eventos que transportam essa informação, enriquecendo o valor destes.

Esta correlação deve ser capaz de resumir os eventos que recebeu num único evento que representa a ocorrência de todos os outros. Com a probabilidade de dois eventos ocorrerem simultaneamente, ou ligeiramente espaçadas no tempo é preciso ter em conta estas situações. Isto significa que estes sistemas terão de ser capazes de fazer a clara distinção entre eventos primários e secundários, de modo a conseguir identificar eventos distintos.

Outro aspecto importante, que deverá ser tomado em consideração, é que os eventos não estão apenas relacionados causalmente. Existe também uma relação temporal entre si. É portanto imperativo manter uma representação temporal consistente entre todos os agentes. Por todas as razões atrás apresentadas, é possível que os algoritmos de correlação não sejam capazes de identificar claramente qual o evento que terá ocorrido.

### 2.5.1 Inteligência artificial

---

A utilização da Inteligência Artificial, no contexto da correlação de eventos, tem por objectivo reproduzir comportamentos humanos na identificação de falhas [17]:

- **Redes Neurais** [33] [34]: tentam reproduzir o funcionamento dum cérebro humano através de ligações complexas entre nós (neurónios). Conseguem lidar com problemas desconhecidos, e permitem algumas inconsistências dos dados de entrada. As grandes desvantagens das Redes Neurais, é que requerem grandes períodos de tempo para treino, e não conseguem lidar com problemas que sejam bastante diferentes daqueles para os quais foram treinadas;
- **Árvores de Decisão** [34] [35]: é uma forma simples e expressiva de representar o sistema, podendo ainda ser transformada num conjunto de regras. O principal problema com este tipo de representação é o facto de ser muito sensível ao ruído nos dados de entrada.

Estas técnicas podem ser utilizadas de várias formas. Seguem-se alguns exemplos de possíveis sistemas [17]:

- **Sistemas baseados em Regras**: representam a sua base de conhecimento sob a forma de regras, constituídas por condições e acções. Este tipo de sistemas, embora útil em redes pequenas, é bastante limitado. Os seus principais problemas são a incapacidade de aprender com a experiência, de resolver problemas desconhecidos (ou seja, para os quais não existem regras específicas), interferências não previstas entre as diversas regras, e a difícil manutenção em ambientes dinâmicos e de larga escala;
- **Sistemas baseados em Modelos**: apresentam um modelo que inclui a descrição da estrutura (estático) e comportamento (dinâmico) do sistema. As condições, associadas às regras, contêm predicados que fazem referência ao modelo, de modo a testar relações entre componentes. Este tipo de sistemas, embora mais preparado para lidar com problemas desconhecidos, introduz novos desafios, nomeadamente a dificuldade em obter e manter actualizado um modelo completo do sistema;
- **Sistemas baseados em Experiência**: estes sistemas tomam decisões com base em situações e decisões feitas no passado. São capazes de aprender a lidar com problemas desconhecidos, o que os torna bastante resilientes a alterações no

sistema. No entanto, necessitam de modelos muito específicos, e a sua execução é bastante demorada.

## 2.5.2 Propagação de falhas

---

A utilização de Propagação de falhas consiste na utilização de grafos para realizar o mapeamento entre alarmes e falhas. Existem, essencialmente, dois tipos de grafos que são utilizados [17]. Este mapeamento permite identificar qual a falha que melhor justifica os alarmes recebidos. As grandes desvantagens deste tipo de abordagem, é a necessidade de se ter, *a priori*, o total conhecimento do sistema (para a construção dos grafos), e o mau desempenho perante problemas desconhecidos (que não estejam previstos no modelo construído).

Existem, essencialmente, dois tipos de grafos que são utilizados:

- **Grafos de Dependências:** são grafos direccionados que permitem, com base nas dependências entre os vários objectos, fazer a análise de alarmes e catalogá-los como primários ou secundários;
- **Grafos de Causa-Efeito:** são grafos direccionados acíclicos que estabelecem relações causais entre os vários alarmes. Ou seja, dados dois alarmes, é possível verificar se existe a possibilidade de um deles ter sido originado como consequência do outro.

Existem muitas técnicas que podem ser construídas com base nestes grafos. Vamos aqui apresentar aquelas que são consideradas as mais importantes:

- **Sistemas baseados em Códigos:** cada falha é representada por um código, que corresponde a um vector de N posições. Cada posição do vector representa um sintoma (alarme) de uma dada falha. Num contexto determinístico, o valor de cada posição destes vectores pode assumir um valor de 0 ou 1, consoante o alarme está ou não está presente aquando da falha. Num contexto probabilístico, cada posição do vector assume um valor entre 0 e 1, que representa a probabilidade desse sintoma estar presente quando ocorre a falha.

Os sintomas podem ainda ser divididos em dois grupos: *significativos* e *candidatos*. Um sintoma é considerado *significativo* caso esteja acima dum dado limiar de probabilidade. Uma falha só é identificada quando tiverem sido observados todos os seus sintomas significativos. Os sintomas candidatos são opcionais, pelo que podem ser, ou não ser, observados.

Estes códigos são a uma representação matricial optimizada, dum grafo bipartido de causa-efeito;

- **Redes Bayesianas:** utilizam grafos direccionados acíclicos, cujos nós (variáveis aleatórias) representam o estado da rede ou a ocorrência de eventos. Neste contexto, o problema de identificação da falha, pode ser visto como o de calcular a falha mais provável na rede bayesiana. O principal problema desta técnica, é o requerer um grande poder computacional quando aplicado a um grande número de nós;
- **Gramáticas livres de contexto:** a principal característica das gramáticas livres de contexto, é o de permitirem construir expressões a partir de sub-expressões. Esta característica pode ser utilizada para construir um modelo hierárquico do sistema, que representada as dependências entre objectos. O uso de gramáticas livres de contexto tem, no entanto, o problema dos algoritmos utilizados serem bastante complexos.

Com o intuito de garantir as métricas faladas nos objectivos, obriga a que se recorra a uma destas técnicas reduzindo assim de uma forma muito significativa o número de mensagens que circulam assim na rede. A abordagem seguida para aplicar a correlação de eventos, foi a utilização da Inteligência Artificial recorrendo a um Sistemas baseados em Regras. Esta abordagem é de todas a mais simples de se perceber e implementar. Tal como o nome indica representa a sua base de conhecimento sob a forma de regras, constituídas por condições e acções, ou seja quando acontece algo é despoletada uma determinada acção, por exemplo uma situação de acidente entre vários veículos.

Uma correlação de eventos eficiente e precisa é essencial para reduzir os custos de funcionamento e para melhorar a disponibilidade, permitindo melhorar o desempenho dos serviços na rede.

## 2.6 Protocolos

---

A transferência de informação numa rede veicular de emergência é um aspecto crítico, pelo que será necessário identificar qual o tipo de protocolo de encaminhamento mais vantajosos na resolução dos problemas do cenário que se pretende estudar. Os protocolos de encaminhamento para VANETs podem ser classificados segundo o seu tipo, englobando quatro classes fundamentais:

- **Protocolos Topológicos** caracterizam-se por encontrar o melhor caminho entre dois pontos origem-destino (em modo unicast), através do uso da topologia de rede.

- **Protocolos Geográficos** caracterizam-se por encontram o melhor caminho origem-destino (em modo unicast) com base na posição geográfica dos nós, o que obriga a que cada nó tenha de possuir de um sistema de localização, tal como o GPS.

- **Protocolos Oportunísticos** caracterizam-se pela possibilidade de manter a informação durante um certo período de tempo, e voltar a transmitir quando for possível, recorrendo ao método carry-and-forward (em modo unicast).

- **Protocolos de Disseminação**, que se caracterizam por transmitir a informação em modo broadcast e ou geocast, que é essencial para diversas aplicações em redes veiculares, como por exemplo aplicações ao nível da segurança rodoviária.

### 2.6.1 Protocolos Topológicos

---

Caracterizam-se por encontrar o melhor caminho entre qualquer par de nós origem-destino através do uso de informação da topologia da rede. Estes protocolos podem ainda ser divididos em:

- **Proactivos**, que mantêm uma lista actualizada das rotas para cada um dos nós da rede.

- **Reactivos**, que só constroem as rotas quando existe informação para enviar.

- **Híbridos**, como o nome indica, usam um pouco de cada uma das técnicas usadas pelos protocolos anteriores.

Alguns exemplos de protocolos desta classe são o *Prediction-Based Routing (PBR)* [12], *Ad hoc On-Demand Distance Vector (AODV)* [18] e *Dynamic Source Routing (DSR)* [19].

Em suma, como os protocolos topológicos se caracterizam por encontrar o melhor caminho entre dois pontos origem-destino, através do uso da topologia de rede, são considerados inadequados pelo facto da topologia ser bastante dinâmica numa VANET. São também inadequados para este cenário da rede de emergência, uma vez que, na maioria das situações não existe um destinatário definido à priori, mas sim a necessidade de propagação de mensagens numa determinada área, como forma de fornecer um aviso efectivo aos condutores. Se bem que poderá ser equacionado, no caso específico de ser necessário enviar as mensagens com destino às autoridades.

### 2.6.2 Protocolos Geográficos

---

Os protocolos de encaminhamento geográfico caracterizam-se por encontrar o melhor caminho origem-destino com base na posição geográfica dos nós, o que obriga a que cada nó tenha de possuir de um sistema de localização, tal como o GPS. Nesta gama de protocolos, um nó envia os dados em direcção à localização do nó de destino por múltiplos saltos. Para tal acontecer, o nó precisa de conhecer a posição dos seus vizinhos, utilizando um serviço de localização, por exemplo o Reactive Location Service (RLS). O objectivo principal nesta gama de protocolos, é permitir uma grande escalabilidade em ambientes de alta mobilidade, já que não é necessário manter informação sobre as rotas de todos os nós da rede, porque esta informação é determinada no momento em que os pacotes têm de ser transmitidos ou retransmitidos.

De entre os vários protocolos desta classe os mais conhecidos são o *Greedy Perimeter Stateless Routing (GPSR)* [20], *Geographic Source Routing (GSR)* [21], *Anchor-Based Street and Traffic Aware Routing (A-STAR)* [23] e o *Connectivity-Aware Routing (CAR)* [4].

Os protocolos geográficos tiram partido do conhecimento da sua posição através de dispositivos como o GPS para construir a topologia de rede e desta forma calcular as rotas. Mais uma vez, estamos perante uma classe de protocolos que tenta calcular as rotas óptimas para um par origem-destino, com a vantagem de usarem informação adicional fornecida pelo GPS. Dentro desta classe, o algoritmo "CAR" apresenta funcionalidades apetecíveis mas é demasiado pesado a nível de computação. Comparativamente com os topológicos, estes parecem ser melhores no caso específico do envio de mensagens para as autoridades.

### 2.6.3 Protocolos Oportunísticos

---

Estes protocolos aparecem com o intuito de tentar resolver problemas das constantes perdas de ligação por parte dos nós e caracterizam-se pela possibilidade de transportar/guardar a informação durante um certo espaço de tempo, e voltar a transmitir quando for possível, recorrendo ao método carry-and-forward.

De entre os vários protocolos existentes, os mais importantes são o *Motion Vector (MoVe)* [24], *MaxProp* [6], *Scalable Knowledge-Based Routing (SKVR)* [25] e o *Assisted Data Delivery (VADD)* [5].

A classe dos protocolos Oportunísticos é tal como as outras, uma classe que calcula as rotas ponto-a-ponto (unicast) e como tal não se adequa de todo aos problemas apresentados. Mas apresenta uma grande vantagem, que se deve tomar em consideração que é a possibilidade de tirar partido da conectividade temporária entre os diversos nós da rede para o encaminhamento das mensagens, assim como pelo uso do método "carry-and-forward".

### 2.6.4 Protocolos de Disseminação

---

A disseminação de informação é essencial para diversas aplicações em redes veiculares, como por exemplo aplicações ao nível da segurança rodoviária. Quando ocorre um acidente num ambiente rodoviário, mensagens de alarme devem ser transmitidas para informar todos os outros veículos. No entanto, os protocolos clássicos como os Topológicos, Geográficos e Oportunísticos que funcionam em modo Unicast, em VANETs, não podem ser utilizados porque estes protocolos têm primeiro que descobrir as rotas e construir cada uma delas para cada nó vizinho. Essa tarefa consome demasiado tempo daí que não pode ser adaptado para a divulgação de informações críticas, como evitar uma situação de colisão. É assim essencial que a informação chegue ao maior número de nós o mais rapidamente possível, sendo estes protocolos caracterizados por enviar a informação, em modo broadcast e ou geocast.

De entre os vários protocolos existentes, dois deles destacam-se na sua utilização e pela sua diferença. O *Urban Multi-hop Broadcast protocol (UMB)* [26] e o *Optimized Dissemination of Alarm Messages (ODAM)* [27].

Como conclusão, esta classe de protocolos é a que mais se adequa ao problema em estudo uma vez que permite chegar ao maior numero de nós o mais rapidamente possível, sendo em modo *Broadcast* e *ou Geocast*. Desta forma é possível avisar o maior número de nós (veículos) de situações de acidentes, ao mesmo tempo que aumenta a probabilidade e diminui o tempo da informação chegar as forças de autoridade.



---

## 3 Proposta de Solução

---

Tendo em conta os vários cenários de utilização, o ambiente, o uso de políticas de escalonamento e correlação de eventos descreve-se a visão geral da arquitectura a implementar nomeadamente os diferentes cenários em que funcionara o modelo, descrevendo também a arquitectura e os requisitos temporais que a solução deverá satisfazer de forma a ter utilidade na prevenção de acidentes.

Nesta proposta de solução dois cenários são apresentados, o primeiro representa o envio de uma mensagem de acidente por parte de um veículo acidentado e o outro refere-se a um acidente que envolve vários veículos onde são geradas mensagens de acidente em cada veículo. Cada cenário apresenta problemas diferentes que têm de ser resolvidos, recorrendo ao uso de prioridade e correlação de eventos.

### 3.1 Requisitos temporais

---

Numa situação de acidente e no aviso da sua ocorrência o factor tempo é determinante. Para que os objectivos sejam cumpridos a arquitectura deverá respeitar tempos de entrega que permitam, que os condutores consigam em segurança evitar novas colisões dos seus veículos. Tendo em conta que a maioria dos acidentes rodoviários devem-se a várias razões entre elas: conversas através do telemóvel [38], conversar com um passageiro, alterações na rádio etc. Todos estes factores alteram directamente o tempo de reacção numa situação de resposta a uma travagem. Seguidamente irá ser apresentado um estudo simplificado dos requisitos temporais que a solução a desenvolver deve respeitar, de forma a ser eficaz na prevenção de acidentes.

#### 3.1.1 Tempo de reacção e tempo de travagem

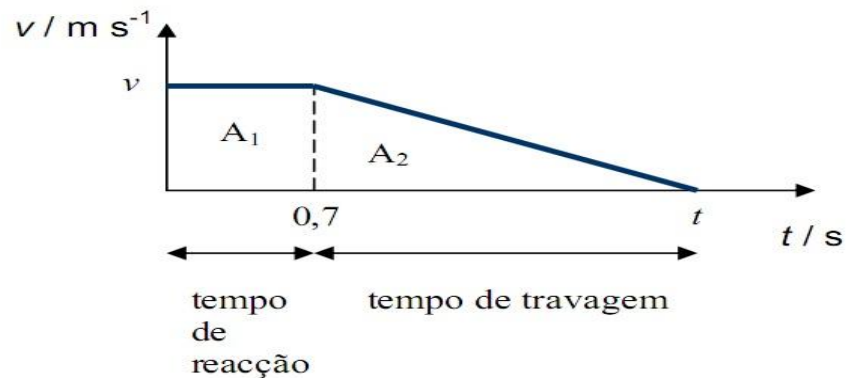
---

O intervalo de tempo que separa o instante em que o condutor detecta o obstáculo, por exemplo um acidente, vendo-o, ouvindo-o ou recebendo uma mensagem que o informa deste, e o instante em que decide travar chama-se **tempo de reacção**. É o tempo que o condutor demora a reagir [40] [41]. Este tempo varia de condutor para condutor, sendo por exemplo influenciado pela idade, existência ou não de álcool no sangue, sonolência etc [38] [39] [42]. Este tempo pode ir dos 0.2s aos 2.0s mas o tempo de reacção media é cerca de 0.7s para um condutor com reflexos normais.

O **tempo de travagem** é o tempo que demora a travagem, isto é o tempo que demora até que o veículo fique completamente imobilizado, depois de começar a travar. Depende da velocidade e da massa do veículo: quanto maior a velocidade e a massa, maior é o tempo que o veículo demora a parar.

### 3.1.2 Distância de travagem e distância de segurança

Para um condutor de reflexos normais, a situação traduz-se graficamente por:



**Figura 2** – Tempo de reacção e travagem

É de notar que:

- Durante o tempo de reacção a velocidade do veículo mantém-se constante, o que significa que o condutor deixa de acelerar antes de começar a travar.
- Durante o tempo de reacção, o veículo percorre uma distância que se pode calcular pela área  $A_1$  do gráfico:

$$A_1 = V \times 0.7s$$

- Durante o tempo de travagem a distância percorrida pode ser calculada através da área  $A_2$  do gráfico – é a **distância de travagem**:

$$A_2 = \frac{(t - 0.7) \times v}{2}$$

- Para o mesmo condutor, tanto a distância de travagem como a distância percorrida durante o tempo de reacção são tanto maiores, quanto maior for a velocidade com que o veículo circula.

À soma da distância percorrida durante o tempo de reacção do condutor e da distância de travagem chama-se **distância de segurança** [43], ou seja, é a distância que é necessário percorrer até o veículo parar sem colidir. Note-se que também é tanto maior quanto maior for a velocidade do veículo e o tempo de reacção do condutor.

Segundo alguns estudos [40] [41] [42], sabe-se que a uma velocidade média de 100km/h ou o seu equivalente em metros 28m/s, um veículo de categoria média e equipado com o sistema de ABS, precisa de 59.4m para efectuar a travagem. Recorrendo assim as fórmulas anteriores:

- A distância necessária para o tempo de reacção é:

$$A1 = 28m \times 0.7s \approx 19m$$

- A distância de travagem é:

$$A2 \approx 59.4m$$

- Somando as duas áreas obtemos a distância de segurança que é:

$$A1 + A2 \approx 78.4 m$$

Fazendo uma análise em termos de tempos face aos metros obtém-se os seguintes valores:

- Tempo de Reacção

$$T1 \approx 0.7s$$

- Tempo de Travagem
  - $A2 = 59.4m$
  - $V = 28m/s$

$$A2 = \frac{(T2 - 0.7) \times v}{2} \approx T2 \approx 4.94s$$

Obtém-se assim um tempo total de travagem:

$$T1 + T2 \approx 5.64s$$

A tabela 1 e 2 apresentam, a distância de reacção, a distância de travagem e a distância de segurança resultante para várias velocidades dos veículos e diferentes condições do piso:

**Tabela 1 – DISTÂNCIA DE SEGURANÇA EM CONDIÇÕES DE PISO SECO**

<b>Velocidade (km/h)</b>	<b>Distância de Reacção (m)</b>	<b>Distância de Travagem</b>	<b>Distância de Segurança</b>
<b>50</b>	9.2	14.8	24.0
<b>60</b>	11.0	21.4	32.4
<b>80</b>	14.7	38.0	52.7
<b>100</b>	18.3	59.4	77.7
<b>120</b>	22	85.5	107.5

**Tabela 2 – DISTÂNCIA DE SEGURANÇA EM CONDIÇÕES DE PISO MOLHADO**

<b>Velocidade (km/h)</b>	<b>Distância de Reacção (m)</b>	<b>Distância de Travagem</b>	<b>Distância de Segurança</b>
<b>50</b>	9.2	26.1	35.2
<b>60</b>	11.0	37.5	48.5
<b>80</b>	14.7	66.7	81.4
<b>100</b>	18.3	104.3	122.6
<b>120</b>	22	150.2	172.2

Em suma e com base nos valores teóricos apresentados neste capítulo, podemos na fase de testes fazer comparações mais realistas à arquitectura, podendo dizer na realidade se é possível fazer o aviso atempado dos condutores para evitar acidentes em cadeia.

## 3.2 Arquitectura da solução

Na figura 3, representam-se os principais blocos da arquitectura proposta, nomeadamente: Aplicação, Conversor de Mensagens, Controlador e Gestor de Comunicação.

O Bloco da Aplicação irá criar a mensagem que corresponde ao novo evento. Após isto ela irá entregar a mensagem ao Conversor de Mensagens. O Bloco Conversor de Mensagens será responsável por converter a mensagem, formatando-a de acordo com o modelo estipulado, no protocolo que se apresenta na secção seguinte.

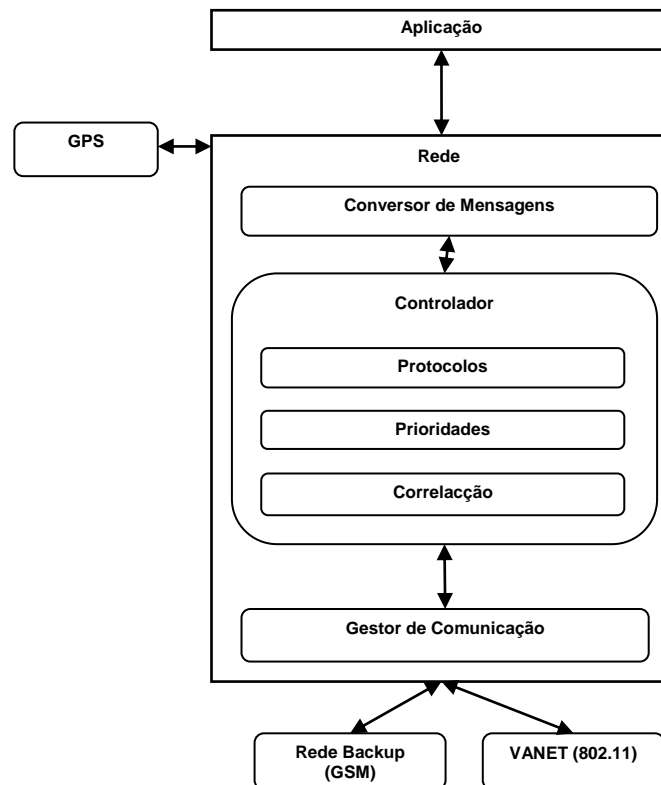
Quando a mensagem chega ao Controlador este é responsável por um conjunto de funções, tais como: decidir a política a ser aplicada à mensagem, definir o protocolo de encaminhamento mais adequado e recorrer ao uso da correlação de eventos, caso necessário.

A política a ser aplicada a mensagem, passa por lhe dar, ou não, prioridade. Quando o sistema se depára com uma mensagem de acidente, esta mensagem recebe prioridade máxima de envio, fazendo com que todo e qualquer outro tipo de tráfego seja relegado para segundo plano. No

caso de uma mensagem de acidente recorre-se ao modo de *broadcast* para o seu envio com o intuito de alcançar o maior número de veículos possível.

O sistema recorre a correlação de eventos quando se depara com mais que uma mensagem de acidente, ou seja quando o acidente envolve mais do que um veículo. Com esta correlação de eventos pretende-se otimizar a solução, garantindo que a informação de acidente chega atempadamente aos diversos veículos. É assim possível independentemente do número de mensagens geradas, apenas uma mensagem de acidente circula na rede.

De seguida a mensagem será entregue ao Gestor de Comunicação que será responsável por enviar a mensagem pelo bloco de rede VANET (802.11) e/ou pelo bloco de rede que funciona como sistema de backup (GSM), de acordo com a política aplicada.



**Figura 3** – Arquitectura de Rede

### 3.2.1 Protocolo

---

Para transferir as mensagens de acidente é usado um protocolo de comunicação, no qual as mensagens têm a seguinte estrutura:

IDmensagem	IDLmensagem	IDveiculo	TTL	IPdestino	PktType	PktSize	GPS
------------	-------------	-----------	-----	-----------	---------	---------	-----

- **IDmensagem**, é um campo que guarda um ID atribuído a cada mensagem gerada pelos veículos que tiveram um acidente. Este número varia entre 1 e 60 que identifica as mensagens que vão ser enviadas pelos veículos durante 60 segundos. Este ID é um valor inteiro respeitando assim o tamanho estipulado para inteiros, normalmente 2 bytes ou 16 bits.
- **IDLmensagem**, é um campo que guarda o ID da última mensagem recebida. É usado para verificação da retransmissão das mensagens, ou seja quando a mensagem já foi transmitida uma vez por um determinado veículo não volta a ser retransmitida evitando assim o fenômeno de *broadcast storm*. Este ID é um valor inteiro respeitando assim o tamanho estipulado para inteiros, normalmente 16 bits.
- **IDveiculo**, é um campo que guarda a identificação do veículo na rede que foi emissor da mensagem de acidente. É com base neste ID que é possível saber o número de veículos envolvidos em um acidente. Este ID é um valor inteiro respeitando assim o tamanho estipulado para inteiros, normalmente 16 bits.
- **TTL**, tempo de vida do pacote.
- **IPdestino**, é o campo que contém o endereço de destino das mensagens, neste caso as mensagens são todas enviadas em modo *broadcast*, significando que o destinatário são todos os nós que se encontrem dentro de um determinado raio.
- **PktType**, este campo contém o tipo de pacote que vai ser transmitido na rede, neste caso o pacote é sempre do tipo MYPBCAST.
- **PktSize**, este campo contém o tamanho do pacote do tipo MYPBCAST.
- **GPS**, este campo guarda as coordenadas GPS do veículo, que gerou a mensagem de acidente pela 1ª vez.

### 3.2.2 Modelo Base

---

Sabe-se *à priori* que existem muitas soluções com desempenho minimamente satisfatório em cada um dos cenários em estudo. Partindo disto, o modelo base apresenta uma solução simples do envio de uma ou várias mensagens de acidente.

O envio desta mensagem recorre ao protocolo descrito na secção anterior, que funciona em modo *broadcast*. Em situação de acidente uma mensagem é enviada a cada segundo durante sessenta segundos, pelo veículo acidentado, e todos os veículos dentro de um determinado raio do acidente irão receber a mensagem. Por sua vez estes irão proceder à sua retransmissão de novo em

modo *broadcast*. No caso de o acidente envolver mais que um veículo então cada veículo enviará a sua mensagem de acidente, de forma independente.

Sendo a comunicação em modo *broadcast* não é do interesse que esta mensagem se propague indefinidamente enquanto houver veículos. Para evitar esta situação, foram utilizadas coordenadas GPS dos veículos. Quando uma mensagem de acidente atinge uma determinada distância do emissor, a mensagem é descartada.

Com o intuito de evitar o fenômeno de *broadcast storm*, este modelo atribui a cada mensagem e a cada veículo um identificador que irá evitar a retransmissão da mesma, caso um determinado veículo já a tenha retransmitido pelo menos uma vez.

Este modelo base, apresenta várias fragilidades que não podem ser ignoradas. Estas fragilidades passam por não existir prioridades relacionadas com o tipo de mensagem, levando a que mensagens de acidente tenham a mesma prioridade que mensagens que se refiram por exemplo a video. Outra fragilidade é a não existir um controle do número de mensagens do mesmo acidente que circulam na rede, levando em última caso a uma saturação completa da largura de banda. Com o intuito de resolver estas fragilidades são apresentados de seguida modelos que cobrem os problemas mais importantes.

### 3.2.3 Modelo com prioridades

---

O que se pretende com este modelo de prioridades, é garantir que as mensagens de acidente têm prioridade face a qualquer outro tipo de tráfego que passe na rede. Isto não acontece no modelo base. Deste modo, este modelo deverá suportar a existência de duas filas distintas, uma prioritária que tratará o tráfego referente as mensagens de acidente e outra responsável por todo o restante tráfego que circula na rede. Existe assim um escalonador que verifica o conteúdo de cada fila, e sempre que a fila das mensagens de acidente tiver algo para enviar, esta tem sempre prioridade face a outra. Assegura-se desta forma que o tráfego de emergência será atendido primeiro que qualquer outro tipo de tráfego.

### 3.2.4 Modelo com correlação

---

Com o intuito de garantir que a informação de acidente atinge todos os interessados nomeadamente situações de acidente em cadeia, autoridades, e por último o aviso de condutores a longa distância para que possam seguir outro caminho evitando assim filas de trânsito, obriga além do uso do modelo de prioridade um outro modelo este com correlação de eventos reduzindo de uma forma muito significativa o número de mensagens que circulam assim na rede.

Num caso de acidente que englobe vários veículos, o número de mensagens na rede tende a crescer exponencialmente. O que se pretende assim neste modelo recorrendo à correlação das mensagens de acidente, é uma redução acentuada do número de mensagens a circular na rede.

Esta correlação foi pensada de várias maneiras:

1. Esperar algum tempo antes de enviar a mensagem de acidente, fazendo com que após um tempo  $X$  um determinado veículo soubesse todos os veículos do acidente e enviasse então apenas uma única mensagem com a informação do acidente, incluindo todos os veículos acidentados e as suas coordenadas GPS etc.
2. Inicialmente não existir correlação, e mais tarde, diga-se alguns segundos depois, ser feita a correlação destas mensagens à medida que estas iam sendo transmitidas pela rede.
3. Tentar usar o melhor de cada uma das opções anteriores e tentar criar uma metodologia de correlação que suportasse todas as situações previstas.

A metodologia usada foi então a seguinte:

No instante em que ocorre o acidente as mensagens são geradas e transmitidas na rede. Esta situação só acontece no primeiro segundo, ou seja, no instante em que ocorre o acidente, e a primeira mensagem é enviada pelos veículos que tiveram o acidente. Caso o acidente envolva mais que um veículo então recorre-se ao uso da correlação de eventos. A correlação é feita entre o instante em que é transmitida a primeira e a segunda mensagem.

O número de veículos que iram receber estas mensagens varia constantemente podendo então aumentar ainda mais o número de mensagens na rede de forma exponencial. Assim com o intuito de evitar o fenómeno conhecido por *broadcast storm* estas mensagens são identificadas por um par de identificadores – veículo e mensagem. Assim sendo, cada veículo só irá retransmitir uma mensagem caso nunca tenha transmitido uma mensagem cujo par seja exactamente o mesmo.



---

## 4 Implementação

---

Tendo em conta a natureza desta tese, é necessário efectuar uma análise de algumas das configurações possíveis do modelo, de forma a atingir o melhor resultado possível. Desta forma decidiu-se adoptar a simulação como ambiente de testes. Utilizou-se a ferramenta de simulação NS-2, inicialmente criou-se um cenário simples para familiarização com a plataforma, seguindo-se uma implementação que modelasse a implementação do modelo previamente apresentado.

### 4.1 Análise de Simuladores

---

Para testar se a solução apresentada vai resolver os problemas reais e trazer as contribuições que se deseja, terá que se recorrer a um simulador, uma vez que não é viável reproduzir os testes que se pretendem realizar num ambiente real, a fim de se poder analisar os resultados. Sendo o simulador uma peça chave para a possibilidade da realização dos testes, um passo importante é escolher o simulador mais certo a usar. Existe uma vasta área de simuladores disponíveis que entre os mais conhecidos encontram-se o NS-2, NS-3, GloMoSim, Qualnet e OPNET. Com base nisto então a escolha deve ser feita tendo em conta:

- Qualidade da documentação disponível e de apoio;
- Facilidade de uso;
- Suporte das funcionalidades desejadas;
- Capacidade e viabilidade do desenvolvimento de novos módulos e alteração dos existentes;
- Se o software é opensource ou não;
- Se o software é amplamente testado, para existir confiabilidade do simulador.

Na tabela 3 é possível observar a percentagem de utilização de cada simulador em MANET's que são os mesmos simuladores utilizados para as VANET's. As informações apresentadas na tabela 3 foram reunidas a partir de [44].

**Tabela 3 – PERCENTAGEM DO USO DE CADA SIMULADOR**

<b>Simulador</b>	<b>Percentagem de Uso</b>
NS-2	43.8%
GloMoSim	10.0%
Qualnet	6.3%
OPNET	6.3%

Segundo os autores de [45] onde foram realizados testes exaustivos aos simuladores apresentados na tabela 3, tanto o GloMoSim como o OPNET são simuladores que apresentam uma dificuldade no desenvolvimento de novos módulos, além do que também é referido que no OPNET a depuração de erros é uma tarefa difícil. Uma análise importante também referida no artigo é o facto de apenas o NS-2 e o Qualnet têm implementações de routing geográfico que é muito importante. Uma nota interessante é o facto de apesar de o Qualnet ter apenas 6,4% de uso em testes realizados em trabalhos relacionados, ele é usado por muitos operadores de rede ao longo do mundo.

A escolha fica assim entre dois simuladores o NS-2 e o Qualnet. Sabendo a partida que o Qualnet não é open source, seria então um factor de descarte, mas por outro lado, o NS-2 não tem facilidade de no seu uso o que também é um factor de descarte. Com base nos factores apresentados ficou decidido que devido ao facto de o NS-2 ser um simulador open source e bastante usado em trabalhos relacionados com MANET's e VANET's, foi o escolhido para a realização dos testes desta tese. Com esta conclusão é então de seguida feito um estudo exaustivo do simulador NS-2 Simulador NS-2.

Tendo em conta a necessidade de efectuar testes à solução que irá ser proposta nesta tese, decidiu-se estudar o simulador NS-2, de forma a verificar a viabilidade de utilizar esta plataforma para testes. Pois, uma simulação é um meio bastante eficiente para a realização de testes e desenvolvimento de novas tecnologias e é uma técnica onde dispositivos de uma rede real, como hosts, routers, switches, entre outros, são modelados de maneira a apresentar comportamento semelhante aos dos dispositivos reais correspondentes.

O *Network Simulator* é um simulador de eventos discretos e orientado a objectos, que embora seja uma ferramenta de simulação usada há anos sobretudo no meio académico, ainda é um software em constante desenvolvimento, permitindo a qualquer programador desenvolver novos modelos e integra-los facilmente no sistema base. Existem várias versões em uso actualmente sendo a versão ns-2.34 [32] a usada nesta tese.

O NS-2 é um simulador que foi desenvolvido utilizando duas linguagens de programação para dois propósitos distintos. O núcleo do NS-2 é escrito em C++, o que lhe confere uma maior facilidade na especificação de protocolos ao nível de bits/bytes, garantindo uma execução rápida característica da linguagem. Para facilitar a aprendizagem por parte do utilizador, o NS-2 usa a linguagem de script OTcl [36] como linguagem de comandos e de configuração de modelos para simulações. Com a escolha das linguagens citadas, os criadores do NS-2 conseguiram de uma forma prática que todos os utilizadores pudessem de uma forma mais ou menos simples adicionar novas funcionalidades ou alterar as existentes, adaptando o simulador às suas necessidades. A interface de comandos e configurações do NS-2 interpreta comandos OTcl. Estes comandos podem ser fornecidos interactivamente pelo utilizador, de forma semelhante à linha de comando de um sistema Linux/UNIX (bash, sh, csh, entre outros), ou na forma de *scripts* OTcl, que são constituídos por linhas de código

em sequência, de forma a descrever completamente um modelo a ser simulado. Assim, o OTcl funciona como um *front-end* mais amigável e prático para o utilizador interagir com o simulador.

Para que seja possível a utilização dessas duas linguagens, OTcl e C++, existe no NS-2 uma correspondência entre objectos na interface OTcl e objectos C++. Esses objectos compilados são disponibilizados ao interpretador OTcl através de um interligação entre a biblioteca OTcl e os componentes compilados em C++, criando um objecto correspondente no interpretador OTcl para cada.

Recorrendo assim a estas linguagens, o NS-2 oferece um grande número de funcionalidades. Entre as várias funcionalidades temos, vários tipos de redes como redes cabladas (Ethernet) e redes sem fios (Wireless), no qual para as redes Wireless o suporte só aparece nas últimas versões do NS-2. São suportados também vários protocolos tanto de nível 2 como de nível 3 do modelo TCP/IP. Com isto surge a necessidade de se representar um modelo real no simulador e para isso o NS-2 oferece a capacidade de se criar uma topografia. A topografia define um tipo de terreno e uma área limite para a localização dos nós, a qual é especificada em metros quadrados. Sendo assim, as coordenadas X e Y que definem as dimensões devem estar em metros.

Tendo um cenário criado, o NS-2 permite-nos gerar tráfego que corra sobre a topografia criada. Para isto o NS-2 oferece uma funcionalidade importante que é o gerador de tráfego automático, que surgiu com o intuito de auxiliar a realização de simulações com várias configurações de posição, movimento e tráfego entre nós em uma rede. A equipa do Projeto Monarch desenvolveu dois programas que estão disponíveis no NS-2: o *setdest* e o *cbrgen.tcl*.

Ambos os programas estão localizados no directório *ns-2.xx/indep-utils/cmu-scen-gen*. O *setdest* é um gerador de cenários, ou seja, posições e movimentos dos nós, desenvolvido em C++. O *cbrgen.tcl* é usado para gerar padrões de tráfego tanto para fluxos UDP simulando aplicações do tipo CBR quanto TCP simulando a aplicação de transferência de arquivos FTP.

Durante o decorrer da simulação, um ficheiro de *log* é gerado armazenando informações relevantes da rede. Este ficheiro vai ter o nome definido pelo utilizador no seu script OTcl. Este ficheiro possibilitará uma análise do que ocorreu durante toda a simulação assim que esta termine.

O NS-2 disponibiliza dois formatos básicos de *trace* para simulações sem fios: o antigo formato do Projeto Monarch, também conhecido como *CMU-Trace*, e um novo formato de *trace* o qual foi desenvolvido como um esforço para integrar o antigo formato de *trace* com o formato de *trace* genérico, ou seja, para qualquer tipo de simulação, do NS-2. No entanto em relação ao novo formato este apenas está disponível para as simulações de redes sem fios.

Além do ficheiro gerado anteriormente também é possível produzir outro tipo de ficheiro de *log*, direccionado e criado especificamente para a análise visual do comportamento da rede. Este ficheiro é utilizado pelo software NAM (*Network Animator*) e possibilita uma visualização gráfica do

comportamento da rede ao longo do tempo, permitindo avaliar artefactos diversos, tais como: movimentação de nós, transmissão de pacotes de várias fontes de tráfego, perdas de pacotes e alcance de sinal numa rede sem fios.

#### 4.1.1 Exemplo de utilização do NS-2 em Redes Móveis

---

O modelo de simulações para redes móveis utilizado pelo NS-2 foi desenvolvido por investigadores do Projeto Monarch [31] (*Mobile Networking Architectures*) originalmente pertencente à CMU (*Carnegie Mellon University*) e agora pertencente à *Rice University* [31]. Inicialmente, esse modelo foi adoptado apenas como uma extensão do NS-2. Actualmente o modelo é parte integrante do NS-2 e tem sido continuamente melhorado ao longo dos anos. Os componentes principais deste modelo são o *Mobile Node*, novos mecanismos de *routing* e componentes de rede que são usados para configurar o *Mobile Node*.

#### 4.1.2 Mobile Node

---

Um nó da classe *Mobile Node* possui as mesmas características de um nó da classe *Node* e os componentes de configuração também são idênticos, mas necessita de alguns parâmetros de configuração adicionais como a de um canal de comunicação sem fios e a capacidade de realizar movimento.

##### 4.1.2.1 Componentes de Redes para configurar o Mobile Node

---

Sendo que todos os nós são móveis é necessário explicar os seus componentes no simulador. Os componentes principais de configuração de um *Mobile Node* são [32] [37]:

- **Canal de Comunicação (Channel):** Define o tipo de canal de comunicação que se vai usar.
- **Interface de Rede:** Define o tipo de Interface de Rede usada, neste caso sendo *Mobile Node* será uma Interface *Wireless*
- **Modelo de Propagação de Ondas de Rádio:** Estes modelos dizem a potência do sinal recebido pelo receptor. Esta intensidade do sinal é usada para determinar se o frame é transmitido com sucesso. Existem três modelos de propagação, *Free Space*, *Two Ray Ground*, e *Shadowing*.
- **Protocolo MAC:** Define o protocolo ao nível MAC para se usar no simulador NS-2
- **Fila da Interface de Rede:** Define o tipo de fila utilizada, por exemplo se é ou não uma fila com prioridades.
- **Modelo de Antena:** Define o tipo de antena usada ao longo da simulação.

Os componentes referidos anteriormente são aqueles que variam para um nó do tipo *Node*, comparativamente a um *Mobile Node*.

Outro componente necessário à configuração do *Mobile Node* é a topografia (ou, em outras palavras, o terreno) onde os nós estão dispostos e sobre o qual se vão deslocar.

Após então a configuração do *Mobile Node*, deve-se realmente criar os nós da rede. A título de exemplo este procedimento está descrito na figura 4.

```
for {set i 0} {$i < $val(nn)} {incr i} {  
    set node_($i) [$ns_ node]  
    $node_($i) random-motion 0           ;# disable random motion  
}
```

**Figura 4** – Criação dos nós da rede

Os nós *Mobile Node* previamente configurados são criados com auxílio da função `node` da classe `Simulator`. A segunda linha dentro do ciclo “for” indica que estes nós não devem realizar movimentos aleatórios.

A configuração da posição e movimento dos nós de uma rede sem fios está directamente relacionado com o cenário desejado para a simulação. Os *Mobile Node* foram projectados e implementados para se movimentar numa topologia tridimensional, no entanto, a terceira dimensão ainda não é usada pelas simulações realizadas com o NS-2 [32].

Existem basicamente duas formas de se especificar posição e movimento de um nó: configurar o nó para realizar movimento aleatório e configurar manualmente sua posição inicial e destino numa movimentação com o uso de coordenadas.

Como se verifica na figura 4, os *Mobile Node* foram a título de exemplo configurados para não terem um movimento aleatório, sendo então a sua posição inicial e final configurada à mão. Este ultimo tipo de configuração é exemplificada na figura 5.

```
$node_(0) set X_ 91.731140362117  
$node_(0) set Y_ 23.531104229966  
$node_(0) set Z_ 0.000000000000  
  
$ns_ at 40.000000000000 "$node_(0) setdest 92.12 24.94 4.0"
```

**Figura 5** – Posição e Movimento dos nós da rede

Com isto então sendo *Mobile Node* falta atribuir movimento aos nós. Conforme anunciado anteriormente, a posição de um nó deve ser configurada apenas através das coordenadas X e Y, sendo a coordenada z configurada como zero. A definição do movimento é feita da forma como se

pode verificar pela figura 5, onde o nó 0 é movimentado para o ponto (92,12;24,94) do plano definido pela topografia da simulação, com uma velocidade de 4,0 m/s.

### 4.1.3 Mecanismos de Encaminhamento

---

Com o aparecimento dos *Mobile Node* no NS-2 e com estes o suporte para a comunicação sem fios (Wireless) apareceu a necessidade de se desenvolver novos mecanismos de routing para este tipo de redes Ad-Hoc mais especificamente Vanets. As últimas versões do NS-2 apresentam assim quatro protocolos de routing para estas redes.

Os algoritmos de *routing ad hoc* implementados no NS-2 são [32]:

- AODV (*Ad hoc On-demand Distance Vector*);
- DSR (*Dynamic Source Routing*);
- DSDV (*Destination Sequence Distance Vector*);
- TORA (*Temporally Ordered Routing Algorithm*).

O protocolo de routing AODV foi o escolhido para gerar tráfego de **Comunicações de Voz (CBR) ou Troca de Ficheiros (TCP)** na simulação, é um protocolo de routing ad hoc que funciona como uma combinação dos protocolos DSR e DSDV. Utiliza uma estrutura básica de descoberta e manutenção de rotas do protocolo DSR e usa *routing hop-by-hop*, números de sequência e beacons (pacotes especiais contendo informações genéricas do nó sem fios) do protocolo DSDV.

## 4.2 Exemplo da criação de uma Vanet

---

Nesta simulação inicial o que se pretende é mostrar é o uso, da forma mais simples possível, de alguns dos componentes falados anteriormente.

A criação de uma VANET começa pela configuração detalhada de todos os componentes obrigatórios da simulação, como se pode verificar pela figura 6.

```

# =====
# Define options
# =====
set val(chan)          Channel/WirelessChannel    ;# channel type
set val(prop)          Propagation/TwoRayGround   ;# radio-propagation model
set val(netif)         Phy/WirelessPhy           ;# network interface type
set val(mac)           Mac/802_11                ;# MAC type
set val(ifq)           Queue/DropTail/PriQueue   ;# interface queue type
set val(ll)            LL                        ;# link layer type
set val(ant)           Antenna/OmniAntenna       ;# antenna model
set val(ifqlen)        50                        ;# max packet in ifq
set val(nn)            3                         ;# default number of mobilenodes
set val(rp)            AODV                      ;# routing protocol
set val(x)             100.0                    ;
set val(y)             20.0                     ;
set val(simtime)       100.0                    ; #sim time

# =====

```

**Figura 6** – Configurações da simulação exemplo

Os componentes principais de configuração de um *Mobile Node* são [32] [37]:

- **Canal de Comunicação (Channel):** Define o tipo de canal de comunicação que se vai usar.
- **Interface de Rede:** Define o tipo de Interface de Rede usada, neste caso sendo *Mobile Node* será uma Interface *Wireless*
- **Modelo de Propagação de Ondas de Rádio:** Estes modelos dizem a potência do sinal recebido pelo receptor. Esta intensidade do sinal é usada para determinar se o frame é transmitido com sucesso. Existem três modelos de propagação, *Free Space*, *Two Ray Ground*, e *Shadowing*.
- **Protocolo MAC:** Define o protocolo ao nível MAC para se usar no simulador NS-2
- **Fila da Interface de Rede:** Define o tipo de fila utilizada, por exemplo se é ou não uma fila com prioridades.
- **Modelo de Antena:** Define o tipo de antena usada ao longo da simulação.
- A variável "**set val(nn)**" define o número total de nós da simulação, neste exemplo simples apenas foram utilizados três.
- A variável "**set val(rp)**" refere-se ao protocolo usado nesta simulação, neste caso o AODV, que funciona como uma combinação dos protocolos DSR e DSDV. Utiliza uma estrutura básica de descoberta e manutenção de rotas do protocolo DSR e usa *routing hop-by-hop*, números de sequência e *beacons* (pacotes especiais contendo informações genéricas do nó sem fios) do protocolo DSDV.

As duas variáveis seguintes fazem parte da definição do cenário de testes, o seu tamanho nas duas grandezas X e Y, e a última variável dá-nos o tempo total que dura a simulação em segundos.

É necessário criar agora os nós e definir a sua posição no cenário de testes. Na figura 7, os três nós definidos em cima, apresentam posições diferentes em X, ou seja estão dispostos ao longo do cenário nas posições 0, 40 e 70m respectivamente. A variável Y define a distância lateral pelo qual eles estão separados.

```
#
# nodes: 3
#
$node_(0) set X_ 0.0
$node_(0) set Y_ 0.0
$node_(0) set Z_ 0.0

$node_(1) set X_ 40.0
$node_(1) set Y_ 5.0
$node_(1) set Z_ 0.0

$node_(2) set X_ 70.0
$node_(2) set Y_ 5.0
$node_(2) set Z_ 0.0
```

**Figura 7** – Disposição dos nós na topografia

De seguida foi gerado o tráfego que vai passar entre os três nós criados. A título de exemplo foi criada apenas uma comunicação CBR simples: comunicação de voz entre o veículo 2 e o veículo 0, sendo o nó 1 apenas um retransmissor dos pacotes.

```
# 50 pacotes por segundo
set cbr_2_0 [cbr-traff $node_(2) $node_(0) 64 0.02]
$ns_ at 1 "$cbr_2_0 start"
$ns_ at 15 "$cbr_2_0 stop"
```

**Figura 8** – Comunicação CBR entre dois nós móveis

Como se pode ver pela figura 8, a comunicação é feita entre o veículo 2 e o veículo 0, onde são enviados pacotes, com um pacote de 64bytes a uma taxa de 50 pacotes por segundo ( $50 * 0.02 = 1s$ ). Esta comunicação dura 15segundos que pela análise da figura se pode verificar que começa no 1segundo e termina no 15segundo.

Além da especificação do tráfego que é usado pelo simulador, existem outros campos de configuração importantes que são usados para que toda a simulação fosse possível. O *General Operations Director* (normalmente definido pela sigla GOD) é um objecto utilizado pelo NS-2 para armazenar informações globais sobre o estado do cenário, nomeadamente a configuração da rede e os nós que nela estão presentes. Este objecto actua fornecendo uma visão completa da rede, semelhante aquela que um observador onisciente teria, sem que este observador fosse parte integrante desta rede.



Quando se termina uma simulação é necessário guardar toda a informação num ficheiro de *log*. Apesar de ainda se utilizar massivamente um formato antigo, neste trabalho optou-se por utilizar o novo formato *de trace*. Esta escolha foi feita devido ao novo formato fornecer novas informações face ao antigo, informações essas que são indispensáveis para os testes que se pretendem realizar, e que contêm por exemplo dados sobre o movimento dos nós e a sua posição global. O uso de um ou outro formato é especificado no ficheiro de simulação como mostra a figura 9.

```
49 #
50 # Initialize Global Variables
51 #
52 set ns_ [new Simulator]
53 set tracefd [open trace.tr w]
54 $ns_ trace-all $tracefd
55 $ns_ use-newtrace|
```

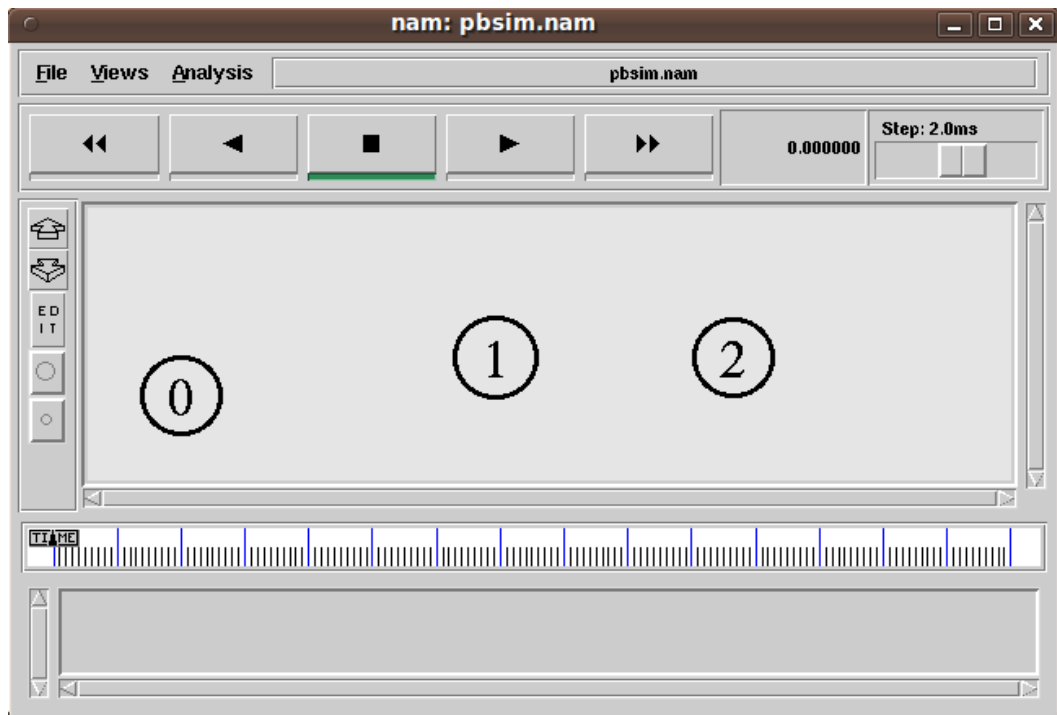
**Figura 9** – Uso do novo formato de trace (*newtrace*)

A figura 9 fornece ainda várias informações úteis. A linha 52 apresenta um comando que está presente em toda simulação e é responsável por instanciar um objecto do tipo *Simulator* e atribui-lo à variável *ns\_*. O objecto *Simulator* é responsável por controlar os eventos da simulação, assim como tratar dos parâmetros da simulação como por exemplo o formato dos pacotes. Possui também várias funções fundamentais para qualquer simulação, como por exemplo a especificação de opções de exibição de animações pelo NAM.

A linha 53 é o comando responsável por especificar o ficheiro onde vão ser registados todos os exemplos da simulação, por outras palavras o ficheiro de *log*. A linha seguinte está directamente relacionada com esta, e especifica que tipo de informação deve ser guardada, neste caso é dito para ser guardado todo o comportamento da rede durante a simulação.

No fim desta simulação são então gerados dois ficheiros de *log*, um ficheiro de texto com tudo o que aconteceu ao longo da simulação e outro criado especificamente para a análise visual do comportamento da rede. Este ficheiro é utilizado pelo software NAM (*Network Animator*) e possibilita uma visualização gráfica do comportamento da rede ao longo do tempo.

Existindo a oportunidade de ver visualmente a rede a figura 10 demonstra o uso do software NAM, e o cenário que foi criado, podendo ver-se a disposição dos nós e a sua identificação.



**Figura 10** – Ficheiro de Trace em modo visual

O ficheiro de texto criado pelo NS-2 é bastante mais complexo como se pode verificar pela figura 11.

```
s -t 1.000000000 -Hs 2 -Hd -2 -Ni 2 -Nx 70.00 -Ny 5.00 -Nz 0.00 -Ne -1.000000 -Nl AGT -Nw ... -Ma 0 -Md 0 -Ms 0 -Mt 0
-Is 2.0 -Id 0.0 -It cbr -Il 64 -If 0 -Ii 0 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 0

r -t 1.000000000 -Hs 2 -Hd -2 -Ni 2 -Nx 70.00 -Ny 5.00 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw ... -Ma 0 -Md 0 -Ms 0 -Mt 0
-Is 2.0 -Id 0.0 -It cbr -Il 64 -If 0 -Ii 0 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 0
```

**Figura 11** – Ficheiro de Trace modo texto

Cada uma destas linhas descreve um evento realizado pelo objecto Simulator. Cada evento possui um tipo e a sua descrição em termos de tempo de ocorrência, informações de posição (GPS), nível de *trace*, informações de pacote no nível da camada de rede (IP), entre outros.

Essas informações estão separadas por um caractere de espaço em branco e, com excepção da primeira informação, que é um caractere de indicação do tipo do evento, todas as demais informações são precedidas por *flags* que indicam seu significado.

De forma resumida é explicado de seguida o significado de alguns campos presentes na figura 11:

- **flag (s)**, indica o tipo do evento;
- **flag (-t)**, indica o tempo em que o evento ocorreu;
- **flag (-Hd)**, indica o id do nó de destino, qualquer valor menor que zero é interpretado como *broadcast*;
- **flag (-Nx, -Ny, -Nz)**, indicam as coordenadas de localização dos nós;
- **flag (-NI)** indica o nível de trace, que pode ser seguido de AGT (Agent), RTR (Router) ou MAC (referente a camada Mac);
- **flag (-Is)**, é o par que indica o endereçoIPOrigem.portaOrigem;
- **flag (-Id)**, é o par que indica o endereçoIPDestino.portaDestino;
- **flag (-I)**, esta flag diz o tamanho exacto em Bytes do pacote;

Este ficheiro fornece assim informações bastante detalhadas de toda a simulação.

### 4.3 Modelo de Simulação Para Testes

---

No modelo de simulação criado para a realização dos testes e validação dos resultados, todos os componentes falados anteriormente foram ajustados por forma a criar um ambiente de simulação o mais real possível. Apresenta-se assim a implementação base e a extensão com suporte a prioridades e correlação de eventos. Foi integrado um protocolo de *broadcast* no NS-2, o **MYPBCAST**, que foi o protocolo criado para o envio das mensagens de acidente e a sua utilização e configuração no simulador é detalhadamente explicada nos capítulos seguintes.

#### 4.3.1 Modelo Base

---

O modelo é inicialmente definido pela configuração detalhada de todos os componentes obrigatórios da simulação, como se pode verificar pela figura 12.

```
# =====
# Define options
# =====
set val(chan) Channel/WirelessChannel ;# channel type
set val(prop) Propagation/TwoRayGround ;# radio-propagation model
set val(netif) Phy/WirelessPhy ;# network interface type
set val(mac) Mac/802_11 ;# MAC type
set val(ifq) Queue/DropTail/PriQueue ;# interface queue type
set val(ll) LL ;# link layer type
set val(ant) Antenna/OmniAntenna ;# antenna model
set val(ifqlen) 50 ;# max packet in ifq
set val(nn) 450 ;# default number of mobilenodes
set val(rp) AODV ;# routing protocol
set val(sc) "./cenario-v1.tcl" ;# scenario file
set val(x) 10000.0 ;
set val(y) 10.0 ;
set val(simtime) 100.0 ; #sim time
# =====
```

**Figura 12** – Configurações da simulação

As configurações da simulação encontram-se divididas em dois grandes blocos. O primeiro está relacionado com as configurações dos *Mobile Nodes*, e o segundo relaciona-se com a configuração da topografia.

Para se simular uma situação de acidente, considerou-se uma elevada densidade de veículos. Neste caso concreto, 450, simulando assim o comportamento dos nós em auto-estrada dispostos ao longo de 10 quilómetros.

De forma a facilitar o trabalho de configuração de simulações com parâmetros diferentes, é comum definir as variáveis referentes aos componentes do *Mobile Node* no início do *script* que descreve o modelo de rede a ser simulado. Um exemplo visível na figura 12 é a especificação da topografia através de outro ficheiro (**cenario-v1.tcl**), facilitando assim a mudança entre diversos cenários. Este ficheiro contém a posição inicial dos nós no cenário específico que neste caso se tenta simular uma auto-estrada.

```
# VEICULOS QUE GERAM MENSAGENS DE ACIDENTE

$node_(1) set X_ 10000.0
$node_(1) set Y_ 15.0
$node_(1) set Z_ 0.0

$node_(95) set X_ 9995.0
$node_(95) set Y_ 15.0
$node_(95) set Z_ 0.0

$node_(96) set X_ 9992.0
$node_(96) set Y_ 0.0
$node_(96) set Z_ 0.0

# FIM DAS MENSAGENS DE ACIDENTE

$node_(2) set X_ 2960.0
$node_(2) set Y_ 15.0
$node_(2) set Z_ 0.0
$node_(3) set X_ 2920.0
$node_(3) set Y_ 0.0
$node_(3) set Z_ 0.0
```

**Figura 13** – Posição inicial dos *Mobile Node* na topografia

Na figura 13 mostra uma parte do ficheiro (**cenario-v1.tcl**), neste sub-conjunto de nós podemos ver os três veículos que geram mensagens de acidente ou seja são os veículos que colidiram (nó 1, nó 95 e nó 96), e dois nós simples que fazem parte da topologia da rede.

Após esta configuração inicial é que se criam os nós da rede. Este procedimento está descrito na figura 14.

```

for {set i 0} {$i < $val(nn) } {incr i} {
    set node_($i) [$ns_ node]
    $node_($i) random-motion 0           ;# disable random motion
}

```

**Figura 14** – Criação dos nós da rede

Os nós *Mobile Node* previamente configurados são criados com auxílio da função *node* da classe *Simulator*. A segunda linha dentro do ciclo “for” indica que estes nós não devem realizar movimentos aleatórios. No caso desta simulação não é desejável que os nós realizem movimento aleatório, mas sim que a movimentação seja especificada em conjunto com a definição do cenário da simulação. Sendo o cenário específico uma Auto-Estrada, pretende-se que os nós se movimentem em linha recta a uma velocidade constante.

A configuração da posição e movimento dos nós de uma rede sem fios está directamente relacionado com o cenário desejado para a simulação como se pode verificar pela figura 13.

Tal como foi referido anteriormente, com o intuito de auxiliar a realização de simulações com várias configurações de posição, movimento e tráfego entre nós em uma rede sem fios, a equipa do Projecto Monarch [31] desenvolveu dois programas que estão disponíveis no NS-2: o *setdest* e o *cbrgen.tcl* que permitem a geração de tráfego automaticamente.

Foram então criados três tipos diferentes de tráfego na rede, com o intuito de tentar simular uma situação o mais realista possível.

### 1. Comunicação de Voz

- Tráfego UDP do tipo: CBR
- Tamanho do pacote: 64 bytes
- Taxa de transmissão dos pacotes: 50 pacotes/segundo
- Número de comunicações de voz: 3 simultaneamente

Esta configuração foi escolhida com base no que é considerado uma comunicação por voz estável e de boa qualidade[46]. Na figura 15 apresenta-se como se configura o simulador para gerar este tipo de tráfego.

```

# INICIA PROCEDIMENTO CBR
proc cbr-traff { src dst size idle } {
    global ns_

    set udp [new Agent/UDP]
    $ns_ attach-agent $src $udp

    set cbr [new Application/Traffic/CBR]
    $cbr set packetSize_ $size
    $cbr set interval_ $idle
    $cbr attach-agent $udp

    set sink [new Agent/Null]
    $ns_ attach-agent $dst $sink

    $ns_ connect $udp $sink

    return $cbr
}

```

**Figura 15** – Função que permite gerar tráfego CBR

O tráfego propriamente dito é gerado como mostra a figura 16, pela invocação das função previamente definida pelo utilizador. Neste caso, foram geradas três comunicações de voz, onde são enviado pacotes de 64 bytes a uma taxa de transmissão de 50 pacotes/segundo. As comunicações acontecem entre veículos distintos, por exemplo no primeiro caso é entre o veículo 2 e o veículo 90. O critério de seleção destes veiculos e não de outros é para que as comunicações cubram quase toda a rede.

```

#SIMULACAO DE VOZ VARIAS CONVERSAS

# 50 pacotes por segundo
set cbr_2_90 [cbr-traff $node_(2) $node_(90) 64 0.02]
$ns_ at 3 "$cbr_2_90 start"
$ns_ at 65 "$cbr_2_90 stop"

# 50 pacotes por segundo
set cbr_20_80 [cbr-traff $node_(20) $node_(80) 64 0.02]
$ns_ at 3 "$cbr_20_80 start"
$ns_ at 65 "$cbr_20_80 stop"

# 50 pacotes por segundo
set cbr_30_75 [cbr-traff $node_(30) $node_(75) 64 0.02]
$ns_ at 3 "$cbr_30_75 start"
$ns_ at 65 "$cbr_30_75 stop"

```

**Figura 16** – Procedimento CBR

Além da comunicação de voz, também foi gerado tráfego do tipo FTP, onde se pretendeu simular a troca de ficheiro entre veículos, recorrendo neste caso ao protocolo TCP. As características do tráfego gerado foram então:

## 2. Troca de Ficheiros

- Tráfego TCP do tipo: FTP
- Tamanho do pacote: 1024 bytes
- Número de comunicações de FTP: 2 simultaneamente

A figura 17 demonstra a função que permite gerar este tráfego no simulador.

```
# INICIO PROCEDIMENTO TCP
proc ftp-traff { src dst } {
    global ns_

    set tcp [new Agent/TCP]
    $tcp set class_ 2
    $tcp set packetSize_ 1024
    set sink [new Agent/TCPSink]

    $ns_ attach-agent $src $tcp
    $ns_ attach-agent $dst $sink

    $ns_ connect $tcp $sink

    set ftp [new Application/FTP]
    $ftp attach-agent $tcp

    return $ftp
}

#END
```

**Figura 17** – Função que permite gerar tráfego FTP

O tráfego propriamente dito é gerado como mostra a figura 18, onde se pode verificar que foram geradas duas comunicações de troca de ficheiros entre veículos que duram 62 segundos. O envio dura 62 segundos para começar e terminar antes do envio e fim das mensagens de acidente. Tal como nas comunicações de voz, estes veículos foram escolhidos mais uma vez para tentar que a comunicação ocorra ao longo da rede.

```

# 2 Comunicacoes via FTP

set ftp_2_20 [ftp-traff $node_(2) $node_(20)]
$ns_ at 3 "$ftp_2_20 start"
$ns_ at 65 "$ftp_2_20 stop"

set ftp_31_50 [ftp-traff $node_(31) $node_(50)]
$ns_ at 3 "$ftp_31_50 start"
$ns_ at 65 "$ftp_31_50 stop"

#END

```

**Figura 18** – Procedimento FTP

Por ultimo, as características do trafego gerado para as mensagens de acidente são:

### 3. Mensagens de Acidente

- Tráfego: UDP
- Tamanho do pacote: 32 bytes
- Taxa de transmissão dos pacotes: 1 pacote/segundo
- Número de mensagens na rede: 1 ou 3 simultaneamente

A geração deste tipo de tráfego segue uma metodologia completamente diferente. Comparativamente às duas situações anteriores, nesta situação é usado um protocolo específico o **MYPBCAST**, que funciona em modo *broadcast* com o intuito de se fazer chegar a mensagem o mais rapidamente possível ao maior número de veículos.

Em suma, foi assim explicada de forma extensa o funcionamento do modelo base. De seguida são apresentados os modelos que permitiram resolver os problemas deixados em aberto neste modelo.

#### 4.3.2 Modelo com prioridades

---

Neste modelo com prioridades tal com dito anteriormente, o que se pretende, é garantir que as mensagens de acidente têm prioridade face a qualquer outro tipo de tráfego que passe na rede. Durante a realização desta tese o NS-2 não tinha suporte a modelos de QoS mais sofisticados, como por exemplo o DiffServ, para uma rede sem fios. Deste modo, optou-se por seguir uma abordagem suportada pelo NS-2, com algumas alterações simples. Sendo assim foram criadas duas filas distintas, uma prioritária que tratará o tráfego referente as mensagens de acidente e outra responsável por todo outro tipo tráfego que circula na rede.

Para tornar isto possível houve a necessidade de alterar algum código fonte do NS-2 de forma a ser possível dar prioridade a um protocolo específico face a qualquer outro. Como se pode



ver pela figura 19, a função *PriQueue* é responsável por dar prioridade caso receba pacotes do protocolo **MYPBCAST**. No exemplo mostra-se na linha 89 o protocolo AODV comentado, pretendendo demonstrar que existe a possibilidade de dar prioridade a mais que um protocolo de cada vez. Todos os protocolos que não aparecerem dentro do “switch” (linha 88) recebem prioridade normal.

```

80
81 void
82 PriQueue::recv(Packet *p, Handler *h)
83 {
84     struct hdr_cmn *ch = HDR_CMN(p);
85
86     if(Prefer_Routing_Protocols) {
87
88         switch(ch->ptype()) {
89             //case PT_AODV:
90             case PT_MYPBCAST:
91                 recvHighPriority(p, h);
92                 break;
93
94             default:
95                 Queue::recv(p, h);
96         }
97     }

```

**Figura 19** – Função no NS-2 que cria as duas filas de prioridades

Este tipo de fila tem que ser configurado, quando a configuração da simulação, como se pode ver na figura 20, a linha referente a configuração do "*ifq*" apresenta a especificação da fila desejada.

```

set val(ifq) Queue/DropTail/PriQueue ;# interface queue type

```

**Figura 20** – Definição do tipo de fila a usar

Um campo importante além da configuração do tipo de fila é o tamanho da própria fila, este campo também é configurado quando a configuração da simulação.

```

set val(ifqlen) 50 ;# max packet in ifq

```

**Figura 21** – Definição do tamanho das filas

Sendo tráfego referente a uma situação de emergência a opção tomada foi que enquanto existirem mensagens de acidente na fila prioritária todo o outro tráfego não é enviado. Assegura-se desta forma que o tráfego de emergência será atendido sempre primeiro que qualquer outro.

### 4.3.3 Modelo com correlação

---

Num caso de acidente que englobe vários veículos, o modelo com correlação de eventos aplica-se com o objectivo de reduzir o número de mensagens na rede e de avisar as forças de autoridade acerca do número de veículos envolvidos no acidente, de forma a prestar a melhor ajuda possível.

Para tornar estes requisitos possíveis houve a necessidade de adicionar dois novos campos (**id e seq**) ao cabeçalho do pacote no simulador NS-2. Na figura 22 é apresentado um excerto do protocolo **MYPBCAST**, onde se pode verificar nas linhas 43 e 44 a atribuição de um *id* por mensagem e um *id* por veículo respectivamente.

```
35 int MyPBCastAgent::command(int argc, const char*const* argv)
36 {
37     if (argc == 2) {
38         if (strcmp(argv[1], "send") == 0) {
39             Packet* pkt = allocpkt();
40             hdr_ip* ih = HDR_IP(pkt);
41             hdr_cmn* ch = HDR_CMN(pkt);
42             hdr_MyPBCast* hdr = hdr_MyPBCast::access(pkt);
43             hdr->id_ = id_;
44             hdr->seq_ = 1;
```

Figura 22 – Função de envio de mensagem de acidente

Com os campos criados anteriormente é possível reduzir, o número de mensagens enviadas. Assim, num acidente que envolva 3 veículos, das 9 mensagens apenas 1 será reenviada. Assim sendo, cada um dos veículos que tiver um acidente guarda todas as mensagens que receber de outros veículos a informar sobre a ocorrência do acidente. Ou seja guarda todos os IPs de veículos diferentes do seu dos quais tenham recebido outras mensagens de acidente. Percebe-se assim facilmente que cada um dos três veículos irá receber duas mensagens dos outros dois veículos acidentados. Na segunda vez que os veículos enviarem a mensagem de acidente, em vez de enviarem somente o par falado anteriormente (**id do veículo e id da mensagem**), enviarão uma mensagem com **o (id de todos os veículos associados ao acidente e com um id de mensagem especial)** que identifica que esta mensagem esta correlacionada.

Para ser possível a realização dos testes relacionados com correlação de eventos foi necessário adicionar linhas de código à classe que cria o ficheiro de trace, ou seja foi necessário novamente alterar o código fonte do NS-2 de modo a criar um ficheiro de trace no formato desejado para as simulações, como se pode ver pela figura 23. Na figura é demonstrado na linha 856 a criação dos dois campos necessários (**fid, no**) que representam respectivamente o *id* da mensagem e o *id* do veículo.

```

852 void
853 CMUTrace::format_mybroadcast(Packet *p, int offset)
854 {
855     struct hdr_MyPBCast *mh = hdr_MyPBCast::access(p);
856     sprintf(pt_->buffer() + offset, "fid: %d -no %d", mh->id_, mh->seq_);
857 }

```

**Figura 23** – Função de criação dos dois novos campos do trace

Com isto é possível saber rapidamente o *id* do veículo que enviou a mensagem, tornando possível identificar a origem da mensagem quando existem vários veículos acidentados.

Como dito anteriormente, cada veículo só retransmite uma mensagem uma vez. Assim sendo consegue-se que apenas circule uma única mensagem na rede de aviso de acidente, com todos os nós envolvidos.



---

## 5 Testes de desempenho

---

Nesta secção serão descritas as simulações e testes realizados ao protocolo, sendo que estes se dividem em testes de *End-To-End Delay* e *Packet-Delivery Ratio*. O primeiro irá permitir analisar várias métricas como por exemplo, se os condutores terão tempo para evitar outras colisões, e quanto tempo demora uma mensagem a percorrer um determinado numero de quilómetros com o intuito de avisar as forças de autoridade. Os testes relativos ao *Packet-Delivery Ratio* irão permitir analisar se todos os condutores têm conhecimento do acidente.

Estas simulações utilizaram 450 nós dispostos ao longo de 10 quilómetros, 3 padrões de tráfego e 2 cenários de acidente, um em que é só um veículo é que tem um acidente sendo gerada apenas uma mensagem do referido acidente, e outro cenário em que três veículos colidem entre si originando assim três mensagens de acidente.

Para analisar quais os melhores parâmetros da solução, foi tido em conta o impacto da diferenciação de serviços, nomeadamente entre as mensagens de emergência sobre todo o outro tráfego dito “normal”, bem como o impacto que a correlação de eventos poderá ter sobre o desempenho global da solução.

### 5.1 Resultado da execução da simulação

---

Quando uma simulação termina, o NS-2 dependendo da sua configuração inicial cria um ou dois ficheiros de *log* texto e/ou visual. Nos exemplos apresentados o NS-2 foi sempre configurado para gerar um ficheiro de texto e outro com o objectivo de dar uma perspectiva visual do ambiente de simulação.

O ficheiro de *log* de texto disponibiliza uma gama de informações sobre as camadas, os nós e o routing dos pacotes entre eles etc. Na figura 24 são apresentadas algumas linhas do ficheiro de saída do simulador. Nesta figura estão transcritas 6 linhas de *trace* que, dado seu comprimento, estão dispostas com quebras de linha. Para facilitar a identificação, cada nova linha de *trace* começa normalmente com o caractere “s” ou “r” mas também existem mais duas possibilidades que são começar com os caracteres “d” ou “f”.

Existem assim 4 tipos de eventos, que são:

- **s: envio (*send*);**
- **r: recebe (*receive*);**
- **d: descarta (*drop*);**
- **f: encaminhamento (*forward*).**

```

s -t 5.000000000 -Hs 1 -Hd -2 -Ni 1 -Nx 3000.00 -Ny 15.00 -Nz 0.00 -Ne -1.000000 -Nl AGT -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 1.0 -Id -1.0 -
It mypbcast -Il 100 -If 0 -Ii 0 -Iv 32 -fid: 1 -no 1
r -t 5.000000000 -Hs 1 -Hd -2 -Ni 1 -Nx 3000.00 -Ny 15.00 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 1.0 -Id -1.0 -
It mypbcast -Il 100 -If 0 -Ii 0 -Iv 32 -fid: 1 -no 1

s -t 5.000000000 -Hs 95 -Hd -2 -Ni 95 -Nx 2995.00 -Ny 15.00 -Nz 0.00 -Ne -1.000000 -Nl AGT -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 95.0 -Id -1.0 -
-It mypbcast -Il 100 -If 0 -Ii 1 -Iv 32 -fid: 1 -no 2
r -t 5.000000000 -Hs 95 -Hd -2 -Ni 95 -Nx 2995.00 -Ny 15.00 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 95.0 -Id -1.0 -
-It mypbcast -Il 100 -If 0 -Ii 1 -Iv 32 -fid: 1 -no 2

s -t 5.000000000 -Hs 96 -Hd -2 -Ni 96 -Nx 2992.00 -Ny 0.00 -Nz 0.00 -Ne -1.000000 -Nl AGT -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 96.0 -Id -1.0 -
It mypbcast -Il 100 -If 0 -Ii 2 -Iv 32 -fid: 1 -no 3
r -t 5.000000000 -Hs 96 -Hd -2 -Ni 96 -Nx 2992.00 -Ny 0.00 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 96.0 -Id -1.0 -
It mypbcast -Il 100 -If 0 -Ii 2 -Iv 32 -fid: 1 -no 3

```

**Figura 24** – Output do novo formato de trace (*newtrace*)

Para o estudo das métricas deste trabalho, apenas algumas informações contidas em cada linha de *trace* são relevantes. Cada uma destas linhas descreve um evento realizado pelo objecto Simulator. Cada evento possui um tipo e a sua descrição em termos de tempo de ocorrência, informações de posição (GPS), nível de *trace*, informações de pacote no nível da camada de rede (IP), entre outros.

Essas informações estão separadas por um caractere de espaço em branco e, com excepção da primeira informação, que é um caractere de indicação do tipo do evento, todas as demais informações são precedidas por *flags* que indicam seu significado.

De forma resumida é explicado de seguida o significado de alguns campos presentes na figura 24 que são necessários para se compreender os resultados:

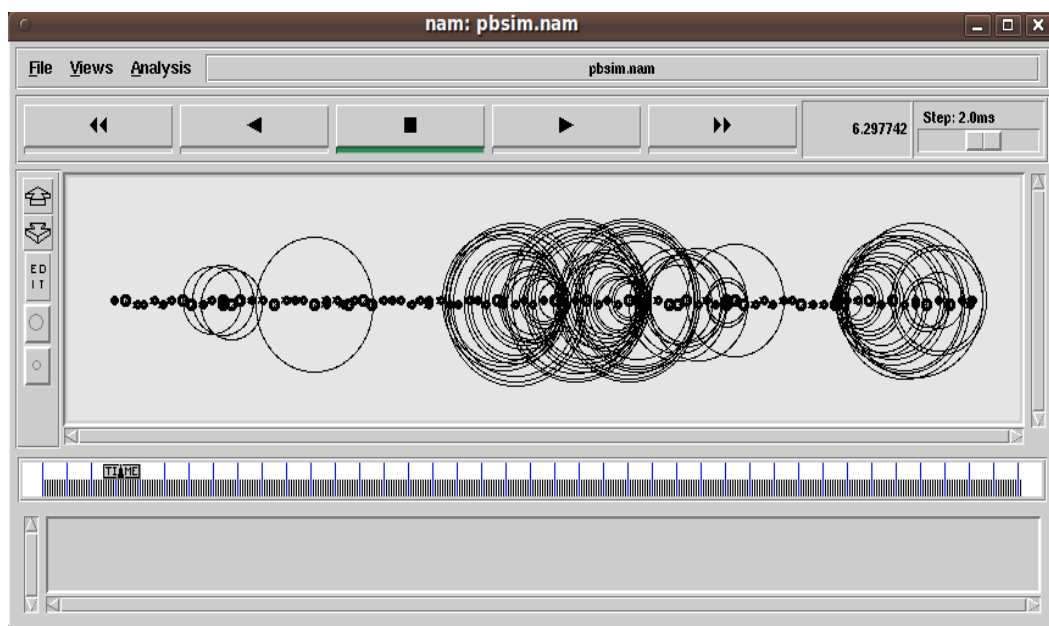
- **flag (s)**, indica o tipo do evento;
- **flag (-t)**, indica o tempo em que o evento ocorreu;
- **flag (-Hd)**, indica o id do nó de destino, qualquer valor menor que zero é interpretado como *broadcast*;
- **flag (-Nx, -Ny, -Nz)**, indicam as coordenadas de localização dos nós;
- **flag (-Ni)** indica o nível de trace, que pode ser seguido de AGT (Agent), RTR (Router) ou MAC (referente a camada Mac);
- **flag (-Is)**, é o par que indica o endereçoIPOrigem.portaOrigem;
- **flag (-Id)**, é o par que inddica o endereçoIPDestino.portaDestino;
- **flag (-Il)**, esta flag diz o tamanho exacto em Bytes do pacote;

Os dois campos seguintes foram criados para ser possível a realização dos testes relacionados com a Correlação de Eventos.

- **flag (-fid)**, esta flag permite identificar univocamente uma mensagem na rede quando associada em par com o campo seguinte;
- **flag (-no)**, este campo permite identificar univocamente um veículo na rede.

A figura 25, representa um momento visual da simulação recorrendo ao software NAM. A título de exemplo podem-se verificar três situações distintas. Do lado direito da imagem ocorreu uma situação de acidente entre três veículos, e os círculos representados nessa zona da figura, referem-se as mensagens de acidente que são enviadas em modo *broadcast*, como explicado anteriormente, cada veículo acidentado envia uma mensagem por segundo. A mensagem irá ser disseminada ao longo de todo o cenário (do lado direito da figura para o lado esquerdo) que tem o comprimento de 10km. A parte central da figura onde se nota um aumento acentuado do número de círculos representa tanto o tráfego referente às **Comunicações de Voz (CBR)** como o tráfego referente à **Troca de Ficheiros (TCP)**. Para terminar, a parte mais a esquerda da figura nota-se um número mais reduzido de círculos que representam a mensagem de acidente enviado no segundo anterior ao actual e que ainda não percorreu os 10km. Nota-se facilmente uma diferença acentuada entre o número de círculos do lado direito e do lado esquerdo, isto deve-se ao facto de o lado esquerdo da figura ser o local mais distante do acidente logo o número de veículos é muito menor comparativamente com o lado direito onde aconteceu o acidente.

É de frisar também que se pode ver no campo superior direito da simulação o instante em que esta se encontra, neste caso no instante 6.29s. Este software permite um controlo sobre os eventos bastante considerável, sendo possível aumentar e diminuir a velocidade com que passa a simulação, é possível também parar, avançar e retroceder no tempo sem qualquer dificuldade.

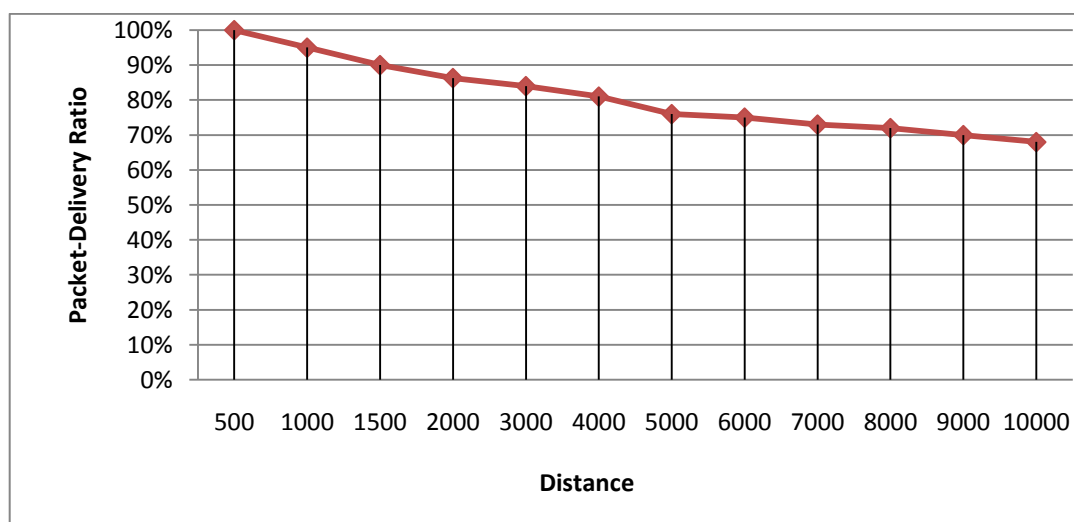


**Figura 25** – Output do NAM

## 5.2 Resultados do *Packet-Delivery Ratio*

Estes testes pretendem mostrar a eficiência de entrega dos pacotes ao longo de vários quilómetros. Apenas a entrega ou não das mensagens de acidentes são aqui testados e analisados ao pormenor. Com este intuito os resultados apresentados de seguida, mostram a percentagem da entrega de uma mensagem de acidente enviada a cada segundo durante 60 segundos ao longo de vários quilómetros.

Nesta situação específica (figura 26) o *Packet-Delivery Ratio* das mensagens de acidente deve-se somente a colisões entre pacotes, numa situação em que não existe mais tráfego na rede. O início do gráfico indica a posição onde ocorreu o acidente.

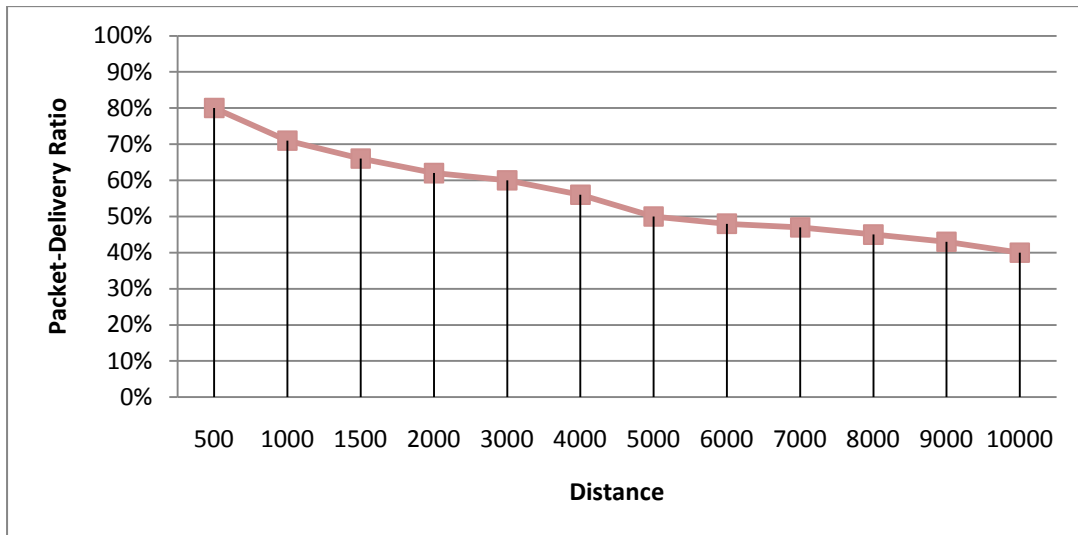


**Figura 26** – *Packet-Delivery Ratio* sem tráfego na rede

Pela análise do gráfico verifica-se que ao longo dos primeiros 500m não houve quaisquer perdas de pacotes. A partir daí dá-se uma maior perda devido afluência de veículos nas proximidades do acidente, que tende a estabilizar a partir do 5km (5000m), reduzindo assim substancialmente o número de colisões das mensagens, fazendo assim com que a mensagem se vá propagando ao longo do tempo.

O gráfico seguinte (figura 27) mostra o *Packet-Delivery Ratio* das mensagens de acidente, mas nesta situação foi adicionado à rede bastante tráfego, tentando simular uma situação real. O tráfego simulado refere-se a três comunicações simultâneas de voz, onde os intervenientes estão separados por vários quilómetros tentando que haja comunicação ao longo de toda a rede. Além das comunicações de voz, existe mais 2 comunicações TCP, recorrendo ao serviço FTP, que pretendem simular a troca de ficheiros entre veículos.



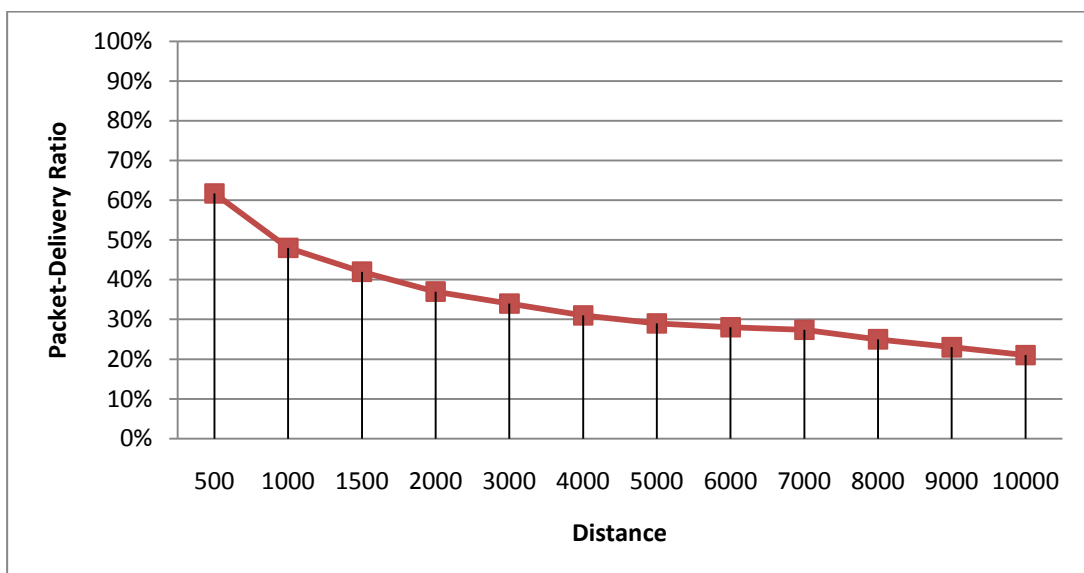


**Figura 27** – *Packet-Delivery Ratio* com tráfego na rede

O número de colisões nesta situação aumenta bastante, levando a que o *Packet-Delivery Ratio* diminua consideravelmente.

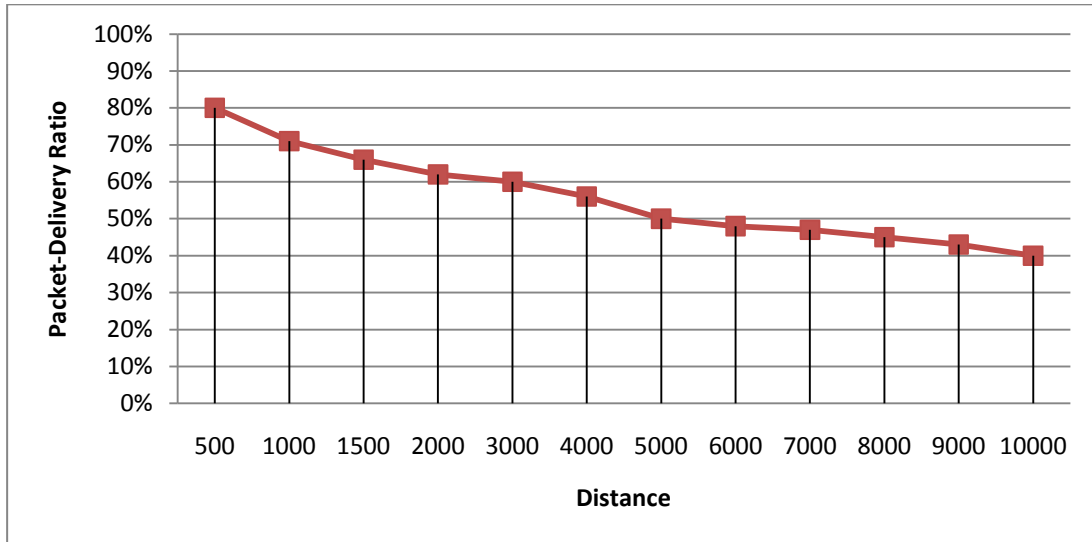
Por vezes, ou numa grande maioria das vezes, as perdas dos pacotes das mensagens de acidente, não acontecem só em situações de colisão, também são perdidas por serem descartadas do buffer que se encontra cheio, ou por exceder o máximo do TTL.

Na figura 28, pode-se verificar o *Packet-Delivery Ratio* quando existem problemas de colisões e de buffer cheio, que levam as mensagens a serem descartadas.



**Figura 28** – *Packet-Delivery Ratio* com situações de buffer cheio

Uma das soluções apresentadas, figura 29, é dar a estas mensagens de acidente prioridade sobre qualquer outro tipo de mensagens, com isto conseguiu-se subir consideravelmente o *Packet-Delivery Ratio*, face ao seu envio sem prioridades.



**Figura 29** – *Packet-Delivery Ratio* recorrendo a prioridades

Os testes apresentados, referem-se a uma situação simples de acidente onde um veículo tem um acidente sózinho e apenas uma mensagem de acidente é gerada por segundo.

Numa situação real, muitas vezes existem colisões entre vários veículos levando a ser criada uma mensagem individual de acidente por veículo. Dependendo do número de veículos que colidem leva a que o número de mensagens na rede cresça de forma exponencial. Numa situação de três veículos, por exemplo, iram ser criadas três mensagens uma por veículo, e cada um deles irá enviar a sua mensagem de acidente e retransmitir a dos veículos que também colidiram consigo sendo desta forma enviadas ao todo nove mensagens de acidente, com isto percebe-se que o tráfego na rede será muito grande.

Desta forma, os testes seguintes passam por apresentar o *Packet-Delivery Ratio* ao longo de 10km sem e com correlação sobre as mensagens de acidente. A razão pela qual foi usada uma tabela na apresentação de resultados e não um gráfico é porque facilita a visualização das percentagens de entrega de cada mensagem ao longo do cenário.

**Tabela 4 – PACKET DELIVERY RATIO SEM CORRELAÇÃO**

<b>Distancia</b>	<b>Entrega MSG A</b>	<b>Entrega MSG B</b>	<b>Entrega MSG C</b>
<b>500</b>	98%	98%	96%
<b>1000</b>	96%	96%	96%
<b>1500</b>	96%	95%	95%
<b>2000</b>	96%	96%	96%
<b>3000</b>	90%	90%	88%
<b>4000</b>	87%	87%	85%
<b>5000</b>	85%	86%	82%
<b>6000</b>	81%	82%	79%
<b>7000</b>	82%	82%	78%
<b>8000</b>	79%	79%	76%
<b>9000</b>	77%	78%	76%
<b>10000</b>	75%	76%	74%

Ao contrário do que era esperado, pela análise da tabela 4, o *Packet-Delivery Ratio* melhorou bastante face ao envio de uma única mensagem de acidente, isto deve-se ao facto de existir muito mais tráfego prioritário não dando possibilidade ao outro tipo de tráfego ser enviado, resultando assim numa redução do número de colisão das mensagens de acidente.

Inicialmente parece uma boa solução não se usar qualquer correlação destas mensagens devido a melhoria dos valores do packet delivery ratio, mas surge o problema de existir a possibilidade de por exemplo as forças da autoridade que precisam de saber o número exacto de veículos acidentados só receber mensagem de dois veículos e nunca receber de um terceiro. Para resolver este problema uma solução é então recorrer ao uso da correlação de eventos, onde em vez de enviar uma mensagem por veículo acidentado, envia uma mensagem com todos os veículos envolvidos no acidente garantindo assim que as forças de autoridade sabem ao certo o número de veículos acidentados podendo prestar assim a melhor ajuda possível.

A tabela 5 mostra assim o *Packet-Delivery Ratio* quando é usada esta metodologia de correlação de eventos, garantindo assim um número de mensagens na rede muito mais reduzido, de nove para apenas uma mensagem que informa do acidente e todos os veículos envolvidos.

**Tabela 5 – PACKET DELIVERY RATIO COM CORRELAÇÃO**

<b>Distancia</b>	<b>Entrega MSG A+B+C</b>
<b>500</b>	76%
<b>1000</b>	64%
<b>1500</b>	60%
<b>2000</b>	55%
<b>3000</b>	52%
<b>4000</b>	47%
<b>5000</b>	43%
<b>6000</b>	40%
<b>7000</b>	39%
<b>8000</b>	38%
<b>9000</b>	37%
<b>10000</b>	35%

Em suma dos resultados obtidos a nível do *Packet-Delivery Ratio* das mensagens de acidente, uma ou várias, usando correlação e prioridades apresentam resultados satisfatórios. Recorrendo à prioridade, uma fila específica é criada para as mensagens de acidentes, fazendo com que estas nunca sejam descartadas por a fila estar cheia. Além disso, esta fila é prioritária face a qualquer outra garantindo assim que as mensagens de emergência serão sempre enviadas. Recorrendo a correlação de eventos, foi possível garantir que as mensagens eram entregues a curtas e longas distâncias com toda a informação necessária, ou seja todos os veículos envolvidos no acidente, considerando assim este protocolo escalável a um número indeterminado de mensagens de acidente.

### 5.3 Resultados do End-To-End Delay

Os resultados deste teste têm um valor muito importante para esta tese. É com base neles que conseguimos analisar se os condutores têm realmente a possibilidade de ser avisados a tempo de evitar um choque em cadeia, e ao mesmo tempo ter uma ideia de quanto demoraria a serem avisadas as forças de autoridade, onde por exemplo um receptor se encontra a vários quilómetros do acidente.

O primeiro teste realizado, passou pela análise do tempo que uma mensagem de acidente demora a percorrer dez quilómetros caso não houvessem colisões, ou seja num ambiente perfeito. Três situações são apresentadas como mostra a tabela 6. A primeira refere-se a uma mensagem de acidente quando não existe tráfego na rede (ruído). As duas seguintes, apresentam os tempos de entrega de uma mensagem de acidente quando existe tráfego na rede (ruído) com e sem prioridade.

**Tabela 6** - TABELA DE TEMPOS MEDIOS DE ENTREGA

Situação / Distância (m)	1000	3000	5000	7000	9000	10000
<b>SEM TRAFEGO (1 Msg)</b>	0.14s	0.8s	1.18s	1.93s	2.54s	2.67s
<b>COM TRAFEGO (1 Msg) - Sem Prioridade</b>	0.35s	1.1s	1.57s	2.31s	2.84s	3.1s
<b>COM TRAFEGO (1 Msg) - Com Prioridade</b>	0.34s	0.97s	1.45s	2.20s	2.73s	2.87s

Analisando a tabela 6, nota-se alguns resultados interessantes. Inicialmente uma comparação entre a existência ou não de tráfego na rede pode-se concluir que as diferenças de tempo são de algum modo significativas, mas sendo imperceptíveis para o condutor. Por outro lado o uso da prioridade, não apresenta aparentemente vantagens significativas na melhoria do tempo de entrega das mensagens.

O teste seguinte, faz-se uma análise dos tempos reais que as mensagens de acidente demoraram a percorrer dez quilómetros na realidade, ou seja sabendo que é enviada uma mensagem de aviso de acidente a cada segundo, saber qual destas mensagens realmente chegou ao destino.

**Tabela 7** - TEMPO REAL DE AVISO DE ACIDENTE ATÉ AO QUILOMETRO X

Situação / Distância (m)	1000	3000	5000	7000	9000	10000
<b>SEM TRAFEGO (1 Msg)</b>	0.14s	0.8s	1.18s	1.93	2.54s	2.67
<b>COM TRAFEGO (1 Msg) - Sem Prioridade</b>	3.4s	3.9s	5.5s	6.4s	6.8s	7.3s
<b>COM TRAFEGO (1 Msg) - Com Prioridade</b>	3.3s	3.5s	5.1s	6.0s	6.3s	7.0s

Pela análise da tabela 7 pode-se verificar que por exemplo no caso em que é enviada uma mensagem de acidente com tráfego na rede e sem prioridade (linha 2), comparativamente com a mesma linha da tabela 6, a mensagem demora mais três segundos, isto deve-se ao facto de só a terceira mensagem enviada pelo veículo acidentado é que realmente chegou aos mil metros, e que as duas mensagens anteriores enviadas no instante em que o veículo bateu, e no instante seguinte se perderam, possivelmente devido a colisões. Comparativamente os valores relacionados com o uso ou não de prioridades verifica-se que o seu uso não apresenta vantagens significativas.

Tal como feito nos testes sobre o *Packet-Delivery Ratio*, os testes de *End-To-End Delay* também têm que tomar em consideração uma situação, onde muitas vezes existem colisões entre vários veículos levando a ser criada uma mensagem individual de acidente por veículo. Tal como foi dito anteriormente dependendo do número de veículos que colidem leva a que o número de mensagens na rede cresça de forma exponencial.

O teste seguinte baseou-se no tempo que as três mensagens de acidente demoram a chegar realmente ao destino. Quando não é usada correlação é medido o tempo que uma mensagem enviada por qualquer dos veículos demora a chegar ao destino. Quando se usa correlação é avaliado o tempo de entrega da mensagem com a informação total do acidente.

**Tabela 8** - TEMPO REAL DE AVISO DE TODOS OS ACIDENTADOS ATÉ AO QUILOMETRO X

<b>Situação / Distância (m)</b>	<b>1000</b>	<b>3000</b>	<b>5000</b>	<b>7000</b>	<b>9000</b>	<b>10000</b>
<b>C/ Tráfego (3 Msg) – PRIO/SEM CORR</b>	0.61s	1.8s	2.21s	2.55s	3.1s	3.5s
<b>C/ Tráfego (1 Msg) – PRIO/COM CORRE</b>	4.45s	4.9s	6.2s	6.9s	7.4s	8.1s

Analisando a tabela 8 podem-se tirar várias conclusões. Inicialmente com o envio de três mensagens de acidente é medido simplesmente o tempo que qualquer umas das mensagens ( mensagem do veiculo 1, veiculo 2 ou veiculo 3) chega à distância desejada. Existindo um maior numero de mensagens a circular na rede a probabilidade de qualquer uma delas chegar ao destino é muito maior. A segunda situação ao inves de serem enviadas três mensagens uma por veiculo é simplesmente enviada uma mensagem com a informação de acidente dos três veiculos. As diferenças de tempo existem porque como a correlação só é feita no envio da segunda mensagem de acidente, e nunca na primeira, o tempo seria logo um segundo mais lento, mas neste caso, obteve-se um valor de 4.45s para os primeiros 1000metros porque a mensagem que realmente chegou não foi logo a segunda devido a colisões, mas sim uma das seguintes.

---

## 6 Conclusões e Trabalho Futuro

---

Como conclusão, verifica-se que os objectivos propostos foram atingidos, recorrendo à arquitectura proposta e testada que nos permitiu tirar elações sobre paramentos importantes neste tipo de redes.

Em suma dos resultados de todos os testes é feita de seguida uma análise geral dos resultados obtidos. Com o intuito de se garantir todas as métricas definidas como por exemplo, aviso atempado dos condutores para evitar situações de acidente em cadeia, aviso das autoridades o mais rapidamente possível de uma situação de acidente, e por ultimo o aviso de condutores a longa distância para que possam seguir outro caminho evitando assim filas de transito, foi perceptível que além do uso de prioridades das mensagens existe a necessidade de se recorrer ao uso da correlação de eventos reduzindo de uma forma muito significativa o número de mensagens que circulam assim na rede, e garantindo que a informação do acidente quando é fornecida a outro veiculo vem completa.

Os testes referentes ao *Packet-Delivery Ratio* demonstram que na situação do envio de uma única mensagem, o uso da prioridade traz resultados bastante significativos comparativamente ao envio da mensagem sem o uso da prioridade.

A correlação de eventos, não apresenta vantagens a nível de *Packet-Delivery Ratio*, verificando-se mesmo pelos valores apresentados que o *Packet-Delivery Ratio* é melhor sem a correlação. Esta métrica só tem haver com a entrega de qualquer mensagem, desta forma não se garante que todas as mensagens de todos os veículos acidentados sejam entregues a todos os outros veículos, no caso da correlação a mensagem que se encontra em transmissão já tem toda a informação, desta forma todos os condutores que receberem a mensagem e principalmente as forças de autoridade são informados de todos os veículos envolvidos no acidente, podendo garantir assim um pronto socorro melhor e mais eficiente as vitimas.

Os dois primeiros testes de End-To-End delay mostram que duas das métricas definidas, são amplamente cumpridas. Estas métricas referem-se aos condutores que se encontram longe do local do acidente têm realmente a possibilidade de ser avisados a tempo de desviar e seguir outro caminho com o intuito de evitar filas de trânsito, e ao mesmo tempo oferecem uma visão de quanto demoraria a serem avisadas as forças de autoridade, onde por exemplo um receptor se encontra a vários quilómetros do acidente.

Analisando os valores obtidos os resultados do *End-To-End delay* só para os primeiros 1000metros, nota-se que as mensagens são entregas em tempos quando imperceptíveis (0.14s) o que leva os condutores que se encontrem mais próximos do acidente recebam as mensagem quase instantaneamente no momento do acidente, dando assim a possibilidade aos condutores de travarem em segurança.

Como trabalho futuro, várias coisas podem ser estudadas e desenvolvidas no âmbito de tornar todo este trabalho melhor, assim como apresentar melhorias ao algoritmo desenvolvido.

Uma área de estudo seria estudar a possibilidade dos veículos, que se encontrassem a poucos metros, travarem de forma automática ao receberem informação de acidente por forma reduzir os tempos referentes à reacção do condutor.

Estudo de soluções que permitam que as mensagens de emergência sejam enviadas mesmo sem a existência de veículos nas proximidades para retransmitir as mensagens. Recorrendo por exemplo a uma rede alternativa para completar o processo de transferência de informação crítica, quando a comunicação pela VANET não for viável, que seja acessível em qualquer lugar, como é o caso da rede GSM.

Interessante será também a realização do Stress testing da solução proposta nesta tese por forma a se identificar os limites reais do número de mensagens possíveis na rede.



1. Whitaker, R. & Hurley, S. (), 'Evolution of planning for wireless communication systems'Proc. of HICSS', Citeseer.
2. Chakeres, I.; Danilov, C.; Henderson, T. & Macker, J. (2007), 'Connecting MANET Multicast'IEEE Military Communications Conference, 2007. MILCOM 2007', 1--7.
3. Bernsen, J. & Manivannan, D. (2009), 'Unicast routing protocols for vehicular ad hoc networks: A critical comparison and classification', *Pervasive and Mobile Computing* **5**(1), 1--18.
4. Naumov, V. & Gross, T. (2007), 'Connectivity-aware routing (CAR) in vehicular ad hoc networks'Proceedings IEEE International Conference on Computer Communications, Anchorage, AK, USA', 1919--1927.
5. Zhao, J. & Cao, G. (2008), 'VADD: Vehicle-assisted data delivery in vehicular ad hoc networks', *IEEE Transactions on Vehicular Technology* **57**(3), 1910--1922.
6. Burgess, J.; Gallagher, B.; Jensen, D. & Levine, B. (2006), 'Maxprop: Routing for vehicle-based disruption-tolerant networks'Proc. IEEE Infocom', Citeseer.
7. Zhu, J. & Roy, S. (2003), 'MAC for dedicated short range communications in intelligent transport system', *IEEE Communications Magazine* **41**(12), 60--67.
8. Lin, X.; Lu, R.; Zhang, C.; Zhu, H.; Ho, P. & Shen, X. (2008), 'Security in vehicular ad hoc networks', *IEEE Communications Magazine* **46**(4), 88.
9. Acosta-Marum, G. (2009), 'WAVE: A Tutorial', *IEEE Communications Magazine*, 127.
10. Jiang, D. & Delgrossi, L. (2008), 'IEEE 802.11 p: Towards an international standard for wireless access in vehicular environments'Proc. of IEEE Vehicular Technology Conference (VTC) Spring'.
11. Crow, B.; Widjaja, I.; Kim, L. & Sakai, P. (1997), 'IEEE 802.11 wireless local area networks', *IEEE Communications magazine* **35**(9), 116--126.
12. Namboodiri, V. & Gao, L. (2007), 'Prediction-based routing for vehicular ad hoc networks', *IEEE Transactions on Vehicular Technology* **56**(4), 2332.
13. F, J. (2002), 'Round robin classification', *The Journal of Machine Learning Research* **2**, 747.
14. Stiliadis, D. & Varma, A. (1998), 'Efficient fair queueing algorithms for packet-switched networks', *IEEE/ACM Transactions on Networking (TON)* **6**(2), 185.

15. Zhang, H. & Keshav, S. (1991), 'Comparison of rate-based service disciplines'Proceedings of the conference on Communications architecture & protocols', ACM, 121.
16. Zhang, H. & Knightly, E. (1996), 'RCSP and stop-and-go: A comparison of two non-work-conserving disciplines for supporting multimedia communication', *Multimedia Systems* 4(6), 346--356.
17. Steinder, M. & Sethi, A. (2004), 'A survey of fault localization techniques in computer networks', *Science of Computer Programming* 53(2), 165--194.
18. Perkins, C. & Royer, E. (1999), 'Ad-hoc on-demand distance vector routing'proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications', Citeseer, 90--100.
19. Johnson, D. & Maltz, D. (1996), 'Dynamic source routing in ad hoc wireless networks', *KLUWER INTERNATIONAL SERIES IN ENGINEERING AND COMPUTER SCIENCE*, 153--179.
20. Rao, S.; Pai, M.; Boussedjra, M. & Mouzna, J. (2008), 'GPSR-L: Greedy perimeter stateless routing with lifetime for VANETS'ITS Telecommunications, 2008. ITST 2008. 8th International Conference on', 299--304.
21. Liu, L.; Wang, Z. & Jehng, W. (2008), 'A geographic source routing protocol for traffic sensing in urban environment'IEEE International Conference on Automation Science and Engineering, 2008. CASE 2008', 347--352.
22. Eklund, P.; Kirkby, S. & Pollitt, S. (1996), 'A dynamic multi-source Dijkstra's algorithm for vehicle routing"Australian and New Zealand Conference on Intelligent Information Systems', Citeseer, 329--333.
23. Liu, G.; Lee, B.; Seet, B.; Foh, C.; Wong, K. & Lee, K. (2004), 'A routing strategy for metropolis vehicular communications', *Lecture notes in computer science*, 134--143.
24. Koubek, M.; Rea, S. & Pesch, D. (), 'Effective Emergency Messaging in WAVE based VANETS'.
25. Ahmed, S. & Kanere, S. (2006), 'SKVR: scalable knowledge-based routing architecture for public transport networks"Proceedings of the 3rd international workshop on Vehicular ad hoc networks', ACM, 93.
26. Korkmaz, G.; Ekici, E.; Ozguner, F. & Ozguner, U. (2004), 'Urban multi-hop broadcast protocol for inter-vehicle communication systems"Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks', ACM New York, NY, USA, 76--85.
27. Benslimane, A. (2004), 'Optimized dissemination of alarm messages in vehicular ad-hoc networks (vanet)', *LECTURE NOTES IN COMPUTER SCIENCE.*, 655--666.

28. Benslimane, A. 'Rapid Generation of Realistic Simulation for VANET', <http://www.csie.ncku.edu.tw/~klan/move/index.htm>.
29. IEEE 802.11 (2008) The Working Group Setting the Standards for Wireless LANs, <http://ieee802.org/11/>.
30. Barken, L., How secure is your wireless network? Safeguarding our Wi-Fi LAN, Prentice Hall PTR, Upper Saddle River, New Jersey, 2004, ISBN: 0131402064.
31. The Rice University Monarch Project (2008): Mobile Networking Architectures, <http://www.monarch.cs.rice.edu/>
32. The Network Simulator - NS-2 (2008), <http://www.isi.edu/nsnam/ns/>
33. Blum, A. & Rivest, R. (1992), 'Training a 3-node neural network is NP-complete', *Neural Networks* **5**(1), 117--127.
34. Duda, R.; Hart, P. & Stork, D. (2001), *Pattern classification*, Citeseer.
35. Huang, Y. & Garcia-Molina, H. (2003), 'Publish/subscribe tree construction in wireless ad-hoc networks' *Mobile Data Management*, Springer, 122--140.
36. GREIS, M., Marc Greis' Tutorial for the UCB/LBNL/VINT Network Simulator "ns", <http://www.isi.edu/nsnam/ns/tutorial/>.
37. Chen, Q.; Jiang, D.; Taliwal, V. & Delgrossi, L. (2006), 'Ieee 802.11 based vehicular communication simulation design for ns-2' *Proceedings of the 3rd international workshop on Vehicular ad hoc networks*, ACM, 56.
38. Consiglio, W.; Driscoll, P.; Witte, M. & Berg, W. (2003), 'Effect of cellular telephone conversations and other potential interference on reaction time in a braking response', *Accident Analysis & Prevention* **35**(4), 495--500.
39. Lerner, N. (1993), 'Brake perception-reaction times of older and younger drivers'(2) *Human Factors and Ergonomics Society Annual Meeting Proceedings*, Human Factors and Ergonomics Society, 206--210.
40. Schweitzer, N.; Apter, Y.; Ben-David, G.; Liebermann, D. & Parush, A. (1995), 'A field study on braking responses during driving. II. Minimum driver braking times', *Ergonomics* **38**(9), 1903--1910.
41. Summala, H. (2000), 'Brake reaction times and driver behavior analysis', *Transportation Human Factors* **2**(3), 217--226.
42. Warshawsky-Livne, L. & Shinar, D. (2002), 'Effects of uncertainty, transmission type, driver age and gender on brake reaction and movement time', *Journal of Safety Research* **33**(1), 117--128.

43. Hoogendoorn, S. & Bovy, P. (2001), 'State-of-the-art of vehicular traffic flow modeling', *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering* **215**(4), 283--303.
44. Kurkowski, S., Camp, T., Colagrosso, M.: Manet simulation studies: the incredibles. In: SIGMOBILE Mob. Comput. Commun. Rev. Volume 9., New York, NY, USA, ACM (2005) 50–61
45. Cavin, D., Sasson, Y., Schiper, A.: On the accuracy of manet simulators. In: POMC '02: Proceedings of the second ACM international workshop on Principles of mobile computing, New York, NY, USA, ACM (2002) 38–43
46. Cole, R.G. and Rosenbluth, JH: Voice over IP performance monitoring. ACM SIGCOMM Computer Communication Review. ACM (2001) 24