

UNIVERSIDADE FEDERAL DO PARANÁ

NICOLE MARTINELLI BRUM

**ESTUDO COMPARATIVO ENTRE ALGORITMOS DE
HANDOFF EM REDES VEICULARES**

**CURITIBA
2017**

NICOLE MARTINELLI BRUM

**ESTUDO COMPARATIVO ENTRE ALGORITMOS DE
HANDOFF EM REDES VEICULARES**

Artigo apresentado como requisito parcial à
conclusão do Curso de Ciência da Computação em 2017, do
Setor de Ciências Exatas, da Universidade Federal do
Paraná.

Orientador: Prof. Dr. Luiz Carlos Pessoa Albini

**CURITIBA
2017**

RESUMO

Redes veiculares são redes nas quais os nós da rede são, principalmente, veículos. Através delas, é possível implantar aplicações inteligentes para os sistemas de transporte (ITS - *Intelligent Transport Systems*), como notificações de acidentes nas estradas, más condições de pistas, informações de tráfego, etc. Entretanto, existe certa dificuldade na implantação desse tipo de aplicação, dadas as características dinâmicas de um cenário veicular. Um dos processos de rede que é dificultado por essa dinamicidade do cenário é o processo de *handoff*, que ocorre quando uma sessão de dados passa de um ponto de acesso para outro (como, por exemplo, a troca de antenas em uma rede celular). O presente trabalho tem como objetivo analisar diferentes algoritmos para *handoff* em redes veiculares na literatura, bem como simular suas execuções em um ambiente veicular. O produto final é uma definição de qual dos algoritmos analisados seria mais adequado para o cenário geral de *handoff* vertical entre duas tecnologias específicas (UMTS e WLAN), deixando para a posterioridade a compilação de simulações e dados sobre cada um dos algoritmos.

Palavras-chave: Redes veiculares. Handoff. Algoritmos.

1 INTRODUÇÃO

Na última década, observou-se um aumento do número de veículos equipados com dispositivos capazes de tornar a experiência de locomoção muito mais confortável para o condutor e para os passageiros, como sensores de proximidade, indicadores digitais de velocidade e sistemas de som e vídeo automotivo [1]. Entretanto, esses dispositivos limitam-se à interação entre pessoas e o próprio veículo, dependendo de sensores para captar sinais dos ambientes interno ou externo e transformar esses sinais em informações relevantes para os ocupantes.

A evolução desse tipo de tecnologia é a comunicação entre os próprios veículos. A criação de algo como uma “Internet dos Veículos” [2] possibilitaria a criação de uma série de outras aplicações que melhorariam a experiência de locomoção e a segurança no trânsito, como sistemas de prevenção de colisão, avisos sobre obstáculos em pistas, notificações imediatas de infrações de trânsito, etc. Essas aplicações dependem da comunicação entre veículos ou entre um veículo e algum outro ponto de acesso, e fazem parte do que chamamos de Sistemas Inteligentes de Transporte (ITS - *Intelligent Transport Systems*) [1], cuja implantação poderia trazer uma série de benefícios para os indivíduos, a economia e o meio ambiente.

No entanto, a implantação efetiva desse tipo de sistema esbarra em alguns problemas técnicos característicos do cenário veicular. A alta mobilidade e alta velocidade dos veículos, por exemplo, são grandes problemas para estabelecimento de uma comunicação de qualidade através de meios sem fio, dado o alto número de desconexões a que os elementos da rede estariam sujeitos.

De acordo com [1], no cenário de uma rede veicular na qual veículos de comunicariam com pontos fixos (antenas), quando um nó da rede (ou seja, um veículo) sai da cobertura deste ponto fixo, a conexão é perdida. Da mesma maneira como ocorre em redes celulares, a comunicação só pode ser restabelecida quando o veículo volta à cobertura de uma outra antena. Chamamos a mudança da sessão de comunicação do veículo de uma antena para outra de *handoff* (ou *handover*). Este trabalho propõe-se a estudar algumas das maneiras como *handoffs* acontecem em redes veiculares, e qual delas seria mais adequada para a minimização do impacto negativo dessas desconexões neste tipo de cenário.

O trabalho apresenta a simulação e análise de três algoritmos para *handoff* em redes veiculares propostos em artigos recentes. Foi utilizado o simulador The ONE (*Opportunistic Network Environment*) como base para a simulação, e o mesmo teve algumas funcionalidades adicionadas para a reprodução dos parâmetros específicos dos algoritmos escolhidos. Tanto o código original do simulador quanto as funções adicionais foram escritos na linguagem Java, utilizando o Eclipse como ambiente de desenvolvimento e *debugging*. O produto final é uma definição de qual dos algoritmos analisados seria mais adequado para o cenário veicular, deixando para a posterioridade a compilação de simulações e dados sobre cada um dos algoritmos.

O próximo capítulo contém um breve embasamento teórico dos tópicos relacionados a redes veiculares e ao processo de *handoff*, levando em conta apenas os aspectos relevantes ao escopo do presente trabalho. No terceiro capítulo, é apresentado um panorama dos trabalhos relacionados e quais oportunidades estes trabalhos dão para a continuidade do estudo do assunto. No quarto capítulo, é apresentado o resumo dos artigos analisados, bem como as modificações feitas para que os mesmos fossem simulados no software escolhido. No quinto capítulo, é detalhado o processo de simulação e apresentados os resultados relevantes da análise. No sexto capítulo, discorre-se sobre os resultados obtidos. No oitavo capítulo apresenta-se a conclusão, bem como um breve resumo dos itens fora do escopo e perspectivas futuras deste estudo.

2 REDES VEICULARES

Redes Veiculares (VANETs - *Vehicular Ad Hoc Networks*) são consideradas um caso particular de rede *Ad Hoc*, ou seja, um caso particular de rede descentralizada.

Nas redes veiculares, veículos automotores servem como nós de uma rede, conectando-se entre si ou com elementos externos e encaminhando informações uns aos outros [3].

Apesar da possibilidade da existência dessa característica descentralizada, existem outros modelos de rede veicular que contam com algum tipo de ponto central de comunicação. Categoriza-se as redes veiculares em dois principais tipos: as redes V2V (*Vehicle to Vehicle*) e V2I (*Vehicle to Infrastructure*). Nas redes V2V, os carros se comunicam apenas com outros carros, funcionando como o equivalente às redes P2P (*Peer to Peer*) no ambiente veicular. Nas redes V2I, os carros se comunicam com pontos fixos distribuídos ao longo de estradas e rodovias. Para que estas comunicações sejam possíveis, cada veículo que se comunica dentro de uma rede veicular deve estar equipado com pelo menos um dispositivo físico para o fornecimento de uma interface de rede. Este dispositivo é conhecido como *On Board Unit*, ou OBU. Para a comunicação V2I, é necessária também a existência de dispositivos fixos ao longo das ruas e estradas. Estes dispositivos são conhecidos como *Road Side Units*, ou RSUs. Existe ainda a comunicação I2I, que envolve a troca de dados entre RSUs [4].

Podemos destacar algumas particularidades das redes veiculares [5]:

1) Desconexões frequentes. Como os carros estão em constante movimento, é comum que os nós se comuniquem com vários outros ao longo de seu caminho, sendo obrigados a trocar várias vezes o seu parceiro de comunicação. Veículos automotores podem também atingir altos níveis de velocidade quando em rodovias, o que pode diminuir o intervalo de tempo para a comunicação com outros nós. Como resultado do dinamismo do ambiente veicular, é de suma importância que os nós não dependam da permanência de um único nó no ambiente, ou seja, que não exista um único nó responsável pelo encaminhamento de informações aos outros. Estas características fazem com que seja particularmente importante encontrar uma maneira segura e eficiente de definir qual é o próximo nó com o qual um carro deve se comunicar, e durante quanto tempo esta comunicação pode acontecer antes que o nó seja trocado.

2) Padrões de mobilidade. Veículos seguem seus caminhos ao longo de ruas e estradas, movimentando-se, na maioria das vezes, de acordo com as regras definidas pelo código de trânsito regional (limite máximo de velocidade, pista padrão para ultrapassagem, direção específica de movimento) o que torna mais fácil a previsão de onde um nó estará em um dado instante de tempo, dada uma localização anterior e sua velocidade. A utilização de dispositivos de GPS pode tornar ainda mais fácil esta previsão, dado que o motorista pode inserir os dados de destino e seguir o caminho indicado.

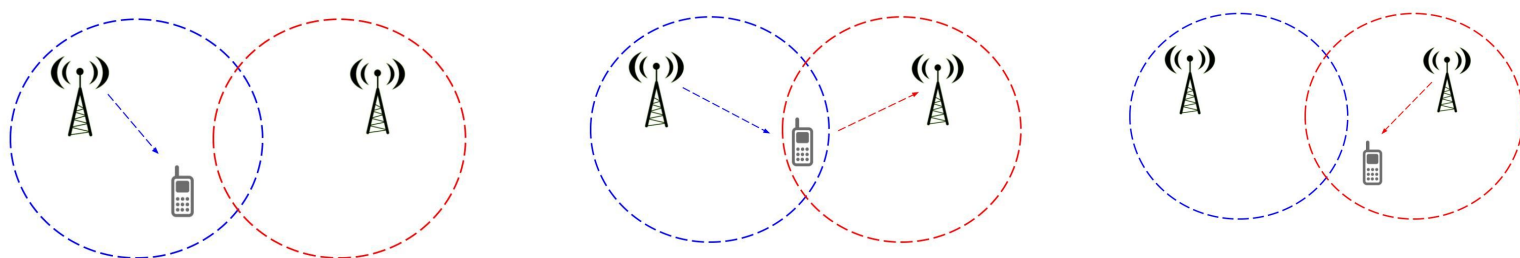
3) Consumo de energia. Diferente de outros protocolos para dispositivos móveis, os protocolos para redes veiculares não precisam ser tão eficientes energeticamente.

Isto acontece devido ao fato de que carros possuem baterias automotivas, que podem ser consideradas fontes de energia quase infinitas para sistemas embarcados pequenos.

Além das necessidades de rede de aplicações comuns, existe uma demanda para aplicações de sistemas de tráfego inteligentes (ITS - *Intelligent Traffic Systems*). A eficiência dos protocolos de redes veiculares poderia abrir muitas portas para o desenvolvimento de aplicações como coleta em tempo real de informações de tráfego, condições da estrada, informações sobre outros veículos, etc. Apesar de aplicações como essas já existirem, elas não são tão comuns, mas poderiam se tornar ubíquas caso existisse um meio eficiente e barato de implementá-las. A existência de soluções em ITS tem grande valor comercial, por apresentar a possibilidade de economia em emissão de poluentes, consumo de combustível e tempo [6]. Dada esta necessidade de protocolos eficientes, este trabalho busca estudar um dos principais problemas no design de protocolos para redes veiculares: a troca de interfaces de rede (também conhecida como *handoff*) durante a execução de aplicações.

3 HANDOFF/HANDOVER EM REDES VEICULARES

No contexto de comunicações celulares, *handoff* (ou *handover*) é o processo de transferência de uma sessão de dados entre dois pontos de acesso diferentes de forma transparente ao usuário. Por exemplo, se um usuário A está sob a cobertura da célula 1 e se move para a área de cobertura da célula 2, é necessária a transferência da sessão da célula 1 para a célula 2, com o objetivo de manter o fluxo de dados.



Nas figuras acima, é possível visualizar o estado das comunicações antes, durante e depois da ocorrência de um *handoff* em uma rede celular.

Categoriza-se o processo de *handoff* em dois principais tipos: o *handoff* horizontal é feito quando há uma troca entre duas tecnologias idênticas de rede (como no exemplo dado, no qual há uma troca entre dois pontos de rede celular). O *handoff*

vertical, por sua vez, é feito quando há uma troca entre duas tecnologias de rede diferentes (como, por exemplo, uma troca entre WiFi e 3G, de maneira transparente ao usuário).

Podemos ainda dividir os cenários de *handoff* em duas outras categorias: *soft handoff* e *hard handoff* [7]. A ocorrência de *soft handoff* está relacionada à passagem de um bastão entre dois pontos de acesso: há uma negociação para que a conexão seja transferida de um ponto para outro. Neste caso, a conexão com o ponto de destino é iniciada antes que a conexão com o ponto anterior seja terminada, então é possível que um mesmo nó esteja enviando e/ou recebendo informações de dois pontos de acesso diferentes em um dado instante de tempo. Em contrapartida, na ocorrência de *hard handoff* há uma quebra imediata da conexão com um ponto de acesso quando um novo ponto é detectado como novo destino do *handoff*. Neste caso, um nó da rede recebe informações apenas do seu novo ponto de acesso, perdendo possíveis pacotes pendentes que ainda seriam enviados pelo ponto anterior.

Quando levamos em consideração o contexto de redes celulares tradicionais (UMTS, GSM, etc.) na análise do processo de *handoff*, é fácil visualizar como isso acontece [8] e como é possível que aplicações móveis funcionem eficientemente na visão dos usuários finais. Agora levemos em consideração um cenário diferente: o cenário de redes veiculares. Podemos tomar como exemplo uma configuração simples de uma rede V2I, na qual veículos se deslocam ao longo de rodovias e recebem, através dos dispositivos dedicados instalados nos próprios carros (OBUs), pacotes enviados por pontos de acesso dispostos ao longo das rodovias. Por enquanto, podemos abstrair detalhes como a tecnologia por trás dos pontos de acesso e quais são as aplicações relevantes para este cenário.

Percebe-se que, no contexto de redes veiculares, existe uma particularidade que torna um grande desafio a decisão e execução deste processo de troca: a velocidade dos nós. As altas velocidades nas quais veículos automotores se movem tornam muito pequena a janela de tempo que existe para a comunicação entre um nó da rede (neste caso, um carro em si) e um ponto de acesso. Para que o mesmo tipo de aplicação que funciona em dispositivos móveis tradicionais funcione no contexto de redes veiculares, é preciso que o processo de passagem da conexão entre pontos de acesso seja rápida e eficiente.

4 TRABALHOS RELACIONADOS

O objetivo deste trabalho é realizar um estudo comparativo entre diferentes métodos para a decisão de *handoff* em redes veiculares. O produto final do trabalho é uma análise de qual dos algoritmos seria melhor para o *handoff* em redes veiculares. Para obter noções gerais de como ocorrem os processos de *handoff* e de como podem

ser melhorados, foram analisados também artigos que não tratam necessariamente do contexto de redes veiculares, mas sim do processo de *handoff* no geral. Depois desta análise geral, foram analisados os principais artigos sobre *handoff* em redes veiculares e, por último, foram analisados outros trabalhos de comparação entre algoritmos de *handoff*, com o objetivo de compilar características e métodos de interesse para este estudo. A seguir são listados os pontos mais interessantes da análise de trabalhos relacionados.

Um artigo de importância histórica para o estudo do processo de *handoff* geral é citado em [9]. Os autores apontam alguns dos principais problemas no *design* de algoritmos eficientes para a realização do *handoff*, que ainda devem ser solucionados por qualquer algoritmo que se proponha a realizar a tarefa satisfatoriamente. Alguns destes problemas são a necessidade de uma baixíssima latência no processo, com o objetivo de não deixar a troca de conexão transparecer para o usuário; a necessidade de economia de energia, o que se torna muito atual se levarmos em consideração o contexto de telefonia móvel de hoje, no qual a maioria dos aparelhos tem uma baixa duração de bateria; a minimização do *overhead* de largura de banda, ou seja, a necessidade de não sobrecarregar a rede com o processamento necessário para a realização dos *handoffs*; e a necessidade de que os algoritmos sejam compatíveis com as tecnologias já implementadas, pois de pouco adianta um algoritmo revolucionário para *handoff* se ele não pode ser utilizado em conjunto com as tecnologias atuais. O modelo proposto pelos autores serve como base para uma série de outros algoritmos que contam com agentes externos para o auxílio no processo de *handoff* (isto é, o processo não depende exclusivamente da comunicação entre um ponto de acesso e o dispositivo que deseja se conectar). Este modelo torna a comunicação similar à estrutura do MobileIP, na qual um agente principal (*Home Agent*) é responsável por detalhes da troca de sessão de dados. O segundo algoritmo que será simulado neste trabalho tem uma arquitetura muito similar a esta.

Outro trabalho bem conhecido na literatura é [10]. Neste artigo há a proposição de uma nova arquitetura, na qual um servidor de *handoff* intermediaria o contato entre a Internet e os hosts, seguindo a ideia da existência de um agente externo responsável por detalhes da comunicação. Os autores direcionam a atenção à necessidade de baixa latência e baixa perda de pacotes no processo. Os objetivos dos autores foram comprovados através de testes realizados apenas para o *handoff* entre 802.11b e Ethernet (teste para o ambiente *indoor*) e 802.11b e 1xRTT (teste para o ambiente *outdoor*).

No que diz respeito a técnicas de *handoff* para redes veiculares, o trabalho [11] apresenta uma maneira de realizar o processo de *handoff* de maneira a cobrir diferentes tecnologias. A proposta é que o algoritmo seja universal, podendo ser aplicado a redes heterogêneas no geral, não apenas para o cenário veicular. Este

algoritmo utiliza como principal parâmetro a força do sinal recebido (RSS - *Received Signal Strength*) para decidir se um nó da rede é candidato à realização do *handoff*. O algoritmo também leva em consideração a economia de energia de um nó e o balanceamento de carga entre diferentes pontos de acesso. Apesar de se propor a ser um algoritmo universal para redes heterogêneas (inclusive redes veiculares), os autores falham em apresentar testes baseados em ambientes com a mesma dinamicidade do ambiente veicular, já que não foram especificadas diferenças decorrentes da variação de velocidade dos nós ou padrões de mobilidade.

No artigo [12], há uma análise geral do panorama de gestão de mobilidade em redes veiculares. Os autores destacam as diferenças entre protocolos de gestão de mobilidade para redes V2V e V2I, apontando que existe uma preocupação especial com roteamento nas redes V2V, enquanto nas redes V2I o foco é a utilização de protocolos para a Internet. Apesar de o artigo conter várias referências interessantes sobre as diferenças entre redes V2V e V2I, não há uma análise específica de algoritmos de *handoff* para redes veiculares.

O trabalho encontrado que mais se assemelha à proposta deste estudo é citado em [13]. Os autores se propõem a estudar a diferença entre um cenário normal de *handoff* e um cenário otimizado, no qual a rede contaria com um ponto de gestão de mobilidade (MAP - *Mobility Anchor Point*). São simulados cenários com os veículos a diferentes velocidades e verificadas as variações de latência do *handoff* e número de pacotes gerados no processo. As simulações foram feitas com o Omnet++, mas não são dados mais detalhes sobre a simulação (como número de veículos, detalhes das rotas simuladas, etc.).

5 RESUMOS DOS ARTIGOS

Descritas todas as particularidades do processo de *handoff* em redes veiculares e os motivos pelos quais seu estudo é interessante, foram escolhidos três artigos para uma análise mais detalhada. Cada um dos artigos propõe um algoritmo diferente para a realização do processo de *handoff* em uma rede veicular.

Para o presente trabalho (análise de algoritmos de *handoff* em redes veiculares), existe uma enorme gama de opções de algoritmos a escolher. À popularização dos conceitos de Internet das Coisas e Cidades Inteligentes seguiu-se o aumento de estudos sobre assuntos correlatos, como a relação entre trânsito e conectividade móvel, e por isso existem muitas opções de artigos publicados nos últimos anos sobre este assunto. Dado este vasto conjunto de possibilidades, justifica-se a escolha dos artigos analisados neste trabalho com os seguintes motivos:

- 1) Data de publicação: os 3 artigos analisados foram publicados nos últimos 4 anos, o que traz a possibilidade de contribuir com análises novas para a literatura
- 2) Diferença de critérios: cada um dos 3 algoritmos leva diferentes critérios em consideração para o processo de decisão. Essa diferença permite realizar uma análise de alto nível sobre a relevância de cada um dos critérios para o processo
- 3) Diferença geográfica: cada um dos algoritmos foi publicado por um conjunto de cientistas de países diferentes. Isso pode levar cada um deles a considerar critérios que podem mudar de país para país, como o custo envolvido na contratação de pacotes de Internet, os níveis de tráfego de automóveis nas estradas ou possíveis custos adicionais de componentes de hardware necessários para as arquiteturas propostas (como, por exemplo, GPS)
- 4) Complexidade e representação de dados: cada um dos algoritmos possui um nível de detalhe diferente em termos de número de fatores levados em consideração para a decisão e estrutura utilizada para a representação dos dados, utilizando de fórmulas simples a árvores de decisão complexas. Esta diferença permite a análise de possíveis impactos das estruturas de dados utilizadas sobre a eficiência dos algoritmos

Embora exista uma série de diferenças entre os algoritmos selecionados, todos eles têm elementos em comum:

- 1) Todos os algoritmos contam com a existência de uma infraestrutura tipo V2I (*Vehicle-to-Infrastructure*), tendo pontos fixos distribuídos ao longo do caminho dos veículos como principais pontos de acesso a receber as conexões.
- 2) A principal aplicação utilizada como exemplo para os algoritmos propostos é o *streaming* de vídeo. O presente trabalho busca, além de fazer uma análise do cenário original de cada algoritmo, criar um cenário genérico para que os algoritmos possam ser testados no mesmo cenário, e não com as aplicações que os autores de cada artigo propõem.

A seguir, temos o resumo de cada um dos artigos que serão analisados:

5.1 A vertical handoff method via self-selection decision tree for the Internet of Vehicles[2]: No primeiro artigo escolhido, os autores apresentam o conceito de Internet dos Veículos (em contrapartida à Internet das Coisas). Trata-se da proposta de atingir a ubiquidade da conectividade através da implantação da comunicação efetiva entre veículos. Embora o conceito não seja muito explorado, pode ser um tópico para futuras pesquisas. O cenário-base do artigo consiste de carros se locomovendo ao longo de rodovias, que contam obrigatoriamente com pontos de acesso (RSUs). O algoritmo apresentado utiliza uma árvore de decisão na qual as folhas representam probabilidades da ocorrência ou não ocorrência de *handoff* para um dado

destino está abaixo do necessário para migrar para esta rede. Se ela não estiver abaixo do necessário, não há *handoff*. Se ela estiver, passamos para o próximo nó na árvore de decisão. Depois de analisar a força do sinal do ponto de acesso atual e a taxa de erro de bits do ponto de destino, analisa-se a probabilidade de bloqueio do ponto de destino, que é definida baseada no número de canais disponíveis. Caso esta probabilidade esteja acima do *threshold* definido, não há *handoff*. Do contrário, existe uma alta probabilidade de que a conexão seja aceita, então passamos para o próximo nó na árvore. O próximo passo é decidir qual o tipo de conexão que se deseja. O algoritmo define 4 opções nesta etapa da decisão. Seguimos o nó de acordo com o que queremos priorizar na conexão: continuidade, largura de banda, custo ou orientação a serviço. No caso da escolha por priorização de continuidade, utiliza-se uma fórmula para definir um valor de mobilidade para os nós, que leva em consideração dados coletados anteriormente sobre a distância entre o veículo e os pontos de acesso pelos quais o mesmo passou

$$\Delta D_d = \sum_{i=1}^{M-1} \left[10^{\frac{K_1 + u(x) - \text{RSS}_{i+1}}{K_2}} - 10^{\frac{K_1 + u(x) - \text{RSS}_i}{K_2}} \right],$$

$$i = 1, 2, \dots, M - 1$$

Figura 2: Fórmula da proximidade dos veículos e pontos de acesso. Extraída do artigo [2]

Caso este valor seja menor que zero, significa que o veículo ainda está próximo o suficiente do ponto de acesso atual para que a conexão com o mesmo continue, e que, portanto, não deve haver o *handoff*. Caso seja menor ou igual a zero, significa que o veículo não está mais próximo o suficiente do ponto atual, e que portanto deve-se realizar o *handoff*.

No caso da escolha por largura de banda, calcula-se a taxa de transmissão máxima da rede atual e da rede de destino através do Teorema de Shannon, que estabelece um limite teórico máximo para a taxa de dados de um canal. Este teorema utiliza a relação sinal-ruído (FSNR) e a largura de banda de um canal para calcular sua taxa de transmissão máxima. Esta taxa máxima é dada por:

$$C = B \times \log_2 \left(1 + \frac{s}{n} \right)$$

Figura 3 - Fórmula do Teorema de Shannon. Extraída do artigo [2].

Onde C é a taxa teórica máxima, B é a largura de banda do canal e S/N é a relação sinal-ruído do canal.

Caso a taxa máxima de transmissão do ponto de acesso atual seja maior que o do ponto de destino, não há *handoff*. Caso seja menor ou igual, há *handoff*.

No caso da escolha por custo, verifica-se uma função de custo fixa atribuída a cada tecnologia do ponto de acesso. Caso o custo do ponto de destino seja maior do que o ponto atual, não há *handoff*, Caso seja menor ou igual, há *handoff*.

No caso da escolha por orientação a serviço, verifica-se qual tipo de serviço se escolheu priorizar: serviços de conversação (nos quais os pacotes contêm menos dados e a qualidade não é imprescindível) ou serviços de *streaming* (nos quais os pacotes contêm mais dados e é interessante que se obtenha uma qualidade melhor). Cada tipo de serviço é associado a uma sequência de prioridade entre as tecnologias de rede que são analisadas no artigo. No caso dos serviços conversacionais, a prioridade é: $3G > WiMax > WLAN$. No caso de serviços de *streaming*, a prioridade é $WLAN > WiMax > 3G$. Isto é devido ao custo monetário e capacidade de transmissão de cada tecnologia de rede.

Em ambos os casos, verifica-se se a rede de destino tem uma prioridade maior ou igual na sequência definida. Caso tiver, há *handoff*. Caso contrário, não há *handoff*. Existe a bifurcação na árvore entre os serviços apenas porque a prioridade de tecnologias é diferente para cada um.

Para o cenário de simulação, foram selecionadas as principais variáveis de interesse. Mais adiante, nas sessões de simulação e análise, será detalhado o método de simulação e quais parâmetros foram escolhidos para compor o código.

5.2 Predictive MobileIP Handover for Vehicular Networks [14]: no segundo artigo escolhido, os autores destacam que, caso fosse possível prever quando as desconexões ocorreriam, seria possível aumentar o desempenho da rede. O artigo se baseia na utilização do mobile IP como tecnologia de transporte de pacotes em redes veiculares. Segundo os autores, o processo de handover no mobile IP possui duas principais etapas: descoberta e registro. Na fase de descoberta, um nó da rede que se afasta de sua área de cobertura procura por um novo ponto de acesso ao qual pode se conectar. Uma vez encontrado, inicia-se a fase de registro, na qual um nó comunica ao seu *Home Agent* (ponto de acesso responsável por gerenciar as conexões) qual é o seu novo ponto de acesso. Isso possibilita o encaminhamento dos pacotes para seu novo endereço.

Este processo levaria, em média, 300ms (tempo durante o qual o nó da rede estaria impossibilitado de receber novos pacotes). Segundo os autores, a execução prévia dos processos de descoberta e registro poderiam levar a uma economia de tempo equivalente a 300ms, tornando a rede mais eficiente. A ideia do algoritmo

proposto no artigo é a execução prévia destas etapas do handover, a fim de aumentar a eficiência temporal da rede.

A principal ideia por trás do algoritmo é a previsão de movimento dos veículos da rede. Por meio da análise probabilística do sinal de GPS de cada veículo, seria possível prever onde eles estariam em um dado instante de tempo futuro, e qual seria o ponto de acesso ao qual se conectariam.

A previsão da posição futura dos veículos seria dada por uma equação cinemática simples:

$$x_t = x_{t-1} + v * t + 0.5 * a * t^2$$

Figura 5 - Equação da posição dos veículos. Extraída do artigo [14].

Sendo que os dados necessários são todos coletados por sensores e GPS no veículo.

Antes de utilizar esta fórmula para prever qual seria o ponto de acesso ao qual o veículo se conectaria, é definido um modelo probabilístico para a utilização dos dados. Primeiramente, são definidas as probabilidades iniciais de cada ponto de acesso da rede receber uma nova conexão de um veículo que entra em sua área de cobertura. Essa probabilidade inicial vem de dados históricos sobre a rede, sendo que os pontos de acesso que receberam mais conexões anteriormente recebem uma probabilidade inicial maior de aceitar novas conexões.

$$X = \begin{pmatrix} p(x_1) & p(x_2) & \dots & p(x_N) \end{pmatrix}$$

Figura 6 - Matriz (1 x N) de probabilidades iniciais. Cada elemento da matriz representa a probabilidade de o ponto de acesso x_i ser escolhido como próximo ponto de acesso. Extraída do artigo [14].

É definida também uma matriz de probabilidades das observações dos veículos em relação ao ponto de acesso ao qual podem se conectar, ou seja, para cada série de dados coletados sobre os veículos em um dado instante, o conteúdo da matriz indica a probabilidade de que este veículo se conecte com cada um dos pontos de acesso da rede.

$$B = \begin{pmatrix} p(o_1|x_1) & p(o_1|x_2) & \dots & p(o_1|x_N) \\ p(o_2|x_1) & p(o_2|x_2) & \dots & p(o_2|x_N) \\ \vdots & \vdots & \ddots & \vdots \\ p(o_M|x_1) & p(o_M|x_2) & \dots & p(o_M|x_N) \end{pmatrix}$$

Figura 7 - Matriz (N x N) de probabilidade de escolha baseada em uma observação. Extraída do artigo [14]

Por último, é definida uma outra matriz, que indica a probabilidade de erro de cada uma das previsões de handover. Ou seja, para cada um dos pares possíveis de pontos de acesso da rede, qual é a probabilidade de o primeiro par ter sido previsto como o futuro ponto de acesso quando na verdade o segundo foi o ponto ao qual o veículo se conectou.

$$T = \begin{pmatrix} p(x_1|x_1) & p(x_1|x_2) & \dots & p(x_1|x_N) \\ p(x_2|x_1) & p(x_2|x_2) & \dots & p(x_2|x_N) \\ \vdots & \vdots & \ddots & \vdots \\ p(x_N|x_1) & p(x_N|x_2) & \dots & p(x_N|x_N) \end{pmatrix}$$

Figura 8 - Matriz (N x N) de probabilidade de erro de escolha. Extraída do artigo [14]

É interessante notar que, nesta matriz de probabilidade de erros, a diagonal principal representa a probabilidade de que um ponto de acesso tenha sido previsto corretamente.

À medida que novas observações chegam até o sistema de predição, as probabilidades da matriz X (matriz de probabilidades individuais de escolha de cada ponto de acesso) são atualizadas de acordo com a seguinte fórmula:

$$f_{O:t} = c_t^{-1} * f_{O:t-1} * T * O_t$$

Figura 9 - Fórmula de atualização da matriz de probabilidades. Extraída do artigo [14].

Uma vez calculada a nova probabilidade para cada ponto de acesso, os valores são comparados a um *threshold* pré-definido. Caso um ponto tenha sua probabilidade acima deste valor, ele é escolhido como novo ponto de acesso

$$p_{th} \leq \text{Min}(P(p_d|p_{th}) \cup P(p_w|p_{th}))$$

Figura 10 - Regra para a escolha do novo ponto de acesso. Extraída do artigo [14].

A previsão de futuros pontos de acesso é feita com a utilização de um Modelo Oculto de Markov (HMM - Hidden Markov Model). O Modelo Oculto de Markov é um modelo estatístico que busca prever o estado futuro de um sistema sem que todos os estados sejam conhecidos. Nas Cadeias de Markov (modelo que origina os Modelos Ocultos de Markov), existe uma série de estados discretos possíveis e uma série de ações que levam a diferentes estados. No Modelo Oculto de Markov, os estados podem não ser conhecidos. A estimação é feita a partir da atribuição de probabilidades aos estados, baseadas em observações passadas sobre estes estados. Desta forma, podemos dizer que os pontos de acesso são estados no HMM e o histórico do ponto de acesso é a observação utilizada para a atribuição de probabilidades.

Os dados do GPS são processados por um filtro de Kalman. Também conhecido como estimação linear quadrática, o filtro de Kalman é um modelo matemático que se baseia em valores de certas variáveis obtidos ao longo do tempo para prever os valores futuros destas variáveis. Desta forma, é possível estimar a posição futura de um veículo através de dados vindos de um GPS.

No modelo proposto no artigo, o processamento do HMM e a descoberta de novos vizinhos é feito todo pelos próprios pontos de acesso, enquanto os dispositivos dos veículos são responsáveis somente pelo processamento do filtro de Kalman. Esta abordagem evitaria que dispositivos com pequeno poder de processamento como as OBU's dos veículos ficassem responsáveis por lidar com detalhes da rede que representariam, potencialmente, um alto custo computacional.

O HMM dos pontos de acesso é alimentado da seguinte maneira: quando os veículos estão sob a cobertura de um dado ponto de acesso, eles trocam pacotes normalmente com este ponto. Uma vez que eles saiam desta área de cobertura, eles enviam os dados do seu novo ponto de acesso ao ponto anterior. Dessa forma, os APs podem descobrir quais são os seus vizinhos. Cada ponto mantém uma lista de pontos vizinhos. Caso um ponto receba um pacote com um IP de um novo ponto, foi descoberto um novo vizinho. Este novo vizinho então recebe uma nova probabilidade inicial. Caso o pacote recebido contenha um IP que já esteja na lista de vizinhos, sua probabilidade é atualizada de acordo.

A atualização das probabilidades de cada vizinho é feita levando-se em consideração o número de veículos que enviaram pacotes contendo o seu IP e a performance do veículo enquanto esteve sob a cobertura deste ponto. A medida de

performance tem como principais variáveis a vazão do ponto de acesso e o tempo de permanência (dwell time) do veículo sob a cobertura do ponto.

$$w = \frac{th_{KB/s} \times t_d}{z \times (k + J)}$$

Figura 11 - Fórmula de atribuição de peso para a performance de um dado ponto de acesso. Extraída do artigo [14].

Todo o processo descrito anteriormente diz respeito à etapa de descoberta de novos pontos. A etapa do registro ocorre da seguinte maneira: quando o ponto de acesso atual conclui qual será o ponto ao qual um veículo irá se conectar, o veículo envia um pacote com um pedido de registro ao ponto previsto. Este pacote é enviado através do ponto atual (já que não seria possível enviar nada diretamente ao ponto previsto, pois o veículo ainda não estaria sob sua cobertura). Ao receber o pedido de registro, o ponto previsto envia ao Home Agent um pacote de registro, para que ele saiba que o veículo irá se conectar a ele. Se a comunicação for bem sucedida, o Home Agent responde ao ponto previsto com um OK, mas sem quebrar a conexão com o ponto atual do veículo, a fim de minimizar a perda de pacotes no processo de transição. Quando um veículo começa a perder a conexão com o ponto de acesso atual, ele envia pacotes em broadcast indicando que deseja se conectar a um ponto de acesso específico. Os pontos de acesso que receberem esta transmissão em broadcast verificarão se o veículo em questão consta em uma lista que eles mantêm, que conta com os dispositivos registrados no processo de comunicação com o Home Agent. Caso o veículo esteja na lista, o ponto de acesso responde com um pacote de completude de registro e envia ao Home Agent a informação de que o veículo está conectado. Neste momento, o HA encerra a conexão com o ponto de acesso anterior, completando o processo de registro. A figura a seguir detalha o processo de registro e a comunicação entre as entidades envolvidas:

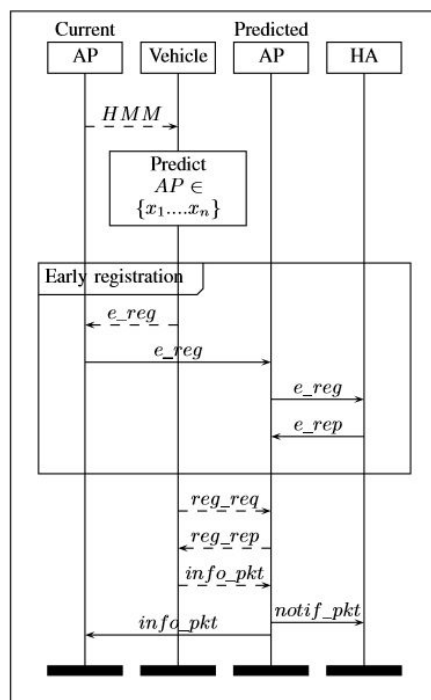


Figura 12 - Esquema de comunicação entre as partes envolvidas. Extraída do artigo [14].

Quando a previsão é feita corretamente, o overhead decorrente das etapas de descoberta e registro pode ser evitado. Entretanto, existem casos em que a previsão não estará correta, já que todo o algoritmo é baseado em modelos probabilísticos. Para estes casos, deve haver um protocolo de erro a ser seguido. Para este algoritmo, um erro não pode ser detectado até que um veículo comece a procurar por um novo ponto de acesso.

5.3 An intelligent vertical handover scheme for audio and video streaming in heterogeneous vehicular networks [15]: no terceiro artigo escolhido, os autores destacam a importância da qualidade de serviço (QoS - Quality of Service) e da continuidade das conexões no processo de streaming de áudio e vídeo, reconhecendo que é algo muito difícil de se atingir em redes veiculares e propondo um algoritmo com o foco na resolução desse problema em particular. Os autores propõem o que chamam de INS (Intelligent Network Selection - Seleção Inteligente de Rede), um algoritmo que se baseia na maximização de uma função de pontuação para a escolha do melhor candidato a ponto de conexão para um veículo, sendo que o principal caso de teste é o handover vertical entre UMTS e WLAN. As simulações dos autores foram realizadas utilizando o OMNET++ como simulador, e, segundo eles, o algoritmo é capaz de reduzir o número de desconexões e handovers desnecessários.

O algoritmo funciona da seguinte forma: três parâmetros principais são considerados para a escolha de um novo ponto de acesso para um veículo. São eles: FSNR (Faded Signal-To-Noise ratio), Residual Channel Capacity e Connection Lifetime. O primeiro é basicamente a razão entre a capacidade de um ponto de acesso dividido pela capacidade do ruído neste ponto. O FSNR de um ponto pode ser afetado por condições climáticas como chuva, vento e temperaturas extremas. O segundo (Residual Channel Capacity) é representado pela largura de banda disponível para o acesso do veículo em um dado ponto. O terceiro (Connection Lifetime) representa quanto tempo um veículo poderá permanecer sob a cobertura de um dado ponto de acesso. No caso no handover entre UMTS e WLAN, o tempo de conexão de um veículo sobre a cobertura de um ponto UMTS é certamente superior ao tempo de conexão do mesmo veículo sobre um ponto WLAN, pois a área de cobertura do UMTS é maior.

Então, em resumo, os principais parâmetros considerados são a razão entre capacidade de um ponto e ruído presente neste ponto (FSNR), a largura de banda disponível (RCC) e o tempo de conexão máximo de um veículo (CL). Os autores associam a verificação do FSNR e RCC à qualidade de serviço obtida, enquanto a verificação do CL é associada à diminuição de handovers desnecessários. Estas três métricas serão maximizadas na função de pontos proposta (ou seja, quanto maiores os valores de FSNR, RCC e CL, melhor a pontuação na função). A função de pontuação proposta é a seguinte:

$$f(FSNR_i, R_{(cc)i}, t_{(connection)i}) = X \times FSNR_i^{\sigma_1} \times R_{cc_i}^{\sigma_2} \times t_{connection_i}^{\sigma_3} + Y_{max}$$

Figura 13 - Fórmula de pontuação de um ponto de acesso. Extraída do artigo [15].

Na qual σ_1 , σ_2 e σ_3 representam os pesos definidos para cada um dos parâmetros, $t_{(connection)i}$ representa o CL e X representa o balanceamento dos demais parâmetros em relação ao valor máximo da pontuação dos pontos de acesso. Sabemos que o valor máximo da função será encontrado quando ela for igualada a zero. Logo, o valor de X pode ser obtido com a seguinte fórmula:

$$X = \frac{-Y_{max}}{FSNR_{max}^{\sigma_1} \times R_{(cc)max}^{\sigma_2} \times t_{(connection)max}^{\sigma_3}}$$

Figura 14 - Fórmula final de atribuição de peso médio a um ponto de acesso. Extraída do artigo [15].

O algoritmo de escolha de pontos de acesso funciona da seguinte maneira: Primeiramente, assume-se que todos os veículos dispõem de pelo menos duas interfaces de rede: uma WLAN e uma UMTS. Cada veículo está ciente de qual é seu ponto de acesso atual e quais são os valores para cada um dos parâmetros definidos na função de custo. O veículo calcula então o valor de sua função de custo atual. A partir do momento em que um veículo começa a se mover e ficar sob a cobertura de pontos diferentes, o algoritmo verifica qual é o modo atual de sua interface. Existem dois modos possíveis: *busy* e *idle*. O busy mode indica que existem aplicações sendo executadas no momento, enquanto o idle indica que o veículo não está rodando nenhuma aplicação. Caso ele esteja em idle mode, o algoritmo escolhe apenas o próximo ponto de acesso com a maior pontuação da função 1, sem considerar outros aspectos. Isso evita uma carga de trabalho desnecessária, já que nenhuma aplicação está sendo executada. Após o handover, o ponto de acesso e valores da função de pontuação atual são atualizados.

Caso o veículo não esteja em idle mode, é possível que aplicações estejam sendo executadas. Como o algoritmo foca em aplicações de tempo real (como o streaming de áudio e vídeo), os autores mencionam a importância da continuidade da sessão em vez da qualidade do sinal ou outras características. Logo, a prioridade nestes casos é evitar handovers desnecessários na rede. Por outro lado, o busy mode da interface não indica necessariamente que a aplicação executada é de streaming de voz ou vídeo, sendo que no caso do streaming de vídeo a qualidade do serviço também desempenha um papel crucial sobre a execução da aplicação, então é necessário se preocupar com a qualidade de serviço também. A fim de resolver o problema, o algoritmo identifica qual tipo de aplicação está sendo executada para saber quais parâmetros priorizar. Para cada tipo principal de aplicação é definido um número que indica a largura de banda mínima para a continuidade do serviço. Se a aplicação em execução for um aplicação de tempo real, o algoritmo compara a largura de banda do ponto de acesso atual com o mínimo definido para a aplicação. Caso sua largura de banda esteja acima desse mínimo, o algoritmo considera que o ponto de acesso atual ainda é capaz de manter uma boa qualidade de serviço, e evita manter a conexão do veículo com o ponto atual até que ela não seja mais possível (pelo fato de o veículo ter saído da cobertura ou por alguma queda da largura de banda). Porém, se a aplicação não pertencer ao grupo de aplicações de tempo real, o aumento da qualidade de serviço é priorizado. Logo, o algoritmo escolherá o ponto com a maior largura de banda em vez de se contentar com o mínimo definido. Neste caso, é verificado qual é a tecnologia do ponto de destino. Caso seja UMTS, o handoff é iniciado. Caso seja WLAN, é necessário verificar também qual a velocidade do veículo. Se a velocidade é maior que 50km/h, o handoff é evitado pelo algoritmo. Isso é necessário por causa da cobertura menor oferecida pelos pontos WLAN. Caso a velocidade do veículo seja

muito alta, aumenta a probabilidade de quebra do canal depois de um tempo muito curto, o que resulta em handoffs desnecessários e queda na qualidade de serviço. Se a velocidade do veículo for menor que 50km/h, o handoff é iniciado. Os autores falham em explicar o porquê do número exato 50km/h.

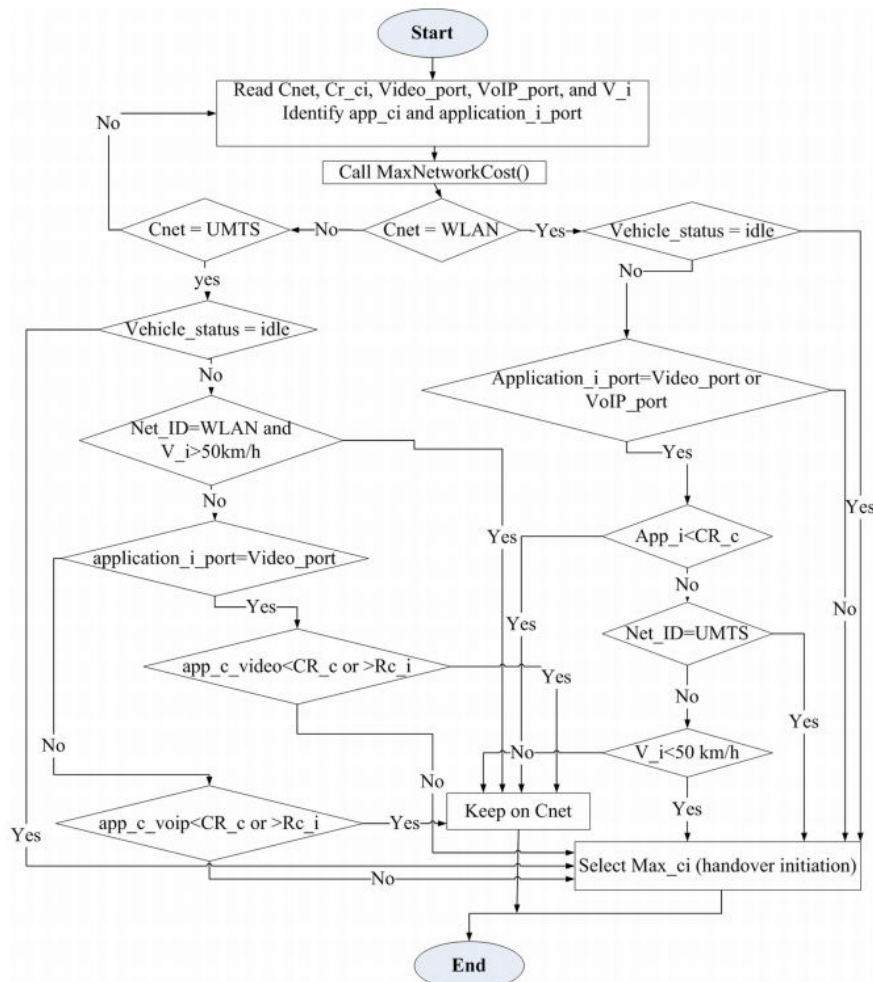


Figura 15 - Fluxo de decisão para a execução de um handoff. Extraída do artigo [15].

Com a descrição completa do algoritmo, verificamos que, no total, o INS considera 6 parâmetros: o três descritos no cálculo da função de pontuação, a aplicação em execução, a velocidade do veículo e a tecnologia do ponto de acesso de destino. Entretanto, veremos mais adiante na sessão de simulação que este fluxo de decisão não é pertinente ao código do trabalho, visto que o principal interesse é nas situações em que o *handoff* ocorre de fato. O que será simulado na prática é a função de cálculo de pontuação para os pontos de acesso, cujas principais fórmulas foram apresentadas nas figuras 13 e 14.

6 SIMULAÇÃO

Cada um dos artigos analisados propõe um cenário diferente para a simulação de seus respectivos algoritmos. O terceiro artigo, por exemplo, é modelado exclusivamente dentro de um cenário no qual a aplicação em execução nos dispositivos móveis seria do tipo VoIP ou streaming de vídeo. No primeiro artigo, são feitas referências a tipos de serviços “com sessão” ou “sem sessão” para a decisão do handoff. Com combinações diferentes entre diversas tecnologias, os artigos mencionam o *handoff* entre UMTS e WLAN, UMTS e WiMAX, ou não há referência a tecnologias específicas (como é o caso do segundo artigo).

Com o objetivo de padronizar as simulações e colocar os algoritmos em um cenário mais comum, a simulação foi modelada de maneira simples, na qual veículos apenas geram pacotes de um tamanho fixo e transmitem através de pontos de acesso modelados ou como UMTS ou como WLAN. Em alguns dos casos, as capacidades específicas de UMTS e WLAN não são tão relevantes quanto a própria característica de diferença entre esses pontos, então características das tecnologias como velocidade e alcance foram simplificadas. Em outros casos, as características específicas das tecnologias são relevantes. Nestes casos os pontos de acesso são modelados de acordo com as capacidades reais de cada tecnologia. As sessões a seguir contém detalhes sobre a modelagem dos pontos de acesso e do cenário no qual a simulação ocorre.

6.1 Características do simulador

Foram testadas algumas ferramentas antes da escolha definitiva de qual seria o simulador. Foram analisados o Omnet++ (simulador de eventos), SUMO (simulador de tráfego) e Veins (*framework* que utiliza o Omnet++ e SUMO para a simulação de redes veiculares). A principal dificuldade encontrada no trabalho com estas ferramentas foi a dificuldade de modificação do código, uma vez que as variáveis e funções não tinham nomes muito significativos, tornando o código ilegível. Além disso, o conjunto dessas ferramentas é um tanto pesado em termos de memória, apresentando lentidão na execução em computadores com pouca memória RAM. O simulador escolhido para a modelagem e teste dos algoritmos foi o The ONE (Opportunistic Network Environment). Ele foi projetado para a avaliação de redes DTN (Delay Tolerant Networks), com suporte a diferentes protocolos de roteamento. Esta característica torna o The ONE um simulador conveniente para o teste de aplicações em redes veiculares que, apesar de não serem consideradas redes DTN, possuem características que as diferenciam de outros tipos de rede. Com o The ONE, usuários podem criar cenários com base em diferentes modelos de mobilidade e mapas reais, além de oferecer um *framework* para

a implementação de novos protocolos. O simulador oferece uma ferramenta de visualização interativa e geração de relatórios, facilitando a avaliação dos experimentos.

Os algoritmos foram modelados de acordo com os parâmetros descritos em seus respectivos resumos, escritos na linguagem Java e adicionados ao módulo do The ONE responsável por transferir a sessão de comunicação de um ponto de acesso para outro. Todas as modificações feitas foram no próprio código do simulador, e não em módulos adicionais. Este módulo identifica a perda do sinal de um ponto de acesso, desfaz as conexões com o ponto perdido e identifica os novos candidatos a ponto de acesso. A modelagem se baseou fortemente na descrição dos algoritmos do capítulo de resumo dos artigos (capítulo 5), mas alguns dos parâmetros e mecânicas de funcionamento foram alterados por motivos diversos, como será descrito mais adiante para cada um dos casos.

A modelagem de cenários de rede veicular no The ONE ocorre da seguinte maneira: existe um arquivo principal, chamado *default_settings.txt*, que contém todas as informações sobre o cenário. As principais informações de interesse são: informações sobre as interfaces de rede disponíveis (velocidade de transmissão, alcance do sinal, etc.), informações sobre os nós da rede (velocidade dos veículos, interfaces conectadas, quantidade de nós, etc.) e informações sobre geração de relatório. Também é possível mudar alguns atributos das mensagens enviadas na rede, o algoritmo de roteamento utilizado, o tipo de movimento que os nós seguem, o mapa no qual os veículos se movimentam e detalhes sobre a exibição de simulação na interface gráfica. Como estes últimos parâmetros não estão diretamente relacionados ao processo de *handoff*, os valores originais foram mantidos, com o objetivo de simplificar a simulação.

```
# Common settings for all groups
Group.movementModel = ShortestPathMapBasedMovement
Group.router = EpidemicRouter
Group.bufferSize = 5M
Group.waitTime = 0, 120
# All nodes have the bluetooth interface
Group.nrofInterfaces = 1
Group.interface1 = btInterface
# Walking speeds
# Message TTL of 300 minutes (5 hours)
Group.msgTtl = 300
Group.nrofHosts = 40

Group1.groupID = antenna
Group1.speed = 0, 0

# group2 specific settings
Group2.groupID = car
# cars can drive only on roads
Group2.okMaps = 1
# 10-50 km/h
Group2.speed = 2.7, 13.9
```

Figura 16 - Exemplo de configuração no arquivo *default_settings.txt*

A primeira modificação necessária para a execução da simulação foi a mudança da característica de comunicação V2V (Vehicle-to-Vehicle, tipo de comunicação descrita no capítulo 2), habilitada por padrão no simulador. Não existe um parâmetro específico que indique o tipo de comunicação V2V, mas nas simulações originais do The ONE, quando a troca de mensagens se inicia na rede todos os nós podem transferir dados para todos os outros. Para que o cenário se parecesse mais com os algoritmos simulados (comunicação V2I), foi necessário modificar o simulador para que não permitisse o envio de mensagens diretamente entre dois veículos. Outro problema seria que, da maneira como as configurações e o código estavam, as antenas teriam certa “autonomia” na decisão pelo *handoff*, de modo que se a interface do veículo não atendesse às mesmas condições avaliadas para as antenas, não ocorreria conexão. Na verdade, nos algoritmos do presente trabalho, são as antenas que são avaliadas como ponto de acesso de acordo com suas respectivas capacidades, e não os veículos, então foi definido que os parâmetros das interfaces dos carros nunca seriam menores do que os *thresholds* definidos nos algoritmos, para que as antenas não “decidissem” que não vale a pena conectar-se a um certo veículo.

Foi utilizado o valor padrão para o tempo de simulação (tempo durante o qual a aplicação é executada e durante o qual os dados são coletados). Originalmente no The ONE, este valor é de 43200, representando 43200 segundos, ou seja, 12 horas.

Um problema enfrentado no início das simulações foi a falta de suporte do simulador a alguns parâmetros necessários para os algoritmos. O primeiro algoritmo, por exemplo, utiliza a taxa de erro de bits e a probabilidade de rejeição de conexões de cada antena para a decisão de realização do *handoff*. Entretanto, não é possível obter estes dados diretamente do código do The ONE. A solução foi a criação manual dos parâmetros faltantes. Foram criados diversos grupos com valores variados para estes parâmetros, simulando a existência de diversas antenas com valores variados para os parâmetros que não podem ser obtidos. Para que estes parâmetros artificiais (ou seja, que não interferem nos processos de rede do simulador) tivessem algum efeito sobre as capacidades reais dos pontos de acesso, foi definido que os valores destes parâmetros intefeririam probabilisticamente sobre as funções de conexão e envio de mensagens do simulador. Foram definidos valores mínimos e máximos para cada parâmetro e valores considerados “baixos demais” ou “altos demais” fariam com que as funções de conexão e/ou envio de mensagens não fosse executada, com probabilidade correspondente a seus valores. Por exemplo, o parâmetro de sinal recebido (RSS) não existe no simulador, mas é utilizado por um dos algoritmos. É criado este parâmetro e são atribuídos valores a ele no arquivo *default_settings.txt*. Porém não existe garantia de que um baixo sinal recebido levaria mais tarde a uma dificuldade ou perda de conexão, já que é apenas um número definido. Digamos que o máximo sinal recebido é 100, mas o sinal definido foi 30. Em código, há a geração de um número aleatório entre

1 e 100. Caso este número seja menor que 30, a função de conexão é executada. Do contrário, ela não é. Isto representa a chance de um ponto de acesso de baixo sinal recebido não apresentar um desempenho satisfatório quanto à qualidade da conexão, e de pontos de alto sinal recebido apresentarem um bom desempenho (já que o número gerado estará entre 1 e 100, um ponto de acesso de RSS 100 com certeza executará a função de conexão). O mesmo processo foi repetido para todos os parâmetros que não puderam ser obtidos através do código nativo do simulador. Assim é possível criar uma representação mais fiel do cenário de cada algoritmo. Na figura a seguir podemos verificar como esta geração de números aleatório foi feita em código, dentro do simulador (linguagem Java). A função *connect(i)* é a função do próprio simulador que faz a conexão entre a interface na qual o código está inserido e a interface *i*.

```
int max = 100;
int min = 1;
int randomNum = 0;
Random rand = new Random();
for (NetworkInterface i : interfaces) {
    randomNum = rand.nextInt((max - min) + 1) + min;
    if (randomNum <= i.rss) {
        connect(i);
    }
}
```

Figura 17 - Geração de números aleatórios para executar a função *connect(i)* com probabilidade correspondente aos parâmetros definidos

6.2 Algoritmo 1:

Na simulação do primeiro algoritmo, foi criada uma árvore de decisão considerando os seguintes parâmetros: sinal recebido de uma antena, taxa de erro de bits, prioridade da conexão (continuidade de conexão, custo, largura de banda ou tipo de serviço) e tipo de tecnologia de rede (ou seja, verificar se uma antena é UMTS ou WLAN). Embora na descrição do algoritmo colocada anteriormente neste artigo sejam mencionados outros parâmetros, foi considerado que eles seriam difíceis de simular, ou que sua simulação seria desnecessária, dado o caráter artificial do simulador (por exemplo, não existem ruídos e as capacidades de processamento das antenas são muito grandes) e esta dificuldade atrapalharia nos resultados. São eles: probabilidade de bloqueio de uma conexão, *Vehicle Movement Trend* (verificar se um veículo está “perto o suficiente” de um ponto de acesso) e taxa de transmissão máxima de uma antena, calculada através da fórmula do Teorema de Shannon.

Mesmo com a simplificação, o simulador ainda não suporta alguns dos parâmetros importantes para o processo, então foi necessário criá-los manualmente em código, como descrito na seção 6.1. Os valores foram definidos no arquivo *default_settings.txt*. Para este algoritmo, criamos 8 grupos diferentes: 6 grupos de antenas e 2 grupos de veículos. Os grupos de antena possuem valores diferentes para cada um dos parâmetros levados em consideração. O primeiro grupo simula pontos de acesso “perfeitos”, enquanto os outros simulam pontos de acesso a serem evitados, visto que possuem valores abaixo de um *threshold* definido para pelo menos um dos parâmetros a avaliar. Quatro dos grupos foram modelados como UMTS e dois como WLAN. Visto que um dos objetivos do trabalho é generalizar as aplicações dos algoritmos, foram ignorados alguns parâmetros específicos de UMTS e WLAN, como alcance dos pontos e velocidade. A modelagem serve apenas como base para a diferenciação de uma escolha entre dois tipos de ponto de acesso por parte do algoritmo. Os grupos de veículos possuem valores diferentes para a velocidade: um está a 50km/h e outro a 100km/h. A posição inicial dos nós é definida aleatoriamente no mapa. O objetivo é verificar que o algoritmo realmente evita conexões a estes pontos considerados ruins.

Na parte do código responsável por decidir se um nó se conecta a outro ou não, é colocada a árvore de decisão, e somente se os critérios são atendidos é que a conexão ocorre. Caso o algoritmo tenha verificado todos os pontos e nenhum atenda exatamente aos parâmetros necessários, a conexão acaba ocorrendo de qualquer forma, pois considera-se que é melhor uma conexão “falha” do que nenhuma. Esta característica não é prevista no algoritmo original, mas foi adicionada para que o processo fosse mais parecido com o dos outros algoritmos, que acabam por realizar uma conexão sempre que o algoritmo é executado.

Foram executadas 4 simulações diferentes para este algoritmo. Estas simulações correspondem às ramificações da árvore descrita. A primeira simulação leva em consideração que a prioridade seria a largura de banda, a segunda o custo da rede e a terceira o serviço, sendo que este foi definido como “serviço de sessão”. Neste caso, o algoritmo deve optar por pontos de acesso do tipo UMTS. Na quarta simulação, foi definido que o tipo do serviço seria “exclusivo”, o que faz com que o algoritmo priorize os pontos de acesso que foram modelados como WLAN.

6.3 Algoritmo 2:

O segundo algoritmo é o mais complexo em termos de comunicação entre diferentes entidades. Como vimos na descrição da simulação do primeiro algoritmo (e como veremos no terceiro) as principais mudanças necessárias eram apenas a inclusão de um processo diferente de decisão na parte do simulador em que as

conexões ou desconexões ocorrem. No segundo algoritmo, temos outras partes envolvidas (o *Home Agent*, as torres às quais o ponto de acesso esteve conectado anteriormente e os pontos de acesso previstos pelo algoritmo), além da utilização de padrões de mobilidade e histórico de conexões de cada ponto de acesso..

A modelagem dos agentes externos foi feita através da criação de uma nova classe no simulador. No código original do simulador, dois veículos se conectam apenas através da chamada de uma função (função *connect()*, implementada em cada classe de interface). No caso do algoritmo 2, devemos inserir uma entidade extra para representar o Home Agent, que tem como função “liberar” as conexões entre pontos. Os Home Agents são classes públicas às quais todos os outros nós possuem acesso. A simulação funciona de maneira similar ao fluxo representado na figura 12, na qual o ponto de acesso atual de um veículo prevê qual será seu próximo ponto de acesso e envia esta informação ao veículo e ao ponto previsto. O ponto previsto, por sua vez, comunica ao Home Agent que um novo veículo irá se conectar a ele e inicia a fase do registro do novo veículo anteriormente à sua chegada na área de cobertura. É justamente a execução prévia dessa fase de registro que dá alguma vantagem ao algoritmo. Como a principal vantagem seria temporal, analisaremos os delays das mensagens de forma mais crítica que outros parâmetros.

O processo de descoberta é facilitado pelas possibilidades do simulador. Em vez de implementar todo o esquema de previsão a partir de filtros de Kalman e Modelos Ocultos de Markov, foi possível utilizar os modelos de movimentação do próprio simulador para descobrir qual seria o próximo ponto ao qual um veículo se conectaria. Um possível problema deste método é que estaríamos eliminando o overhead resultante da manipulação estatística de matrizes, não permitindo a análise crítica do desempenho do algoritmo. Entretanto, como neste caso o maior interesse é a verificação da variação do *delay*, considera-se que o *overhead* pode não ser tão importante na análise do algoritmo 2. Além disso, nenhum dos três algoritmos é analisado quanto ao seu *overhead* no presente trabalho, então não precisaríamos desta métrica para comparação.

Visto que o algoritmo 2 não traz um fluxo de execução variável com diferentes parâmetros ou pesos diferentes para os parâmetros, as simulações foram feitas diversas vezes para o mesmo código (sem alterações entre elas). A execução de mais de uma simulação para o mesmo código tem como objetivo a eliminação de um possível viés resultante da aleatoriedade do posicionamento dos nós no mapa. Foram executadas quatro simulações, mesmo número de simulações do algoritmo 1, a fim de obter uma amostra uniforme para a média.

6.4 Algoritmo 3:

O terceiro algoritmo é, de algumas formas, similar ao primeiro. Ambos modelam o processo de decisão como uma árvore, e levam em conta parâmetros como o sinal recebido, a tecnologia da rede de destino e a aplicação em execução no momento do *handoff*. O algoritmo contém, como colocado na figura 15, uma árvore de decisão para a execução (ou não) do *handoff*. Dentro desta árvore estaria a função de cálculo de custo para cada ponto de acesso. Como o principal interesse deste trabalho é o processo de *handoff* em si, foi simplificada a árvore de decisão de ocorrência do *handoff*, de modo que ele sempre ocorrerá, independente dos parâmetros desta árvore. A parte principal da simulação, então, é o cálculo da função de custo, a fim de decidir qual dos pontos de acesso disponíveis é o melhor para a futura conexão.

A função de cálculo de custo proposta utiliza três principais parâmetros: o FSNR (*Faded Signal to Noise Ratio*), a capacidade do canal (RCC - *Residual Channel Capacity*) e o tempo de vida da conexão (CL - *Connection Lifetime*). Uma vez calculados cada um desses parâmetros, multiplica-se todos eles, obtendo o valor final da função de custo para aquele ponto de acesso. O objetivo é maximizar esta função, então será escolhido o ponto de acesso com o maior valor para a função de custo.

Da mesma maneira que os algoritmos anteriores, foi necessário criar os parâmetros em código e fazer as devidas modificações para que passassem a fazer alguma diferença no processo, visto que o simulador não suporta características como o FSNR ou a capacidade de cada canal. Foram criados, então, diferentes grupos de ponto de acesso com diferentes valores para os parâmetros. O número de grupos é o mesmo da simulação do algoritmo 1: 8 grupos, sendo 6 de antenas e 2 de veículos. Os grupos variam em valores de FSNR, RCC e tempo de vida de conexão. Para este artigo, a tecnologia do ponto de acesso é importante no fluxo de decisão para a ocorrência ou não do *handoff*, mas não é relevante no momento em que o processo ocorre. Visto que o fluxo de decisão foi abstraído, a modelagem WLAN/UMTS não é importante no momento do cálculo da função de custo, mas os pontos de acesso foram modelados de maneira a refletirem as capacidades dessas tecnologias no arquivo *default_settings.txt*.

No artigo 3, os autores apresentam a fórmula de cálculo de custo de cada ponto de acesso, como colocado na figura 13. O algoritmo faz o cálculo dessa função de custo baseado nos valores apresentados no artigo. Os autores colocam um exemplo de vetor de pesos com os valores 0.9, 0.2 e 0.05, atribuindo estes pesos a FSNR, RCC e CL, respectivamente. Como o algoritmo não apresenta diferentes caminhos a seguir (como é o caso do algoritmo 1), foi utilizado apenas um conjunto de dados para a análise, obtido a partir de quatro execuções diferentes do mesmo algoritmo com as mesmas fórmulas e valores, como indicado pelos autores do artigo. Como nas simulações dos algoritmos anteriores, as quatro execuções diferentes servem para

diminuir um possível viés da aleatoriedade do posicionamento dos pontos do mapa e para uniformizar a média dos conjuntos de dados.

7 ANÁLISE

Primeiramente, é feita uma análise detalhada da simulação de cada algoritmo. Após a análise individual, comparamos os resultados entre si para identificar vantagens e desvantagens de cada algoritmo.

Para as análises, foram considerados como parâmetros mais relevantes para a comparação entre os algoritmos o tempo e a taxa de entrega das mensagens e o número de conexões que ocorreram entre os nós da rede durante a simulação.

7.1 Algoritmo 1:

Foram executadas quatro simulações diferentes para o algoritmo 1. Visto que a árvore de decisão do algoritmo tem diversas possibilidades, foram escolhidas as quatro principais, e cada uma delas foi simulada com um parâmetro diferente. Na primeira simulação, escolhemos priorizar a largura de banda na parte da árvore que decide qual parâmetro principal da rede será levado em consideração. Na segunda, priorizamos o custo da rede. Na terceira, priorizamos o tipo de serviço e consideramos que a aplicação em execução é do tipo sessão de dados (ou seja, há preferência por redes UMTS). E, finalmente, na quarta, priorizamos também o tipo de serviço, mas consideramos que a aplicação envolve streaming de dados (ou seja, há preferência para redes WLAN). Para os quatro parâmetros de comparação, obtivemos os seguintes resultados.

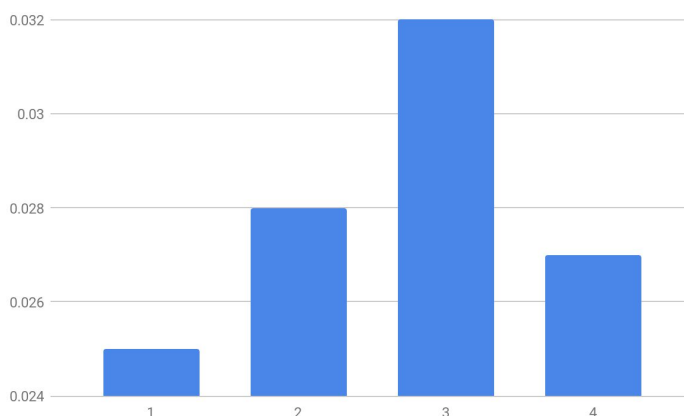


Gráfico 1: tempo médio de entrega (em segundos) para as simulações do algoritmo 1.

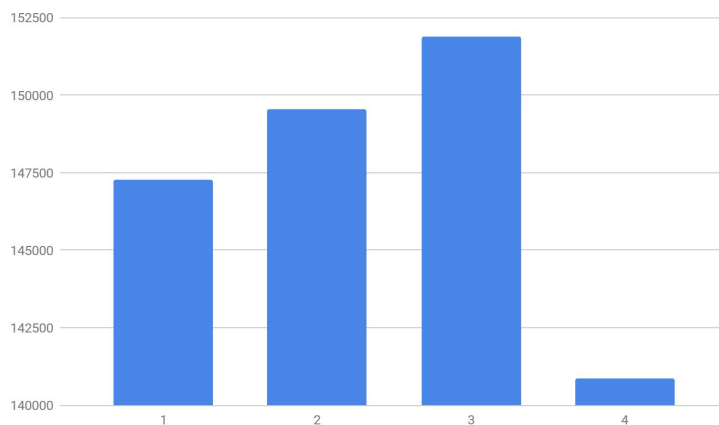


Gráfico 2: número de conexões para as simulações do algoritmo 1

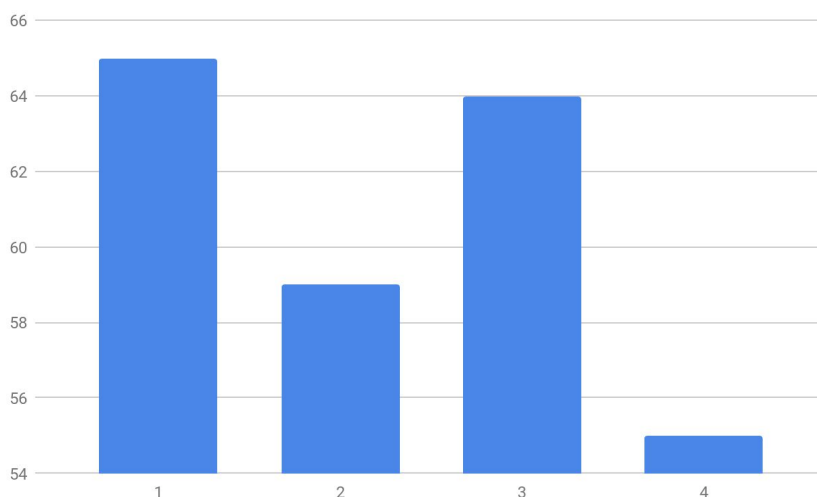


Gráfico 3: taxa de entrega (%) para as simulações do algoritmo 1

O primeiro detalhe que se nota nos gráficos é o baixo desempenho relativo na simulação 4. Isso pode ser explicado pelo fato de que se escolheu simular uma situação na qual a aplicação em execução seria de streaming, priorizando redes WLAN. Como existem menos pontos de acesso WLAN disponíveis no cenário, são feitas menos conexões e menos mensagens são entregues.

O alto desempenho relativo da simulação 1 pode ser explicado pelo fato de que apenas um dos pontos de acesso foi modelado com uma baixa largura de banda. Logo, existem mais possibilidades de bons pontos para a conexão.

Há também um esclarecimento a ser feito é a respeito da taxa de entrega geral das mensagens nas simulações. Não seria esperado que o algoritmo tivesse altíssimas taxas, pois foram inseridos pontos de acesso “defeituosos” nos termos do algoritmo,

com o objetivo de verificar se esses pontos seriam evitados. As taxas de entrega estariam então relativamente baixas devido à presença desses pontos de acesso a evitar.

Para que possamos comparar os números com os demais algoritmos (e não entre diferentes simulações de um mesmo algoritmo), considera-se que um bom parâmetro possa ser a média dos conjuntos de dados obtidos nas múltiplas simulações. Desta forma, os número que serão utilizados para a comparação final são:

- Tempo médio de entrega (média do primeiro gráfico): 0,028s
- Número de conexões iniciadas (média do segundo gráfico): 147383 conexões
- Taxa de entrega (média do terceiro gráfico): 60%

Uma possível melhoria para o primeiro algoritmo seria a consideração de pesos para os parâmetros, e a criação de uma fórmula de escolha de um ponto de acesso “menos pior”, para os casos em que nenhum dos pontos de acesso disponíveis atendem aos requisitos definidos. Nas simulações, foi colocado que caso não fosse encontrado um ponto de acesso que correspondesse aos parâmetros, seria escolhido o último ponto de acesso da lista de pontos disponíveis, independente de suas características. O ideal seria que fosse possível escolher o melhor dentre os piores pontos.

7.2 Algoritmo 2

Visto que o algoritmo 2 foi simulado de maneira simples (quatro execuções do mesmo algoritmo com os mesmo parâmetros e valores), o conjunto de dados utilizado para a análise é uma média simples entre os conjuntos de cada uma das quatro simulações. Não foram obtidas diferenças significativas para as diferentes simulações (as diferenças só são vistas depois da quarta casa decimal no caso de tempo médio de entrega e taxa de entrega, e não são vistas diferenças em termos de número de conexões), por isso não há um gráfico com os valores individuais para cada uma das simulações. Foram obtidos os seguintes dados:

- Tempo médio de entrega: 0,028s
- Número de conexões iniciadas: 138995
- Taxa de entrega: 58%

O algoritmo 2 foi simulado de maneira a facilitar o processo de descoberta de pontos de acesso futuros, de modo que o modelo de mobilidade de simulador e o mapa utilizado tornam possível a previsão do ponto com um altíssimo nível de certeza. Um

possível ambiente alternativo de testes seria um no qual as características estatísticas do algoritmo realmente fossem postas à prova, no qual não saberíamos tantos detalhes sobre o mapa ou sobre como os veículos se movimentam por ele.

7.3 Algoritmo 3

Como descrito anteriormente, o algoritmo 3 é baseado em decisões mais simples que os outros algoritmos. Então, o mesmo código foi executado quatro vezes, e a média das simulações será utilizada como conjunto de dados para comparação. Apenas uma das simulações teve resultados diferentes das outras, indicando que o viés da aleatoriedade do posicionamento dos pontos já não seria um problema tão grande para a análise. Da mesma maneira que o algoritmo anterior, não houve diferença significativa entre os diferentes conjuntos de simulação, por isso não há gráfico individual para cada um deles. Seguindo os mesmos parâmetros das simulações anteriores, os valores obtidos pela média foram:

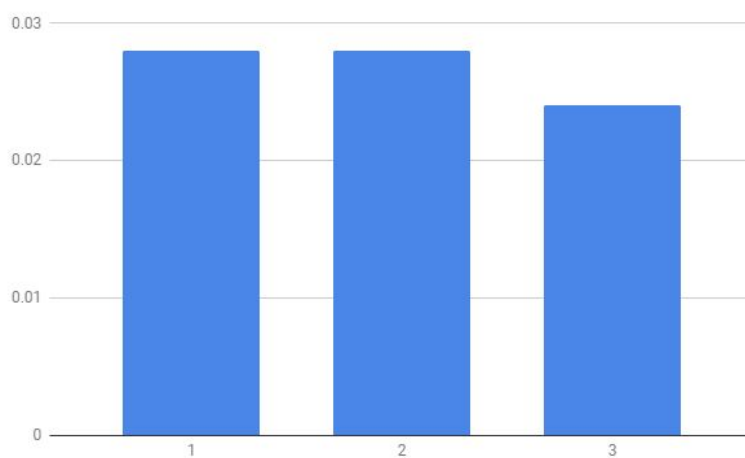
- Tempo médio de entrega: 0.024s
- Número de conexões iniciadas: 105225 conexões
- Taxa de entrega: 70%

A simulação do algoritmo 3 é talvez a mais fiel em relação ao algoritmo original proposto no artigo. A maior mudança em relação ao original é a não consideração do fluxo de decisão para a ocorrência ou não do *handoff*, o que consideramos válido dados os objetivos deste trabalho. Uma possível melhoria para o algoritmo seria levar em consideração outras variáveis de interesse, visto que utiliza apenas três variáveis para sua decisão. Entretanto, isto poderia acarretar em um maior *overhead* do algoritmo. Serão discutidas a seguir as possíveis razões pelas quais o desempenho do algoritmo 3 é superior aos anteriores.

7.4 Comparação

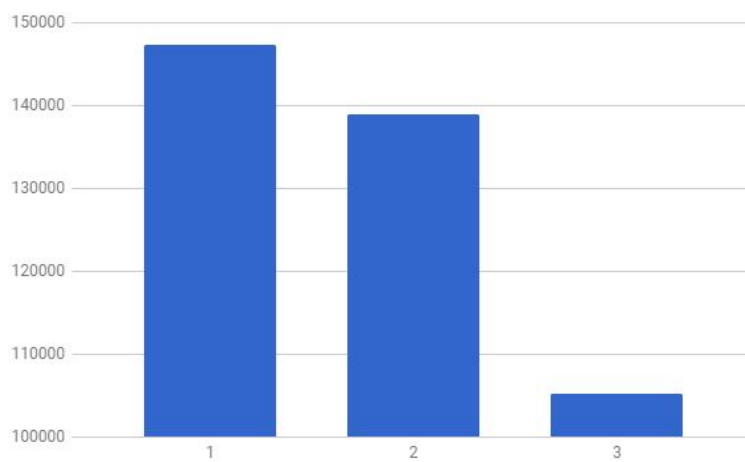
Com o objetivo de comparar os algoritmos diretamente, foram analisados em conjunto os dados das simulações de cada algoritmo. As métricas utilizadas são as mesmas das análises individuais dos algoritmos. Para obter um conjunto de dados finais para comparação, foi feita a média entre as diferentes simulações de um mesmo código. Os resultados são como apresentado a seguir:

Tempo médio de entrega:



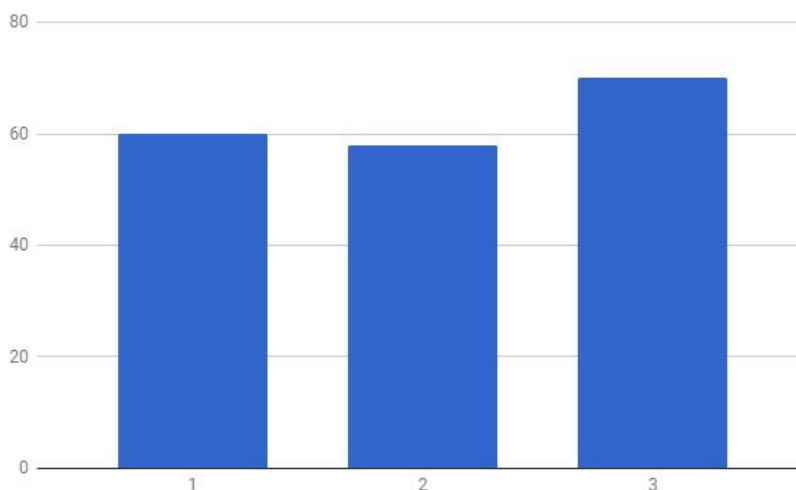
- Algoritmo 1: 0,028s
- Algoritmo 2: 0,028s
- Algoritmo 3: 0,024s

Número de conexões iniciadas:



- Algoritmo 1: 147383
- Algoritmo 2: 138995
- Algoritmo 3: 105225

Taxa de entrega (%):



- Algoritmo 1: 60%
- Algoritmo 2: 58%
- Algoritmo 3: 70%

Existem importantes diferenças conceituais a se destacar em como cada algoritmo lida com o processo de *handoff*. O algoritmo 1 apresenta uma árvore de decisão que é consultada para cada ponto de acesso com cobertura disponível em um dado momento. Todo o processamento leva a decidir se o *handoff* para o ponto analisado será executado ou não. Ao final, se nenhum ponto corresponde aos critérios, é escolhido o último ponto analisado. O algoritmo 2, por sua vez, não coloca como parte de seu processo a possibilidade de não execução de *handoff* para um ponto. No contexto deste algoritmo, existe a certeza da execução do *handoff* para um ponto muito específico, mesmo que não haja sinal disponível deste ponto ainda. O objetivo é apenas minimizar a latência do processo. No algoritmo 3, temos algo similar: a menos que não haja nenhum sinal de antena disponível, o *handoff* também é certo. O algoritmo 3 é principalmente sobre escolher o melhor dos pontos em vez de decidir se o *handoff* ocorre ou não. Embora exista um processo de decisão para a ocorrência ou não do processo (como apresentado na figura 15), este processo não leva muitos parâmetros específicos de rede em consideração, mas sim detalhes sobre a escolha do usuário, velocidade dos veículos e aplicação em execução. Quando o *handoff* ocorre no contexto do algoritmo 3, ele muito provavelmente ocorrerá para o melhor ponto possível.

O algoritmo 2 tem um desempenho razoável se comparado aos outros dois, tendo uma taxa de entrega ligeiramente pior que o primeiro, mas executando um número menor de conexões e tendo o mesmo tempo médio de entrega. Uma hipótese para este ocorrido pode ser a não utilização do histórico dos pontos (como prevê o algoritmo original), mas sim a utilização da localização dos pontos no mapa do simulador para a previsão do próximo ponto. Podemos simular, em um trabalho futuro, o algoritmo com a previsão através do histórico dos pontos e verificar quais seriam os resultados.

Podemos notar a vantagem da implantação de uma função de escolha do melhor ponto se compararmos os algoritmos 1 e 3. Embora o algoritmo 3 tenha gerado um número menor de conexões no mesmo intervalo de tempo, sua taxa de entrega acaba sendo maior do que a do algoritmo 1. Isso pode ser explicado pela escolha inteligente dos pontos, em vez de se conectar ao primeiro que ofereça condições mínimas para a execução de uma dada aplicação.

8 CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho apresentou dados de três diferentes algoritmos para *handoff* em redes veiculares, bem como dados para comparação do desempenho de cada um em relação a métricas de rede. Dos três algoritmos analisados, o algoritmo 3 teve o melhor dos desempenhos, o que leva à conclusão de que a razão entre o sinal recebido por um ponto de acesso e o ruído no local (FSNR), a capacidade dos canais disponíveis e o tempo de vida das conexões podem ser boas métricas para a decisão do processo de *handoff* em redes veiculares. Também se conclui que a implementação de uma função de escolha do melhor ponto dentre uma lista de pontos de acesso pode ser uma boa maneira de evitar desconexões futuras ou *delay* na entrega das mensagens. O algoritmo em si é relativamente simples, então também podemos concluir que um bom processo de decisão de *handoff* não está necessariamente relacionado a altas taxas de *overhead* e complexidade computacional.

Embora os resultados tenham sido esclarecedores em relação aos critérios considerados em cada algoritmo analisado, ainda existem tópicos de interesse a se investigar na busca por um processo ótimo de *handoff* em redes veiculares. Os seguintes tópicos possuem relação direta com o assunto estudado, mas não foram investigados a fundo neste trabalho:

Simulações em vários tipos de ambiente: todas as simulações foram feitas exclusivamente em um mapa, considerando-se apenas o cenário urbano. Também foi considerado que todas as aplicações possuiriam as bases estruturais para a realização das operações (por exemplo, existência obrigatória de OBUs, RSUs e GPS na

simulação). Um possível tópico de estudo é testar os algoritmos em outros mapas e, se possível, adaptá-los para o cenário V2V.

Considerações de segurança: não são verificados protocolos de segurança para a autenticação dos veículos junto a seus pontos de acesso, ou mesmo para a transferência de dados entre um veículo e um ponto. É possível estudar o mesmo tipo de algoritmo sob a perspectiva da segurança computacional, como apenas levar em consideração os protocolos de segurança das tecnologias originais e seu impacto sobre o desempenho dos algoritmos, ou se seria possível criar novas formas de tornar o processo como um todo mais seguro.

Roteamento em redes veiculares: o simulador utilizado fornece alguns modelos prontos de roteamento na rede. Como não era o interesse principal para este trabalho, todos os algoritmos utilizaram como padrão um modelo único de roteamento. Uma outra possível modificação no estudo seria analisar o impacto de diferentes tipos de roteamento sobre a eficiência do processo de handoff.

Considerar outros parâmetros: cada um dos algoritmos leva em consideração um conjunto limitado de parâmetros. Como visto, os mais comuns são a velocidade dos veículos, a força do sinal recebido pelas RSUs e a largura de banda. Um possível cenário de interesse seria a consideração de outros parâmetros na simulação, talvez incluindo parâmetros que foram considerados por um algoritmos na simulação de outro algoritmo, a fim de verificar a relevância de outras variáveis no processo de handoff.

Testar aplicações específicas: os algoritmos foram testados em um cenário simples de conexões e desconexões, no qual os veículos simplesmente se deslocam através de rotas pré-estabelecidas e trocam pacotes com alguns de seus vizinhos. Pode ser interessante testar os mesmos ou outros algoritmos para aplicações diferentes. Algumas das mais comuns no contexto de sistemas de transporte inteligentes são notificação de condições graves na pista (notificar motoristas sobre acidentes e obstáculos em um dado ponto do caminho) e alertas de infração de trânsito (notificar motoristas sobre ultrapassagem de sinal vermelho ou velocidade permitida em um radar).

REFERÊNCIAS

- [1] R. S. Alves, I. V. Campbell, R. S. Couto, M. E. M. Campista, I. M. Moraes, M. G. Rubinstein, L. H. M. K. Costa, O. C. M. B. Duarte, M. Abdalla (2009). Redes Veiculares: Princípios, Aplicações e Desafios. Minicursos do XXVII Simpósio Brasileiro de Redes de Computadores. Capítulo 5, páginas 199–254. Recife.
- [2] S. Wang, C. Fan, C. H. Hsu, Q. Sun, F. Yang (2014). A vertical handoff method via self-selection decision tree for internet of vehicles. IEEE Syst. J. páginas 1–10.

- [3] H. B. Mello (2009). Proposta de roteamento para redes veiculares (VANETs) em ambientes urbanos. Tese de Doutorado, PUC-RJ. Capítulo 2, páginas 19-29. Rio de Janeiro.
- [4] R. S. Deshmukh, T. S. Chouhan, P. Vetrivelan (2015). VANETS Model: Vehicle-to-Vehicle, Infrastructure-to-Infrastructure and Vehicle-to-Infrastructure Communication using NS-3. SENSE Department VIT University. India.
- [5] M. S. Wangham, M. Nogueira, C. P. Fernandes, O. Paviani, B. F. da Silva (2014). Segurança em Redes Veiculares: Inovações e Direções Futuras. Minicursos do XIV Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais — SBSeg.
- [6] Equipe de Redação da USP (2012). Redes veiculares inteligentes ampliam segurança e eficiência no transporte. Editorial de Tecnologia da USP.
- [7] Y.B. Lin, A.C Pang (2000). Comparison between soft and hard handoffs. IEEE Transactions on Vehicular Technology, Vol. 49, N°. 3.
- [8] J. Khan (2010). Handover management in GSM cellular systems. International Journal of Computer Applications, Volume 8, N°.12.
- [9]] M. Stemm, R. H. Katz (1998). Vertical handoffs in wireless overlay networks. ACM MONET.
- [10] L.J. Chen, T. Sun, M. Gerla (2006). USHA: simple and practical seamless vertical handoff solution. IEEE CCNC.
- [11] S. Lee, K. Sriram, K. Kim, Y.H. Kim, N. Golmie (2009). Vertical handoff decision algorithms for providing optimized performance in heterogeneous wireless networks, IEEE Transactions on Vehicular Technology 58, páginas 865–881.
- [12] K. Zhu, D. Niyato, P. Wang, E. Hossain, D.I. Kim (2009). Mobility and handoff management in vehicular networks: a survey. Wireless Communications and Mobile Computing.
- [13] Z. Simate (2013). A Comparative Study of Handoff Schemes in Vehicle to Infrastructure using Mobile WiMAX. IJCST, Volume 4.
- [14] A. Magnano, X. Fei, A. Boukerche (2015). Predictive mobile IP handover for vehicular networks. Proc.of 40th IEEE Conference on Local Computer Networks (LCN), páginas 338–346.

[15] A.S. Sadiq, K.A. Bakar, K.Z. Ghafoor, J. Lloret, R. Khokhar (2013). An intelligent vertical handover scheme for audio and video streaming in heterogeneous vehicular networks.