Uma Abordagem Adaptativa para Detecção de Falhas em Redes Veiculares *Ad hoc*[*]

Eduardo Cambruzzi[1], Jean-Marie Farines[2], Raimundo José de Araújo Macêdo[3], Werner Kraus Jr.[2]

[1]Instituto Federal de Educação, Ciência e Tecnologia da Bahia - IFBA

[2]Universidade Federal de Santa Catarina - UFSC

[3]Universidade Federal da Bahia - UFBA

Abstract. *The capacity to detect component failures is an important characteristic for fault-tolerant distributed systems. This article presents a failure detector suitable for Vehicular Ad hoc NETworks (VANETs) environments. The detector is composed of two parts: a detector for the communications link and the failure detector proper. While the link detector verifies the validity of a link between two process, the failure detector adapts itself to the mobility of the vehicles and variations in the communication load in the network. Simulation results show that the detector presents a low number of false-suspicious and a reduced detection time.*

Resumo. *A capacidade de detecção de falhas é uma característica importante para aplicações distribuídas tolerante a falhas. Este artigo apresenta um detector de falhas adaptado ao ambiente das redes veiculares ad hoc (VANETs). O detector proposto possui duas partes: um mecanismo detector de conectividade e o detector de falhas propriamente dito. Enquanto o detector de conectividade verifica a validade dos enlaces de comunicação, o detector de falhas adapta-se à mobilidade dos veículos e variações na carga de comunicação da rede. Os resultados das simulações mostram que o detector apresenta um baixo número de falso-suspeitos e reduzido tempo de detecção.*

1. Introdução

A evolução da eletrônica embarcada e das tecnologias de comunicação sem fio permitiu o desenvolvimento das redes móveis ad hoc (MANETs - *Mobile Ad hoc Networks*) e das redes veiculares ad hoc (VANETs - *Vehicular Ad hoc Networks)*. As VANETs são redes dinâmicas, não estruturadas, auto-organizáveis, com características assíncronas e distribuídas, nas quais os nós (veículos) se movem em alta velocidade. Nessas redes, os nós se comunicam diretamente entre si ou através de outros nós intermediários que atuam como roteadores.

O principal objetivo das redes veiculares ad hoc é estabelecer um meio para comunicação interveicular que permita a troca de mensagens entre veículos (V2V - *Vehicle to Vehicle Communication*) e destes com uma infra-estrutura de beira de pista (V2R - *Vehicle to RoadSide Communication*), de modo, por exemplo, a permitir seu uso em Sistemas Inteligentes de Transporte (SIT) [Taha and Hasan 2007].

[*]Agradecemos a CAPES e ao CNPq pelo apoio financeiro através das bolsas de pesquisa.

Os sistemas inteligentes de transporte originam-se do uso intensivo de tecnologias de comunicação e informação em aplicações de transporte como, por exemplo, veículos automáticos, prevenção de colisão, sistemas de cooperação veículo-via, controle de tráfego, gerenciamento serviços de emergência, entre outros. Várias destas aplicações dependem das informações sobre o comportamento de outros veículos no seu entorno ou seja, em sua vizinhança, para então, definir quais serão as ações a serem tomadas. Por exemplo, um veículo que deseja entrar em uma rodovia através de uma rampa de acesso, pode reduzir ou acelerar, dependendo das informações que recebe sobre o comportamento de outros veículos em sua vizinhança.

Recentemente, várias aplicações para sistemas inteligentes de transporte que utilizam comunicação em VANETs foram propostas [Harri et al. 2007, Ibrahim and Weigle 2008, Cambruzzi et al. 2009]. Na proposta apresentada por Cambruzzi et al. [Cambruzzi et al. 2009], utiliza-se comunicação em VANETs para coletar em tempo real os dados individuais de veículos que trafegam em vias urbanas. Nesta proposta os veículos se organizam em agrupamentos[1] *(clusters)* e, a partir destes agrupamentos, os dados dos veículos são enviados periodicamente para pontos de coleta instalados junto aos semáforos. O algoritmo proposto utiliza as características de mobilidade dos veículos e seu comportamento no ambiente de tráfego, para manter estes veículos juntos em um mesmo agrupamento pelo maior tempo possível. Com isso, há uma redução do número de filiações, refiliações ou formação de novos agrupamentos, reduzindo também a carga de comunicação gerada por estas operações.

No entanto, esta e outras propostas para aplicações em VANETs pressupõem um ambiente sem perda de mensagens e no qual os nós (veículos) são sempre saudáveis, isto é, os nós não apresentam falhas. A possibilidade de nós falharem e a dificuldade em diferenciar uma falha de um atraso na comunicação entre os processos podem ocasionar diversos problemas para aplicações em VANETs, seja para formação e manutenção de agrupamentos, seja para coletar ou difundir dados na rede. Entre os principais problemas estão, por exemplo, contar com serviços de nós que já falharam há algum tempo, o que poderia aumentar o atraso de comunicação e inviabilizar uma coleta de dados dentro de uma determinada restrição temporal.

Apesar da importância de sistemas de detecção de falhas para a construção de aplicações distribuídas como, por exemplo, na comunicação de grupo, replicação de serviços e em consenso distribuído, a implementação de detectores de falhas adequados às características de comunicação e mobilidade das redes veiculares ad hoc tem sido negligenciada.

Neste artigo propõe-se um detector de falhas escalável, adequado às características de comunicação e mobilidade das redes veiculares ad hoc. O objetivo principal deste detector é permitir a formação e manutenção de agrupamentos mais estáveis, de modo que os dados individuais dos veículos possam ser coletados dentro das restrições temporais impostas pelas aplicações SIT.

A precisão e velocidade do detector de falhas proposto foram avaliadas através de

[1]Neste artigo utilizam-se as seguintes traduções para termos originalmente em inglês: *cluster* para agrupamento; *heartbeat* para sinalização; *fault-error-failure* para falta-erro-falha; *crash-fault* para falha por parada; *timeout* para limite de tempo; *overhead* para sobrecarga.

simulações, utilizando o simulador de eventos discretos OMNET++ *(Objective Modular Network Testbed in C++)* [Varga et al. 2001]. Sua eficiência foi medida a partir da análise do número de falsas suspeitas, tempo médio de detecção de falhas por parada *(crash-fault)* e tempo médio para recuperação de falsas suspeitas.

Este artigo está organizado da seguinte forma: A Seção 2 descreve de modo resumido o ambiente de comunicação em VANETs. Na Seção 3 apresentam-se os principais requisitos para a implementação de detectores de falhas em VANETs e algumas propostas de detectores de falhas em redes móveis. Nas Seções 4 e 5 apresentam-se respectivamente o modelo do sistema e o detector de falhas proposto neste artigo. Os resultados das simulações e as conclusões são apresentados nas Seções 6 e 7.

2. Comunicação em VANETs

As redes veiculares ad hoc podem ser organizadas de modo plano ou hierárquico. Na organização plana, os nós são vistos todos no mesmo nível e nenhum deles ocupa uma função especial dentro da rede. Além disso, a comunicação acontece através da difusão de dados, utilizando, por exemplo, mecanismos de inundação (flooding). Apesar da simplicidade das redes planas, sua utilização em VANETs implica na perda de escalabilidade, devido principalmente ao grande número de transmissões e retransmissões geradas durante a comunicação [Xu and Gerla 2002].

Por outro lado, na organização hierárquica os nós da rede são separados em agrupamentos (clusters). Nestes agrupamentos, os nós podem assumir diferentes funções, por exemplo: um nó assume a função de líder se responsabilizando pelo controle da comunicação dentro do agrupamento. Outros nós podem exercer a função de nós de borda *(gateways)*, construindo canais de comunicação entre os agrupamentos. A utilização de comunicação hierárquica em VANETs melhora a escalabilidade, pois os problemas de comunicação são tratados dentro dos agrupamentos e não são propagados pela rede. Além disso, otimiza o uso do espectro, oferecendo a possibilidade de utilização de múltiplos canais, por exemplo, um canal para comunicação intra-agrupamento e outro para inter-agrupamento [Ohta et al. 2003].

Nos últimos anos várias aplicações para sistemas inteligentes de transporte tem utilizado comunicação em VANETs como parte de suas soluções e serviços [O'Gorman and Eng 2002, Gunter et al. 2007, Cambruzzi et al. 2009]. O'Gorman e Eng, [O'Gorman and Eng 2002] apresentam uma proposta de aplicação que utiliza comunicação plana para coordenar a passagem de veículos em um cruzamento não controlado, ou seja, sem controlador semafórico instalado. Nesta proposta, os veículos que se aproximam de um cruzamento negociam sua passagem por ele através da troca de mensagens e esta negociação ocorre antes mesmo que os motoristas tenham contato visual entre si. Uma vez que cada veículo tenha identificado seus vizinhos, que como ele pretendem passar pelo cruzamento, é estabelecida uma ordem de acesso a este cruzamento, de modo, que estes veículos percam o menor tempo possível nesta passagem. O objetivo principal dessa proposta é otimizar o fluxo de tráfego, evitando a formação de filas junto aos cruzamentos.

Gunter et al. [Gunter et al. 2007] propõe um protocolo para a camada de acesso ao meio (MAC) que utiliza comunicação hierárquica em VANETs. Nessa proposta os veículos trocam mensagens de HELLO entre si, de modo, que após algum tempo são

estabelecidos os agrupamentos. Um agrupamento é formado por um nó líder e por vários nós membros. O nó líder é responsável pela coordenação da comunicação de todos os membros do agrupamento. Esta coordenação é feita determinando-se uma fatia de tempo, semelhante ao modo TDMA *(Time Division Multiple Access)*, na qual, cada membro do agrupamento tem acesso ao meio físico, podendo assim estabelecer a comunicação com os demais nós da rede. O controle do acesso ao meio proposto pelos autores, reduz o número de colisões e melhora na escalabilidade da rede.

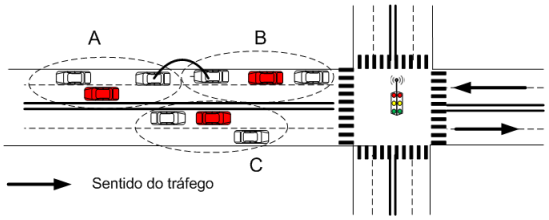


Figura 1. Agrupamentos Sobre as Vias [Cambruzzi et al. 2009]

Cambruzzi et al. [Cambruzzi et al. 2009] também apresenta uma proposta de organização hierárquica da rede, como mostra a Figura 1. Nesta proposta, os veículos que trafegam no mesmo sentido sobre as vias formam agrupamentos, nos quais, o líder é responsável pela coleta dos dados de cada um dos membros deste agrupamento. Posteriormente estes dados são enviados para os semáforos instalados nos cruzamentos. Estes dados podem ser utilizados, por exemplo, para decidir em tempo real as ações de controle que um semáforo deve executar. A formação dos agrupamentos ocorre logo após os nós terem identificado suas respectivas vizinhanças, ou seja, os veículos que estão dentro de sua área de cobertura de comunicação. Esta identificação é feita a partir das mensagens de sinalização enviadas periodicamente pelos veículos. Uma vez que um nó tenha informações sobre seus vizinhos como, por exemplo, identificador, posição, sentido e velocidade, ele calcula para si um peso. O nó com menor peso em uma vizinhança é escolhido líder (veículos em negrito na Figura 1) e os demais veículos, cujo sentido seja o mesmo deste líder, se filiam a ele formando o agrupamento veículos, (representados pelos círculos pontilhados).

Apesar das propostas descritas anteriormente apresentarem resultados que indicam sua viabilidade de aplicação em VANETs, todas pressupõem um ambiente de comunicação estável, sem atrasos ou com atrasos previsíveis. Além disso, os nós da rede não apresentam falhas, como, por exemplo, falha por parada *(crash-fault)*, sendo que, tais hipóteses são no mínimo artificiais e não podem ser adotadas em um ambiente real de comunicação em VANETs.

Uma vez que as VANETs são um campo recente de pequisa e poucos são os trabalhos que envolvem detecção de falhas nestas redes, a próxima seção apresenta algumas das propostas mais recentes e conhecidas para implementação de detectores de falhas em redes ad hoc móveis.

3. Detectores de Falhas em Redes Ad hoc Móveis

A detecção de falhas é um serviço fundamental para o desenvolvimento de aplicações tolerantes à falhas como, por exemplo, na comunicação de grupo,

replicação de serviços e em consenso distribuído, tendo sido exaustivamente estudada na última década [Chandra and Toueg 1996, Macêdo 2000, Macêdo and Lima 2004, Chen et al. 2002, Friedman et al. 2005].

Sistemas tradicionais de detecção de falhas como o descrito por Chandra e Toueg [Chandra and Toueg 1996], utilizam a troca periódica de sinalizações *(heartbeats)* entre os processos, tempo de atraso fixo *(timeout)* e, de modo geral, pressupõem que estes processos estejam em uma rede completamente conectada. No entanto, o uso de sinalização com intervalos fixos leva a um grande número de suspeições, pois não considera as variações na carga de comunicação, nem as constantes quebras de enlace nas redes móveis. Assim, para minimizar o impacto da variação de carga de comunicação na rede e da mobilidade dos veículos nos sistemas de detecção de falhas, novas propostas mais adequadas ao ambiente das redes móveis introduziram os conceitos de *timeouts* adaptativos [Chen et al. 2002, Tai et al. 2004, Gupta and Younis 2003].

Em Chen et al. [Chen et al. 2002], o conceito clássico de *timeout* fixo é substituído pelo que o autor denomina de ponto de folga *freshness point*. Nesta proposta um novo valor de tempo τ_i^v é estimado para a chegada da *i-ésima* sinalização do processo v, independentemente do tempo em que a sinalização anterior ocorreu. A utilização de uma estimativa para a chegada da próxima sinalização incrementa a confiabilidade e robustez do detector, pois torna o tempo de detecção independente da última sinalização, reduzindo o número de falsas suspeitas geradas pelos atrasos entre sinalizações.

Um detector de falhas adaptativo é proposto em Macêdo e Lima [Macêdo and Lima 2004]. Este detector utiliza comunicação SNMP *(Simple Network Management Protocol)* e redes neurais artificiais para treinar padrões da MIB *(Management Information Base)*. Esses padrões são utilizados para perceber falhas e adaptar os *timeouts*, estimando assim a chegada da próxima sinalização. Na primeira fase o detector obtém a carga de comunicação do sistema através de agentes SNMP sobre as variáveis locais da MIB e na segunda fase a rede neural é inquerida para estimar o próximo timeout.

Outras propostas, por Tai et al. 2004 [Tai et al. 2004] e Gupta and Younis 2003 [Tai et al. 2004], apresentam um conceito hierárquico para detecção de falhas em redes móveis ad hoc. Nestas propostas a rede é dividida em agrupamentos e cada agrupamento estabelece uma visão local da rede, dentro da qual identifica nós faltosos através da troca de mensagens entre seus membros. As falhas identificadas nos agrupamentos são propagadas pela rede através de canais de comunicação interagrupamento, de modo, que um nó suspeito é conhecido pelos demais nós da rede.

Sistemas de detecção de falhas que utilizam sinalização adaptativa e organização hierárquica são bastante adequados ao ambiente das redes móveis ad hoc, fazendo com que os detectores de falhas se esquivem dos dois principais problemas destas redes: o tempo de comunicação variável e a perda de mensagens [Tai et al. 2004]. Porém, de modo geral os sistemas de detecção de falhas para redes ad hoc móveis supõem pouca variação na densidade da rede e baixa mobilidade do nós. Além disso, assumem na maioria das vezes que todos os nós da rede são alcançáveis entre si. Essas hipóteses são pouco prováveis em VANETs devido principalmente a grande mobilidade dos veículos sobre as vias.

Uma vez que as atuais propostas de detectores de falhas não são adequadas para o ambiente das redes veiculares ad hoc, nas próximas seções descreve-se um modelo de sistema e uma proposta de detector de falhas projetado para o ambiente das VANETs.

4. Modelo do Sistema

O sistema é composto por um conjunto $P = (p_1, p_2, ..., p_n)$ de processos no qual $n \in N$ e é desconhecido, mas finito. Cada processo possui um identificador único p e representa um veículo equipado, que por sua vez, representa um nó na rede. Neste modelo os termos nó, processo e veículo são intercambiáveis entre si. Um veículo é dito *equipado* quando possui capacidade de comunicação, processamento embarcado, GPS *(Global Position System)* e mapa digital das vias.

Cada processo p possui um relógio local que representa o tempo através de uma sequência T, na qual um instante de tempo t_p é um elemento de T e t_p^i representa o *i-ésimo heartbeat* difundido por um processo p. Além disso, é possível assumir que os relógios locais são sincronizados através do GPS e que a derivação entre eles não é significativa para a aplicação.

Os veículos se movimentam sobre uma mesma pista cujo sentido é único. Estes veículos podem se comunicar diretamente entre si através dos canais de comunicação, desde que estejam dentro da área de cobertura de comunicação um do outro. A área de cobertura de comunicação de um nó p é representada por um círculo de raio r_p, no qual p encontra-se no centro. Uma vez que dois nós p e q estejam um sob a área de cobertura do outro, estes nós são ditos vizinhos ou membros de uma vizinhança.

4.1. Canais de Comunicação

A comunicação entre os nós de uma vizinhança acontece através dos canais de comunicação e utiliza os eventos *enviar()* e *receber()*. Assim, para enviar uma mensagem m um processo p dispara um evento *enviar(m)* e quando q recebe esta mensagem, dispara um evento *receber(m)*. As mensagens são enviadas por difusão e uma mensagem transmitida por um nó pode ou não ser recebida por seus vizinhos.

Assume-se também um atraso de transmissão $a_p(q)$ para as mensagens de sinalização recebidas por um processo p de um vizinho q. Este atraso não é constante e pode ser calculado por:

$$a_p(q) = t_p - (t_q^i + D_q), \tag{1}$$

na qual, t_p é tempo atual em que p, t_q^i é o tempo no instante em que o processo q monta sua mensagem de sinalização e D_q é a latência em segundos, necessária para q enviar a mensagem pelo canal de comunicação. A latência D_q, descrita a seguir, pode ser calculada a partir da função utilizada para medir a qualidade do enlace entre dois nós no padrão 802.11s [Camp and Knightly 2008]:

$$D_q = \left[H + \frac{B}{C} \right], \tag{2}$$

sendo que H é um valor de *overhead* constante da camada MAC *(Media Access Control)* e indica um tempo em segundos para o processamento e envio da mensagem da

aplicação para o meio físico. A variável B indica o tamanho da mensagem em bits e C é a taxa de transmissão em bits/segundo do canal de comunicação.

Neste modelo os canais de comunicação não são confiáveis, mas uma mensagem obtida através de um evento *receber()* sempre é uma mensagem correta. Define-se também que um processo p difunde periodicamente a cada Q segundos uma mensagem m de sinalização para todos os seus vizinhos. Os campos que compõem esta mensagem são mostrados na tabela abaixo:

Tabela 1. *Campos em uma mensagem m de sinalização.*

Campo	Descrição
Id_p	identificador do nó p
v_p	velocidade de p em metros por segundo
t_p^i	i-ésimo *(timestamp)* enviado pelo processo p
$p(x, y)$	posição atual do veículo p
N_p	lista contendo os *Id* e *timeStamp* da última sinalização recebida de seus vizinhos

Como mostra a Tabela 1, cada processo p difunde periodicamente seu identificador único (Id_p), sua velocidade instantânea (v_p), seu $timestamp$ (t_p^i), sua posição ($p(x, y)$) sobre o mapa digital no momento da sinalização e uma lista contendo os Ids e $timestamps$ recebidos mais recentemente de seus vizinhos (N_p).

4.2. Modelo de Falhas

Assume-se que os processos, quando faltosos, apresentam dois tipos de comportamento: de falha por parada *(crash-fault)* ou falha por abandono da via.

Quando um processo apresenta uma falha por parada ele não é capaz de enviar ou receber mensagens. Por outro lado, um veículo que deixa a via, por exemplo, entrando em um estacionamento, não para de enviar ou receber mensagens.

Assume-se também neste modelo, que tanto falhas por parada quanto por abandono da via são permanentes. Isto porque, se o hardware falhar ou se o veículo entrar em um estacionamento, decorrerão minutos ou horas até que este veículo retorne à via. De modo, que ao retornar é provável que sua antiga vizinhança não esteja mais dentro de seu raio de comunicação e este veículo será reconhecido por seus atuais vizinhos como um novo nó na rede.

Para um processo p determinar se outro processo q em sua vizinhança abandonou ou não a via, o processo p compara os dados de sinalização recebidos de q (veja Tabela 1) com as informações de seu GPS e mapa digital das vias.

Neste modelo não são consideradas nem tratadas falhas maliciosas ou bizantinas, nas quais os processos que falham continuam a enviar mensagens, prejudicando o funcionamento do sistema. Para evitar tais falhas, podem ser utilizados, por exemplo, mecanismos de autorização que verificam a existência de intrusos na rede, porém o estudo de tais mecanismos não é foco deste trabalho. Na próxima seção descreve-se o sistema para detecção de falhas proposto neste artigo.

5. Sistema de Detecção de Falhas Proposto

As redes veiculares ad hoc estão sujeitas a grandes variações na carga de comunicação devido principalmente à mobilidade e mudanças bruscas na densidade de nós na rede. Assim, uma característica desejável para detectores de falhas a serem utilizados nestas redes, é que estes adaptem seus intervalos de espera *timeouts* juntamente com as variações na carga de comunicação [Macêdo 2000].

O detector de falhas proposto neste artigo não é perfeito e pode se equivocar, adicionando à lista de suspeitos processos corretos ou considerar processos falhos como corretos. Cada processo p tem um módulo de detecção de falhas que adiciona ou remove um nó q da lista de processos suspeitos, utilizando para isto duas informações: a informação do *timeout* de q e a percepção da vizinhança a partir dos dados contidos nas listas N_p, descrita na Tabela 1 e enviadas pelos vizinhos deste processo.

5.1. Cálculo do *Timeout*

Detectores de falhas adaptativos lidam melhor com as mudanças nas condições de comunicação da rede, modificando dinamicamente o intervalo de espera por uma sinalização. Nesta proposta, um processo p calcula um *timeout* β_q para cada um de seus vizinhos q através de:

$$\beta_q = Q + A_q + \Delta_q, \tag{3}$$

na qual, Q é o período de sinalização *(heartbeat)*, A_q é a média quadrática dos atrasos das n últimas sinalizações de q e Δ_q um tempo em segundos que varia em função da distância entre dois processos vizinhos.

Para calcular a componente A_q, cada processo p mantém uma lista L_n com o atraso das últimas n sinalizações de q. A partir destes atrasos calcula-se A_q da seguinte maneira:

$$A_q = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left[a_p(q)\right]^2}, \tag{4}$$

na qual, n é o número de sinalizações recebidas de q e armazenadas pelo processo p na lista L_n e $a_p(q)$ é o atraso entre cada uma destas sinalização.

A média quadrática é utilizada nesta proposta, pois é mais sensível que outras médias às mudanças de valores (neste caso, os atrasos entre sinalizações). Assim, é possível uma adaptação mais rápida do *timeout* às variações da carga de comunicação, porém sem desconsiderar o histórico de sinalizações anteriores.

O tamanho de L_n afeta o comportamento do detector, pois caso seja muito pequeno o detector será pouco tolerante às pequenas variações na carga da rede e caso seja muito grande, variações bruscas na carga de comunicação serão amenizadas e não necessariamente representarão o estado atual da comunicação entre os processos. Assim, o tamanho da lista L_n deve ser ajustado levando em consideração alguns parâmetros, como por exemplo, mobilidade dos nós, período da sinalização e restrições temporais das aplicações que utilizam o detector.

A componente Δ_q acrescenta ao *timeout* um valor de tempo que aumenta à medida que dois nós vizinhos se afastam um do outro. Este acréscimo de tempo ao *timeout*

minimiza o impacto da atenuação de sinal gerada pelo aumento da distância entre dois vizinhos e variações na densidade do tráfego sobre a via. Tanto a atenuação de sinal quanto o aumento da densidade de nós na vizinhança de um veículo, aumentam a perda ou o atraso das mensagens de sinalização.

Considerando estes fatores, a componente Δ_q é composta por um termo α que define um coeficiente mínimo de atraso entre as sinalizações e um segundo termo variável associado à distância entre os veículos:

$$\Delta_q = \begin{cases} \alpha & \text{se} \quad d(p,q) > r_p \\ \alpha + \left[k \frac{d(p,q)}{r_p} \right] & \text{se} \quad d(p,q) \leq r_p, \end{cases} \tag{5}$$

na qual, α é um atraso de tolerância constante na sinalização entre dois processos, $d(p,q)$ é a distância euclidiana entre estes processos, r_p é o raio de comunicação do processo p e k é um fator de ponderação que ajusta a importância relativa da entre estas distâncias em relação aos outros termos na Equação 3. De modo, que quanto mais dois processos vizinhos se afastam um do outro, maior o valor do intervalo de espera da sinalização $(timeout)$.

5.2. Percepção da Vizinhança

A percepção que um processo p tem de sua vizinhança depende das sinalizações recebidas de seus vizinhos q e das informações contidas na lista N_q de cada processo q.

Assim, um nó p armazena as informações sobre cada sinalização recebida diretamente de seus vizinhos a um salto, além das informações da lista N_p com os Ids e *timestamps* da vizinhança de seus vizinhos. Com isto, um veículo p pode construir uma imagem da rede além do seu raio de comunicação, permitindo que: i) caso não receba a sinalização de um vizinho dentro de seu *timeout*, os dados contidos na lista N_p possam ser utilizados para evitar que um veículo p insira um vizinho q em sua lista de suspeitos e; ii) caso um veículo p suspeite de um vizinho q, o tempo de suspeita pode ser minimizado, pois p pode perceber que q está ativo através das mensagens enviadas por outro vizinho.

Utilizar informações sobre processos a mais de um salto de comunicação torna o detector de falhas mais confiável e robusto diante da mobilidade e atrasos na comunicação.

5.3. Detector de Conectividade

Nas redes veiculares ad hoc os nós podem atingir velocidades relativas de até 50 m/s e muitas suspeitas de falha podem surgir da quebra de enlaces. A principal utilidade de um detector de conectividade em um ambiente de grande mobilidade é evitar que enlaces perdidos devido a movimentação dos nós sejam confundidos com falhas.

Mesmo que não seja possível antecipar as ações individuais dos motoristas, o comportamento do tráfego segue um conjunto de normas conhecidas, como por exemplo, os veículos em uma pista trafegam no mesmo sentido e a velocidade e posição dos veículos não mudam brusca e instantaneamente. Assim, é possível utilizar algumas destas características juntamente com informações difundidas periodicamente pelos veículos para determinar se um enlace entre dois processos vizinhos ainda é válido.

O detector de conectividade $DC_p(q)$ proposto neste artigo é executado em todos os processos. Utilizando este detector, um veículo p calcula se um vizinho q ainda se encontra dentro de sua área de cobertura de comunicação. Para tanto, o veículo p utiliza as seguintes informações: a última velocidade conhecida do vizinho q, raio de comunicação de r_p, a última posição conhecida de ambos os veículos P^p_{atual} e $P^q_{anterior}$ e o tempo da última sinalização t^i_q do vizinho q e o tempo atual t_p em p. O algoritmo a seguir descreve de forma simplificada o funcionamento do detector de conectividade.

```
1  begin
2      P^p_atual ← posição atual de p
3      P^q_anterior ← última posição conhecida de q
4      P^q_estimada ← calcular a posição estimada para o processo q
5      if (|P^p_atual − P^q_estimada|) < r_p then
6          retorna true;
7      else
8          O processo p retira q de sua lista de vizinhos;
9          retorna false;
10     end
11 end
```

Algoritmo 1: Detector de Conectividade

O algoritmo 1, (P^p_{atual}) representa a posição atual do processo p e ($P^q_{anterior}$) a última posição conhecida do processo q (linhas 2 e 3). A próxima posição estimada para um processo q ($P^q_{estimada}$) é calculada utilizando os dados da última sinalização recebida do processo q pelo processo p, juntamente com os dados do GPS e mapa digital das vias (linha 4). Se a diferença entre as posições de p e q for menor que o raio de comunicação p, ou seja q está dentro de r_p, então o enlace é considerado válido, caso contrário o nó é retirado da lista de vizinhos o enlace é considerado inválido (linha 5 a 10). Na próxima seção apresenta-se o algoritmo de detecção de falhas proposto e no qual o detector de conectividade é utilizado.

5.4. Algoritmo de Detecção de Falhas Proposto

O sistema de detecção de falhas proposto executa paralelamente três tarefas, T1, T2 e T3. A tarefa T1 é responsável por receber as sinalizações e atualizar os dados de cada vizinho do processo p. Na tarefa T2, o detector de falhas verifica continuamente se algum processo q deve ou não ser inserido na lista de suspeitos e a tarefa T3 é responsável pela recuperação das falsas suspeitas.

O algoritmo simplificado do detector de falhas proposto é descrito a seguir e funciona da seguinte maneira: considere dois processos p e q, no qual p monitora q, que por sua vez difunde uma mensagem m a cada Q segundos.

Na tarefa T1, assim que um processo p recebe uma mensagem de sinalização de um vizinho q, o processo p atualiza os dados deste vizinho e calcula o novo valor de *timeout* β_q (linhas 3 a 5). Na linha 6 o processo p atualiza a variável t^i_q com o valor de *timestamp* mais recentemente recebido do processo q. Isto é, o processo p atualiza a variável t^i_q com o último $timestamp$ recebido direta ou indiretamente do processo q e cujo valor é o mais próximo ao tempo atual t_p no processo p.

```
1  Tarefa: T1
2  repeat
3      When receber(m)
4      Atualiza dados do vizinho q a partir da mensagem de sinalização
5      Calcula β_q
6      t_q^i ← timestamp mais recente do processo q
7  until forever ;
8  Tarefa: T2  /* Detecção de Falhas */
9  repeat
10     When (t_p − t_q^i) > β_q: timeout expirou
11     if (q ∉ lista de suspeitos) and (DC_p(q) = True) then
12         Inserir q na lista de suspeitos de p
13     end
14 until forever ;
15 Tarefa: T3 /* Recuperação de Falhas */
16 repeat
17     if (DC_p(q) = True) and (q ∈ lista de suspeitos) and ((t_p − t_q^i) <= β_q)
       then
18         Retirar q da lista de suspeitos
19 until forever ;
```

Algoritmo 2: Detector de Falhas

A geração de suspeitas de falhas ocorre na tarefa T2. Ao iniciar a detecção de falhas, p verifica se existe algum vizinho q não suspeito que tenha ultrapassado o *timeout* (linha 10). Quando um processo q ainda não suspeitado ultrapassa seu *timeout* β_q o processo p executa o detector de conectividade e, se o enlace ainda for considerado válido, o processo p insere q na lista de suspeitos (linhas 11 a 13). Finalmente na tarefa T3 é executado o processo de recuperação de falhas. Se um processo p recebe uma sinalização de um processo q suspeito, seja diretamente ou através de um de seus vizinhos, p verifica se esta sinalização está dentro do *timeout* β_q e, se isto for verdadeiro, retira q da lista de suspeitos (linhas 17 e 18). Em ambas as tarefas T1 e T2, o detector de conectividade informa aos processos a validade dos enlaces antes de definir se um processo é ou não suspeito.

6. Avaliação do Detector de Falhas Proposto

Nesta seção apresentam-se os resultados da aplicação dos algoritmos descritos nas seções anteriores em um cenário de simulação. A eficiência do detector de falhas proposto foi medida analisando seu desempenho em relação à três métricas: i) número de falsas suspeitas; ii) tempo médio de detecção de falhas e; iii) tempo médio para recuperação de falsas suspeitas. Uma quarta métrica avalia a influência do atraso entre as sinalizações na geração de falsas suspeitas. Isto é, como o tamanho da lista L_n, utilizada para o cálculo do atraso médio entre as sinalizações influencia na precisão do detector.

O número de falsas suspeitas é o somatório das suspeitas individuais produzidas pelos processos durante a simulação subtraído do número de falhas que realmente ocor-

reram na vizinhança destes processos. Esta métrica indica a confiabilidade do detector diante da mobilidade e mudanças na carga de comunicação da rede.

As métricas que avaliam o tempo médio para detecção de uma falha e o tempo médio para recuperação de falsas suspeitas indicam a velocidade de detecção, ou seja, o tempo de resposta do detector de falhas proposto. O tempo de detecção de falha consiste em um intervalo entre o momento no qual a falha aconteceu e o instante em que ela foi detectada por um processo correto. Já o tempo para recuperação de falhas é o intervalo de tempo decorrido entre a inserção e retirada de um processo da lista de suspeitos.

6.1. Modelo de Simulação

A simulação foi feita utilizando o simulador de eventos discretos OMNET++ *(Objective Modular Network Testbed in C++)* [Varga et al. 2001]. Define-se que os veículos possuem um raio de comunicação $r_p = 150$. O padrão de comunicação utilizado é o IEEE 802.11 [IEEE 2007] com um modelo de atenuação de sinal *(path-loss, fading e shadowing)* implementado no pacote MiXiM [Kopke et al. 2008]. A taxa de comunicação e o *overhead* da camada MAC adotados são respectivamente: $C = 2Mbps$ e $H = 0.01s$

O cenário de simulação consiste em uma via com 4000m de comprimento, na qual os veículos se movem no mesmo sentido e com velocidades variando entre 11m/s e 22m/s, aproximadamente 40 km/h e 80 km/h. Foram realizados quatro experimentos, cada um com quatro diferentes densidades de veículos sobre a via (50, 100, 200 e 400 veículos).

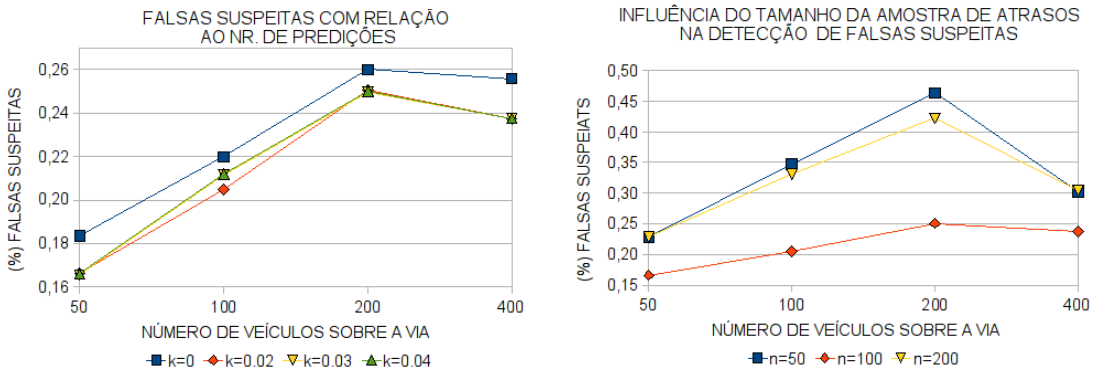
A duração de cada experimento é de 100 segundos e o período de sinalização $Q = 0.1s$. O valor $\alpha = 0.02s$ é constante em todos os experimentos, enquanto os valores de k foram variados entre 0s e 0.04s. Assim, variando os valores em k é possível medir qual a contribuição do fator Δ_q na precisão e o tempo de resposta do detector de falhas em cada uma das densidades de nós na rede.

Define-se que a janela L_n, que armazena os atrasos entre as sinalizações de um veículo, é de tamanho $n = 100$, para todos os processos no sistema. Durante a simulação 20% dos processos sofrem falhas. Estas faltas ocorrem aleatoriamente ao longo da simulação e um processo faltoso não retorna ao sistema.

6.2. Resultados do Experimento e Discussão dos Resultados

As Figuras 2 e 3 mostram resultados relacionados à confiabilidade do detector de falhas proposto. Na Figura 2(a) é mostrado o percentual de falsas suspeitas em relação ao número de predições realizadas pelos processos durante toda a simulação. Note-se que o aumento da densidade de veículos sobre a via aumenta o número de falsas suspeitas. Este aumento da densidade também se reflete em um aumento do número de sinalizações *(heartbeats)* recebidas por cada um dos processos sobre a via. No entanto, o número absoluto de falsas suspeitas não aumenta proporcionalmente ao número absoluto de sinalizações, fazendo com que haja uma pequena queda no percentual de falsas suspeitas no cenário de maior densidade. Este resultado mostra que o detector de falhas proposto oferece boa confiabilidade diante das variações de densidade do tráfego e o percetual de falsas suspeitas não cresce proporcionalmente ao aumento desta densidade.

A estabilidade de um detector de falhas diante das variações de densidade é uma característica muito importante para DF em VANETs, já que a densidade do tráfego pode



(a) *Percentual de Falsas Suspeitas em Relação ao Número de Predições*

(b) *Influência do Tamanho de L_n Percentual de Falsas Suspeitos*

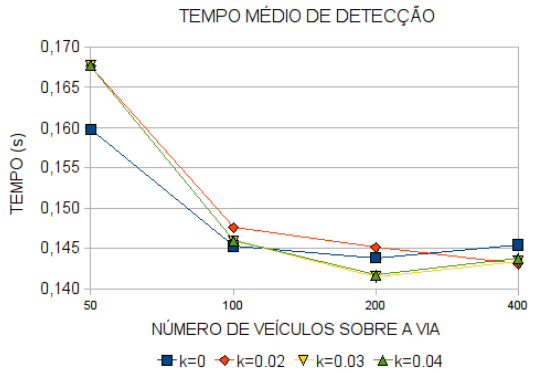
Figura 2. Avaliação da Confiabilidade do Detector de Falhas

variar bruscamente sobre as vias, por exemplo, a densidade junto aos semáforos é de modo geral, muito maior que ao longo da pista. Outro aspecto que pode ser notado na Figura 2 é que a inserção de um valor positivo para k reduz o número de falsas suspeitas, melhorando o desempenho do DF.

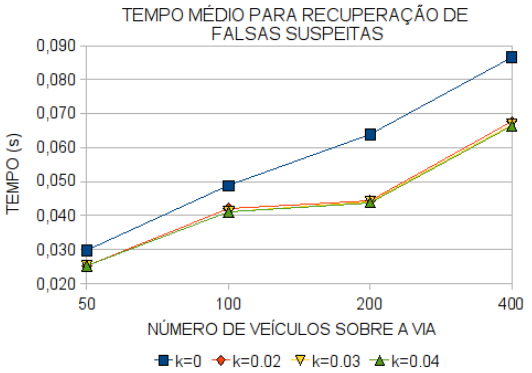
Note-se na Figura 2(b), que o tamanho da amostra dos atrasos entre as sinalizações, armazenadas em L_n, afeta o desempenho do DF e que sua escolha deve ser criteriosa e não aleatória. Os critérios para determinar o tamanho da amostra dependem de fatores como período de sinalização, restrições temporais da aplicação, densidade da rede, etc. Pode-se observar nesta figura que: i) amostras muito pequenas deixam o detector de falhas muito sensível às mudanças de carga de comunicação na rede; ii) amostras muito grandes amenizam demasiadamente a atuação do detector diante de mudanças bruscas nesta carga e portanto, em ambos os casos o detector de falha perde confiabilidade.

Na Figura 3(a) mostra-se o tempo médio de detecção de falhas durante a simulação. Mais uma vez o detector proposto mostra estabilidade diante da variação de densidade na rede, condição essencial para o ambiente de comunicação das redes veiculares ad hoc. Nota-se também que o aumento do valor em k não afeta significativamente o tempo médio de detecção. Por outro lado, se tempo médio de detecção não é afetado pelo aumento da constante k, o percentual de falsas suspeitas diminui quando $k > 0$, como mostra a Figura 2(a). Isto é, há um aumento na confiabilidade do detector sem redução do seu tempo de resposta.

A Figura 3(b) indica o tempo médio de recuperação de falsas suspeitas. Note que o tempo médio de recuperação de falhas aumenta juntamente com o aumento da densidade. No entanto, seu valor é relativamente baixo em relação ao período de sinalização *(heartbeat)*. Isto ocorre porque apesar da densidade afetar a carga de comunicação e, portanto, o atraso entre as sinalizações, o aumento de vizinhos colabora para que um processo recupere rapidamente uma falsa suspeita através dos dados contidos nas listas l_p enviadas por seus vizinhos (Veja Tabela 1).



(a) *Tempo Médio de Detecção de Falha*



(b) *Tempo Médio de Recuperação de Falsas Suspeitas*

Figura 3. Tempo de Resposta do Detector de Falhas

7. Conclusões

A capacidade de detectar falhas e reduzir o número de falsas suspeitas, é um componente vital para a aplicações distribuídas em redes veiculares ad hoc, especialmente para aquelas com restrições temporais críticas e que baseiam suas decisões a partir das condições de sua vizinhança. Aplicações como, por exemplo, controle de tráfego em tempo real, semáforos atuados pelo tráfego, sistemas anticolisão.

Apesar da importância, as atuais propostas de sistemas de detecção de falhas não foram projetadas para ambientes com alta mobilidade dos nós e variações rápidas na densidade da rede. Este artigo propõe um detector de falhas adaptado ao ambiente das redes veiculares ad hoc. O detector de falhas proposto possui um mecanismo de detecção de conectividade que tenta minimizar o impacto da mobilidade no processo de detecção de falhas e outro mecanismo que adapta o tempo de espera de uma sinalização *timeout*, as condições de comunicação no entorno de cada processo. As simulações mostram que o detector de falhas proposto adapta-se bem às variações de densidade e comunicação das VANETs, gerando um pequeno número de falsas suspeitas e uma rápida recuperação destas falsas suspeitas. Em trabalhos futuros este detector será utilizado pelos processos para reconhecer suas vizinhanças. A partir do conhecimento que cada um dos processos tem de seus vizinhos saudáveis, estes processos se unirão em agrupamentos *(clusters)*. Estes agrupamentos serão utilizados para coletar e enviar os dados individuais dos veículos para a infra-estrutura de beira de pista, como é proposto por Cambruzzi et al. [Cambruzzi et al. 2009].

Referências

Cambruzzi, E., Farines, J. M., and Kraus Jr., W. (2009). Um Algoritmo Baseado em Peso para Formação e Manutenção de Agrupamentos em Redes Veiculares. *27o-Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos-Recife-Brazil*.

Camp, J. and Knightly, E. (2008). The ieee 802.11 s extended service set mesh networking standard. *IEEE Communications Magazine*, 46(8):120–126.

Chandra, T. and Toueg, S. (1996). Unreliable failure detectors for reliable distributed systems. *Journal of the ACM (JACM)*, 43(2):225–267.

Chen, W., Toueg, S., and Aguilera, M. (2002). On the quality of service of failure detectors. *IEEE Transactions on computers*, pages 561–580.

Friedman, R., Tcharny, G., and LTD, I. (2005). Evaluating failure detection in mobile ad-hoc networks. *Int. Journal of Wireless and Mobile Computing*, Vol. 1(8).

Gunter, Y., Wiegel, B., and Grossmann, H. (2007). Cluster-based medium access scheme for vanets. *IEEE Intelligent Transportation Systems article (ITSC)*, pages 343–348.

Gupta, G. and Younis, M. (2003). Fault-tolerant clustering of wireless sensor networks. *Proceedings of IEEE WCNC*, 3:1.

Harri, J., Fiore, M., Filali, F., Bonnet, C., Casetti, C., and Chiasserini, C. (2007). A realistic mobility simulator for vehicular ad hoc networks. Technical report, Technical Report RR-05-150, Institut Eurecom.

Ibrahim, K. and Weigle, M. (2008). CASCADE: Cluster-based accurate syntactic compression of aggregated data in VANETs. *2008 IEEE GLOBECOM Workshops*.

IEEE (2007). (draft) standard for information technology - telecommunications and information exchange between systems-local and metropolitan area networks - specific requirements. IEEE Computer Society, New York, USA - http://standards.ieee.org/getieee802/download/802.11-2007.pdf.

Kopke, A., Swigulski, M., Wessel, K., Willkomm, D., Haneveld, P., Parker, T., Visser, O., Lichte, H., and Valentin, S. (2008). Simulating wireless and mobile networks in OMNeT++ the MiXiM vision. *Proceedings of the 1st international article on Simulation tools and techniques for communications, networks and systems & workshops*.

Macêdo, R. (2000). Failure Detection in Asynchronous Distributed Systems. *2nd Workshop on Tests and Fault-Tolerance. Curitiba-PR, Brazil*.

Macêdo, R. and Lima, F. (2004). Improving the quality of service of failure detectors with SNMP and artificial neural networks. *Anais do 22o. Simpósio Brasileiro de Redes de Computadores, Gramado, RS, Brazil*, pages 583–586.

O'Gorman, E. and Eng, B. (2002). Using Group Communication to Support Inter-Vehicle Coordination.

Ohta, T., Inoue, S., and Kakuda, Y. (2003). An adaptive multihop clustering scheme for highly mobile ad hoc networks. *The Sixth International Symposium on Autonomous Decentralized Systems (ISADS)*, pages 293–300.

Taha, M. and Hasan, Y. (2007). VANET-DSRC Protocol for Reliable Broadcasting of Life Safety Messages. In *2007 IEEE International Symposium on Signal Processing and Information Technology*, pages 104–109.

Tai, A., Tso, K., and Sanders, W. (2004). Cluster-Based Failure Detection Service for Large-Scale Ad Hoc Wireless Network Applications. *Proceedings of the 2004 International article on Dependable Systems and Networks*.

Varga, A. et al. (2001). The OMNeT++ discrete event simulation system. *Proceedings of the European Simulation Multiarticle*, pages 319–324.

Xu, K. and Gerla, M. (2002). A heterogeneous routing protocol based on a new stable clustering scheme. *MILCOM 2002. Proceedings*, 2:838–843.