Universidade Federal do Rio de Janeiro

Escola Politécnica

Departamento de Eletrônica e de Computação

Um Sistema de Localização e Previsão de Chegada dos Veículos de Transporte Público Usando Redes IEEE 802.11

Autor:

Vitor Borges Coutinho da Silva

Orientador:

Prof. Miguel Elias Mitre Campista, D.Sc.

Coorientador:

Prof. Luís Henrique Maciel Kosmalski Costa, Dr.

Examinador:

Prof. Marcelo Gonçalves Rubinstein, D.Sc.

Examinador:

Prof. Marcelo Luiz Drumond Lanza, M.Sc.

DEL

Agosto de 2013

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
Escola Politécnica - Departamento de Eletrônica e de Computação
Centro de Tecnologia, bloco H, sala H-217, Cidade Universitária
Rio de Janeiro - RJ CEP 21949-900

DEDICATÓRIA

À minha família.

AGRADECIMENTO

Agradeço a meu pai Julio e minha mãe Iêda por sempre me apoiarem e por investirem em minha formação. Agradeço ainda a meus pais por minha educação como pessoa, pelos conselhos e pelo carinho. Agradeço a minha irmã Patricia por não ter me ensinado a fazer brigadeiro. Brincadeiras à parte, agradeço a minha irmã por todos os conselhos e brincadeiras. Agradeço a toda minha família pela atenção e carinho.

Agradeço a todos os amigos da faculdade, em especial Vitor Rosa, Vitor Antunes, Nilson Carvalho, Jonathan Gois, e Alan Dantas, por tornarem a graduação mais divertida e agradável. Agradeço a todos os amigos de fora da faculdade pelo apoio e pela diversão, em especial Cristiano Barbosa, Luiza Ferreira, Pedro Freitas e Ubirajara Ribeiro.

Agradeço a todos os professores que me deram aula, sem exceção, todos foram essenciais à minha formação. Agradeço a equipe do Grupo de Teleinformática e Automação pelo apoio durante este trabalho. Em especial agradeço aos meus orientadores Miguel Campista e Luís Costa pela dedicação e auxílio neste projeto. Agradeço também a banca avaliadora pela atenção.

Por fim, agradeço a Deus por estar presente em minha vida.

RESUMO

Atualmente, as grandes cidades sofrem com os engarrafamentos gerados pelo crescente volume de veículos nas ruas. A principal causa desse problema se deve ao uso excessivo de meios de transporte particulares, em detrimento ao uso dos transportes públicos de massa, como ônibus e trens. Esse comportamento ocorre, pois as pessoas julgam os transportes de massa desconfortáveis e não confiáveis o suficiente. Portanto, a baixa qualidade de serviço oferecida aos usuários, os repele do uso dos transportes públicos de massa. Com isso, cada vez mais pessoas utilizam veículos particulares, agravando a situação do trânsito nas cidades. Uma solução para resolver o problema do trânsito é prover o alargamento das vias mais utilizadas, aumentando a capacidade de escoamento do trânsito. Contudo, essa solução é temporária, visto que soluciona o sintoma do problema e não o problema em si. Para efetivamente solucionar o problema do trânsito nas grandes cidades, uma alternativa é estimular a utilização dos transportes de massa. Todavia, para isso é necessário aumentar a qualidade de serviço oferecida aos usuários. O presente projeto atua em favor desse aumento através do desenvolvimento de um sistema que calcula e entrega informações de horário de chegada de ônibus aos pontos de parada aos usuários. Os usuários ao receberem essa informação podem se programar para ir ao ponto de parada apenas quando estiver perto de seu meio de transporte chegar, aumentando o conforto dos usuários. Para estimar o horário de chegada dos ônibus nos pontos de parada, o sistema desenvolvido utiliza informações recentes do trânsito. Os resultados obtidos mostram que o sistema criado é capaz de atender as demandas de uma cidade grande como o Rio de Janeiro.

Palavras-Chave: Sistemas de Transporte Inteligente, Estimativa de Hora de Chegada, Localização Automática de Veículo.

ABSTRACT

Currently, major cities suffer from congestion generated by the growing volume of vehicles on the streets. The main cause of this problem is due to the excessive use of private means of transportation, rather than the use of mass public transportation such as buses and trains. This behavior occurs because people judge mass transportation uncomfortable and not reliable enough. Therefore, the low quality of service offered to users, repels them from the use of mass public transportation systems. With this, more and more people use private vehicles, worsening the traffic situation in the cities. A solution to solve the traffic problem is to provide the extension of the most used routes, increasing the flow capacity of the transit. However, this solution is temporary, since it addresses the symptom of the problem and not the problem itself. To effectively solve the traffic problem in big cities, an alternative is to encourage the use of public mass transportation. However, to achieve this goal, it is necessary to increase the quality of service offered to users. This project operates in favor of this increase through the development of a system that calculates and delivers bus arrival time information to users. With users recieving this information, they can plan to go to the bus stop only when the buses are close to arrive, increasing users comfort. To estimate the time of arrival of buses in bus stops, the developed system uses the latest information about the traffic. The results show that the system created is able to meet the demands of a big city like Rio de Janeiro.

Key-words: Intelligent Transportation Systems, Bus Arrival Time Estimation, Automatic Vehicle Location.

SIGLAS

API - Application Programming Interface;

APTS - Advanced Public Transportation Systems;

ARP - Analisador de Redes de Petri;

ARM - Advanced RISC Machine;

AVL - Automatic Vehicle Location;

GB - Gigabyte;

GPS - Global Positioning System;

GTA - Grupo de Teleinformática e Automação;

ITS - Intelligent Transportation Systems;

MB - Megabyte;

RAM - Random Access Memory;

SSID - Service Set IDentifier;

UA - Unidade de Acostamento;

UFRJ - Universidade Federal do Rio de Janeiro;

USB - Universal Serial Bus;

V2I - Vehicle-to-Infrastructure.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

As grandes metrópoles mundiais vêm sofrendo com engarrafamentos gerados pelo aumento de volume de veículos. Um número crescente de pessoas escolhe, diariamente, se locomover pela cidade através de meios de transporte particulares, como carros e motos. Ao menos no Brasil, a principal justificativa é que os meios de transporte públicos não são confortáveis o suficiente e que os transportes particulares são mais rápidos. O problema é que essa escolha aumenta o volume do trânsito, gerando mais engarrafamentos e fazendo com que mais pessoas optem pelo uso de transportes particulares, agravando mais ainda o problema.

O problema do trânsito nas grandes metrópoles pode ser resolvido de maneira paliativa através do aumento das vias, o que já vem sendo feito nas principais cidades do mundo. Contudo, essa solução não resolve o problema, mas sim os sintomas criados em um primeiro momento, já que o número de veículos particulares continua aumentando. De maneira mais efetiva, uma solução que agiria sobre a fonte dos problemas, isto é, sobre o excesso de veículos particulares, é o estímulo do uso dos meios de transporte públicos. Entretanto, para atrair mais passageiros para os sistemas de transporte públicos, estes devem ser melhorados, de forma a atingir uma qualidade maior dos serviços prestados aos usuários, bem como uma maior confiabilidade. Uma vez que mais pessoas utilizem os veículos de transporte públicos, por exemplo, os ônibus, os problemas do trânsito na cidade serão reduzidos. Essa melhoria é consequência direta da maior capacidade de transporte de pessoas por unidade de área ocupada de via, se comparado aos veículos particulares.

1.1 Tema

O presente trabalho está inserido na área de Sistemas Inteligentes de Transporte (`Intelligent Transportation Systems` - ITS), mais especificamente na área de Sistemas Avançados de Transporte Público (`Advanced Public Transportation Systems` - APTS). Os APTS vêm sendo pesquisados e desenvolvidos com a intenção de aumentar a qualidade de serviço dos transportes públicos. O objetivo é atrair mais passageiros para esses meios de transporte, reduzindo o volume de veículos no trânsito.

1.2 Delimitação

Os meios de transporte público usualmente seguem rotas predefinidas, exceto em caso de acidentes, interdição de vias ou desvios de rota realizados pelo condutor, sejam esses intencionais ou por erro humano. Portanto, em condições normais de operação, efetuar o monitoramento dos veículos a fim de estimar o tempo de chegada em pontos de interesse futuros é computacionalmente viável.

Este projeto final possui como premissa a existência de meios de transporte público com rotas previamente definidas. Mais especificamente, os testes e simulações realizados avaliam o caso dos ônibus. O modelo computacional foca no monitoramento dos veículos em pontos de interesse. Nesses pontos, assume-se que existam equipamentos IEEE 802.11 para coleta de dados e que o sistema realize a previsão de chegada apenas para esses pontos. No caso dos ônibus, os pontos de interesse são os pontos de ônibus, onde os usuários embarcam e desembarcam.

1.3 Justificativa

O presente projeto visa melhorar a qualidade dos transportes públicos através da entrega de mais informação acerca desses transportes aos seus usuários. Ao disponibilizar informações sobre o tempo que os próximos ônibus levam para chegar aos pontos de ônibus, pode-se minimizar o tempo de espera dos usuários. A pessoa, ciente do horário do ônibus, se programará para ir até o ponto de ônibus apenas

quando estiver perto do ônibus desejado chegar, provendo um melhor atendimento ao usuário.

Em outros países, os sistemas de previsão de chegada de transporte público são comuns. No Brasil, entretanto, ainda são poucas as cidades com iniciativas semelhantes. Logo, o presente projeto visa preencher essa lacuna.

1.4 Objetivos

O objetivo é propor e desenvolver um sistema que, utilizando roteadores sem-fio de baixo custo instalados nos ônibus e nos pontos de ônibus, seja capaz de rastrear e estimar os horários de chegada dos ônibus nos pontos requisitados pelos usuários. Mais especificamente, são desenvolvidos:

1. um programa para roteadores sem-fio com baixa capacidade de processamento que informe a posição do ônibus na infraestrutura da rede;
2. um programa que centralize as informações dadas pelos ônibus monitorando a trajetória e estimando os tempos nos trechos das trajetórias, e;
3. uma interface para disponibilizar a informação gerada aos usuários dos meios de transporte público.

Ao final deste projeto, pretende-se oferecer uma alternativa para solucionar o problema da espera prolongada dos usuários por transportes públicos nos pontos de parada devido à falta de informação do serviço.

1.5 Metodologia

Para atender os objetivos do projeto, inicialmente, foi necessário avaliar se a arquitetura proposta funcionaria corretamente. Para realizar a avaliação, foi desenvolvido um modelo em redes de Petri que representa, simplificadamente, a arquitetura do sistema. Em seguida, a rede desenvolvida foi submetida a um simulador, para que as suas propriedades, assim como possíveis problemas, fossem identificados.

Após a análise, a implementação do sistema foi iniciada, começando pela criação de uma versão simplificada do programa que centralizaria a informação dos ônibus, onde somente os envios e recebimentos de mensagem foram programados. Em seguida, foram criados os programas para os roteadores sem-fio necessários para informar a posição dos veículos. Então, para verificar se o programa central recebia as mensagens que indicavam a posição dos veículos, testes preliminares foram realizados.

Após verificar que a comunicação entre os roteadores e o programa central estava funcionando, o programa central foi incrementado para tratar as mensagens recebidas dos ônibus, de forma que os ônibus fossem monitorados e os tempos nos trechos das rotas estimados.

Ao final de tudo, criou-se uma página da Internet. Essa página é a interface gráfica do sistema, na qual os usuários podem solicitar os cálculos de estimativa que eles desejarem e ainda receber as respostas às suas solicitações.

Após a realização das etapas anteriores, o funcionamento do sistema foi avaliado por meio de simulações. Dentre as simulações, foi utilizado um protótipo com roteadores, agregando as características dos roteadores à simulação e, portanto, tornando-a mais realista. Os resultados mostram que o sistema é capaz de prover o serviço de estimativa de chegada dos ônibus em métropoles como o Rio de Janeiro.

1.6 Organização do Texto

O Capítulo 2 aborda os Sistemas Inteligentes de Transporte, sendo apresentados os principais serviços visados por esses sistemas. O Capítulo 3 apresenta trabalhos relacionados do projeto. A arquitetura proposta bem como uma visão mais geral do sistema são apresentadas no Capítulo 4. No Capítulo 5, os detalhes da implementação do sistema e exemplos de funcionamento são explicitados. Já o Capítulo 6 mostra os testes aos quais o sistema foi submetido e seus resultados. Por fim, o Capítulo 7 apresenta as conclusões e os trabalhos futuros.

Capítulo 2

Sistemas de Transporte Inteligente nos Meios de Transporte Públicos

2.1 Introdução

Na cidade do Rio de Janeiro, em 2011, aproximadamente dois milhões e meio de veículos compunham a frota ativa [2]. Hoje, o controle e o monitoramento do tráfego criado por essa imensa frota são realizados por agentes de trânsito auxiliados por câmeras espalhadas pela cidade. Os agentes dependem da existência dessas câmeras para visualizar o panorama do trânsito na cidade e tomar decisões baseadas no que eles veem. Contudo, algumas vezes, situações que podem trazer complicações para o trânsito da cidade ocorrem fora da abrangência dessas câmeras. Além disso, muitas vezes são necessárias interferências manuais para resolver os problemas de trânsito, por exemplo, o destacamento de guardas de trânsito nas ruas para coordenar os semáforos em períodos de grande fluxo de veículos.

Uma alternativa moderna às câmeras para controle de trânsito são os sistemas inteligentes de transporte (`Intelligent Transportation Systems` - ITS) que reúnem as tecnologias desenvolvidas nas últimas décadas aplicadas no cenário dos sistemas de transporte. Essas tecnologias incluem desde sensores até comunicações sem-fio [3]. O uso dessas novas tecnologias disponibiliza mais informação, permitindo que as decisões sejam tomadas com maior segurança. Além disso, soluções computacionais estão sendo criadas para automatizar alguns processos do sistema de

transporte, por exemplo, o pagamento de passagens e a contagem de passageiros, o que além de permitir parcialmente o gerenciamento remoto do sistema, também facilita e agiliza seu monitoramento. No presente trabalho, somente as soluções de ITS voltadas para meios de transporte públicos são avaliadas, ou seja, apenas as soluções para sistemas avançados de transporte público (`Advanced Public Transportation Systems` - APTS) são abordadas.

2.2 Entidades

A separação por entidades é uma forma de facilitar a visualização dos papéis que existem no contexto dos APTS. As entidades podem oferecer serviços para as outras entidades e consumir os serviços criados pelas outras entidades, ou por ela própria. Nos APTS, três entidades podem ser distinguidas [4]: Cliente, Operador e Veículo.

O Cliente é o principal consumidor de serviços do APTS. Neste trabalho, o Cliente é caracterizado como uma pessoa que tem a intenção de usar os meios de transporte público e os serviços disponibilizados pelos Operadores do APTS.

O Veículo é a principal fonte de informação dos APTS. O Veículo é também o local onde alguns dos serviços criados pelos Operadores são disponibilizados para os Clientes. Usualmente, os Veículos realizam o sensoriamento de informações como posição geográfica, número de passageiros em seu interior, velocidade, quantidade de combustível, consumo de combustível, entre outras.

Os Operadores são as empresas públicas ou privadas que através do processamento das informações recebidas dos Veículos e de outras origens externas ao sistema oferecem serviços ou monitoram e controlam o sistema de transporte. As fontes externas são necessárias, pois fatores fora do sistema podem influenciar muito no trânsito urbano. Por exemplo, em dias de chuva intensa o trânsito geralmente fica mais lento, já que o condutor é forçado a reduzir a velocidade e aumentar a distância de segurança frontal, ou seja, a distância até o veículo a sua frente.

2.3 Serviços

De maneira geral, os ITS tratam o gerenciamento do trânsito como um todo. Contudo, como já mencionado, este trabalho se restringe a avaliar os ITS somente no contexto dos meios de transporte públicos. Nesse cenário específico, os benefícios dos ITS se estendem desde os usuários até as empresas que realizam o transporte em si. Os benefícios são fruto dos serviços prestados pelos operadores, sejam os serviços para atender os requisitos dos usuários, sejam eles para atender as demandas das empresas. Nesta seção, os serviços mais importantes providos pelos sistemas de transportes inteligentes são brevemente descritos.

2.3.1 Localização Automática de Veículos

Quando não se utilizam sistemas de localização automática de veículos, o monitoramento dos veículos, e consequentemente do trânsito, é realizado tendo como base imagens obtidas por câmeras espalhadas pela cidade em um processo não automatizado. Tal fato prejudica a granularidade da informação obtida pelos sistemas de monitoramento tradicionais sobre a distribuição e posicionamento dos veículos na cidade. Um serviço de Localização Automática de Veículos (`Automatic Vehicle Location` - AVL), quando utilizado, possibilita uma granularidade superior da informação, permitindo aos ITS avaliar melhor a real situação do trânsito na cidade, e com isso tomar decisões mais seguras e melhores.

O conhecimento da posição de cada veículo cria, ainda, a possibilidade de novos serviços serem oferecidos aos usuários dos transportes públicos, por exemplo, um serviço de distribuição de conteúdos baseado na posição dos veículos. Nesse sistema seriam transmitidos notícias do bairro ou outros assuntos de interesse da região ao ônibus, onde poderiam ser vistos pelos usuários em televisores instalados nos veículos.

O primeiro passo para a construção de um sistema AVL é determinar a localização geográfica do veículo. Existem diversos métodos para localizar um dado objeto no espaço, e eles se dividem basicamente em três grandes áreas [5] que serão explicadas posteriormente:

- rádio localização;
- *dead reckoning*;
- localização por proximidade.

O segundo passo para a construção de um sistema de localização automática de veículos para sistemas inteligentes de transporte é enviar a informação da localização do veículo obtida para os Operadores, para que estes usem a informação para monitorar os veículos ou ainda criar novos serviços com base na informação disponível. As redes mais usadas para disponibilizar essa informação são as redes celulares móveis, através do uso do 3G e do 4G; e as redes sem-fio, através do uso de pontos de acesso com o padrão IEEE 802.11 distribuídos pela cidade.

2.3.1.1 Rádio localização

Os sistemas de localização via rádio podem ser classificados como unilaterais e multilaterais.

Os sistemas de localização via rádio unilaterais, de maneira geral, utilizam sinais de radiofrequência de diferentes fontes conhecidas para determinar a posição geográfica de um nó em relação às fontes. Os sinais de rádio são recebidos pelo nó a ser localizado, e quando isso ocorre, ele calcula a distância dele até cada uma das fontes emissoras dos sinais recebidos. Após o cálculo das distâncias, será realizado um processamento conhecido como trilateração para determinar de fato a posição do nó em relação às fontes dos sinais.

A técnica de trilateração se baseia em um princípio geométrico e nas distâncias entre o nó a ser localizado e as fontes de sinal. O princípio geométrico usado na trilateração se baseia na ideia da circunferência, que é o lugar geométrico em que todos os pontos de um plano que se localizam a uma mesma distância de um ponto fixo, denominado o centro da circunferência, recaem.

Usando o princípio citado e a informação da distância entre o nó e a fonte de um sinal, pode-se traçar uma circunferência com raio igual à distância calculada e

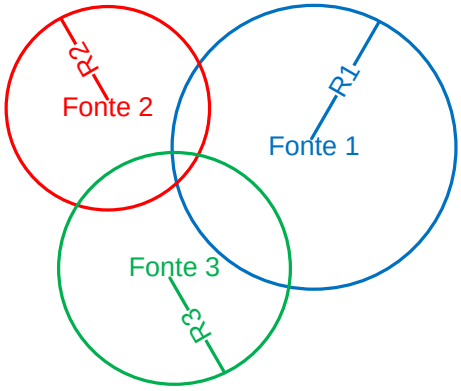

Figura 2.1: Exemplo de trilateração bidimensional.

o centro na posição da fonte de sinal. A circunferência traçada representa, então, todos os possíveis pontos que o nó pode estar em relação a posição da fonte do sinal. Se o procedimento anterior for aplicado para duas fontes de sinal distintas, passam a existir duas circunferências interceptando a posição real do nó. Como as circunferências não são coincidentes, existirão no máximo dois pontos possíveis nos quais as circunferências se interceptam, esses pontos representam os pontos do espaço onde o nó pode estar. Então, para determinar com certeza a posição do nó no espaço bidimensional, é necessário aplicar o procedimento mais uma vez, para uma terceira fonte distinta, o que irá determinar qual dos dois pontos anteriores era a localização real do nó. Como existem erros associados às distâncias medidas, no final do procedimento ao invés de obter-se um ponto exato, obtém-se uma área na interseção das três circunferências, representando as posições geográficas mais prováveis de se encontrar o nó, como ilustrado na Figura 2.1.

O processo descrito anteriormente se trata da localização de um objeto em um plano para um sistema unilateral. Entretanto, o processo pode ser facilmente estendido para a localização de um objeto no espaço tridimensional, bastando para isso utilizar esferas ao invés de circunferências e quatro fontes distintas de sinal ao invés de três. A localização efetuada é em relação às fontes de sinal usadas, no caso específico em que essas fontes estão fixas em uma posição conhecida do espaço, como latitude e longitude, ou se movem em uma trajetória pré-estabelecida. Ao

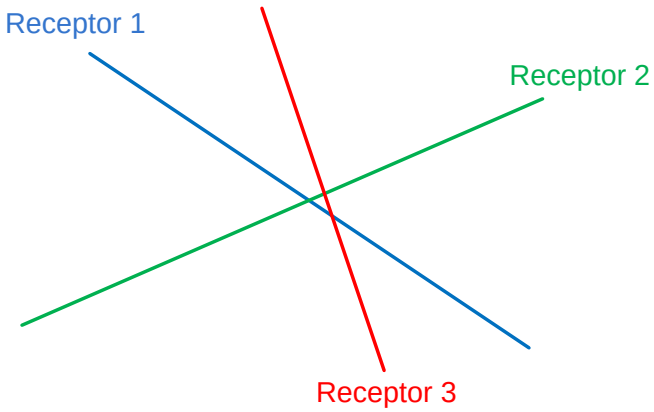

Figura 2.2: Exemplo de triangulação bidimensional.

determinar-se a posição em relação às fontes, determina-se também a posição em relação à Terra.

Nos sistemas de localização via rádio multilaterais, ao invés de o nó a ser localizado receber sinais de diferentes fontes conhecidas para determinar sua posição geográfica, diversos receptores de posição geográfica conhecida recebem um sinal transmitido pelo nó e através das diferenças na recepção desse sinal os receptores calculam a posição do nó. A técnica usada nesses sistemas para localizar o nó no espaço é a triangulação.

A triangulação se baseia na medição de ângulos ao invés da medição de distâncias. Na triangulação, os receptores devem descobrir a direção de onde o sinal recebido partiu. Duas linhas de direção distintas definem um ponto, mas como sempre existem erros nas medições, uma terceira linha de direção deve ser usada para definir a posição do nó como uma área, um triângulo, que abrange os locais mais prováveis do nó estar, como ilustrado na Figura 2.2.

Diversos processos conhecidos e estabelecidos usam a rádio localização, entre eles estão o GPS e sistemas de localização usando torres de telefonia móvel. O GPS é um exemplo de sistema unilateral, enquanto que os sistemas que usam as torres de telefonia móvel são multilaterais.

2.3.1.2 *Dead reckoning*

Nessa técnica, a posição do ponto de origem do objeto rastreado é conhecida, ela é a condição inicial do sistema. A partir desse ponto, sensores no próprio objeto rastreiam os movimentos. Assim, as informações de distância percorrida e direção são sensoriadas permitindo que a localização do objeto seja conhecida em relação à origem a todo o momento.

A vantagem dessa abordagem é que o objeto não depende de nenhum elemento externo além da condição inicial para se localizar. A condição inicial, por sua vez, geralmente é dada por um sistema de rádio localização, como o GPS. O grande problema desses sistemas é o acúmulo de erro. Especificamente para o caso de um veículo, se a distância percorrida for calculada no veículo usando um odômetro, o raio das rodas deve ser constante. Todavia, isso não acontece sempre, pois o raio das rodas varia com mudanças de temperatura, com mudanças no peso dos veículos e com o esvaziamento das próprias rodas, que acontece com o tempo. Essas variações se acumulam a cada rotação das rodas, produzindo erros no cálculo da distância. Mesmo no caso de serem utilizados acelerômetros, a acumulação de erro acontece. Os erros citados podem ser reduzidos se existirem outras referências no percurso dos veículos. Referências adicionais limitam o acúmulo dos erros de integração ao trecho entre duas referências, já que ao alcançar uma nova referência descarta-se o erro acumulado.

Sistemas de navegação inercial são sistemas que se baseiam na técnica *Dead reckoning* e que utilizam giroscópios em conjunto com os acelerômetros. Esses sistemas estão presentes em aviões modernos e são usados quando os sistemas dependentes de fatores externos param de funcionar. Isso é realizado para que ainda seja possível estimar a posição dos aviões em casos de pane.

2.3.1.3 Localização por proximidade

Nos sistemas que efetuam localização por proximidade a posição do objeto é determinada descobrindo qual é o dispositivo de controle mais próximo do objeto. Nesses sistemas, dispositivos de controle são espalhados estrategicamente pela cidade, nos

pontos de interesse onde se quer descobrir a localização do objeto. Os dispositivos de controle podem ser ativos ou passivos, variando desde sensores magnéticos ou óticos espalhados pela cidade, até rádios transmissores e receptores.

Os métodos de localização por proximidade podem funcionar de formas distintas. O objeto pode possuir um papel ativo ou passivo na sua localização. Quando o objeto possui papel ativo, ele determina sua própria posição em relação à rede de sensores espalhada. Nesse caso os sensores da rede emitem periodicamente algum tipo de mensagem com sua identificação, e o objeto descobre sua posição na rede de sensores ouvindo o canal de comunicação em busca dessas mensagens. No caso de o objeto possuir papel passivo na sua localização, é o objeto que emite mensagens periodicamente com sua identificação, sendo a rede de sensores responsável por escutar o canal em busca dessas mensagens, localizando o objeto.

A forma de localização utilizada neste trabalho é a localização por proximidade, com o veículo possuindo papel ativo na sua localização. A localização por proximidade é escolhida por possibilitar o uso de um mesmo tipo de dispositivo na fase de localização do veículo e na fase do envio de informação aos Operadores. Isso reduz a complexidade e o custo do sistema em relação às outras formas de localização, que precisam de um tipo de dispositivo na fase de localização e outro na fase do envio de informação aos Operadores. Escolheu-se fazer o veículo possuir papel ativo em sua localização, pois essa forma é mais simples de ser aplicada no cenário deste trabalho. Neste projeto os dispositivos utilizados são roteadores sem-fio.

2.3.2 Informação ao Passageiro

Os clientes dos transportes públicos, hoje, não tem informações acerca do transporte a sua disposição. Utilizando os APTS, os clientes podem ter acesso a informações como estimativa para o transporte desejado chegar onde o cliente está ou mesmo no destino e número de passageiros nos veículos. As estimativas são possíveis graças ao serviço de localização automática de veículos, que permite que os clientes saibam quando o transporte irá chegar e não percam tempo desnecessariamente. O número de passageiros nos veículos pode ser usado pelos clientes para decidir se

devem ir para o ponto de ônibus ou se devem aguardar um transporte mais vazio. Essas informações ajudam o cliente a se programar, e aumentam a confiabilidade dos sistemas de transporte públicos.

2.3.3 Gerenciamento de Sinalização de Trânsito

O gerenciamento de sinalização de trânsito entrega aos operadores um poder maior para atacar os problemas do trânsito. Nesse tipo de controle, os operadores podem modificar os intervalos de abertura e fechamento de semáforos, aumentando ou diminuindo a prioridade dos fluxos de veículos das vias. Por exemplo, suponha que em um cruzamento, uma via esteja com tráfego maior que o da outra via do cruzamento. Nesse caso, os operadores podem modificar os intervalos de abertura e fechamento dos semáforos do cruzamento, aumentando o tempo que a via congestionada permanece com o sinal aberto. Como consequência, o intervalo de tempo que a via com trânsito menos intenso permanece fechada aumenta, permitindo um maior escoamento do tráfego. Outro possível uso para esse serviço seria no caso da presença de um veículo com prioridade, por exemplo, ambulâncias, carros do corpo de bombeiros ou da polícia. Nesse cenário, o operador poderia controlar os semáforos de forma a agilizar a chegada desses veículos em seus destinos. De forma similar, o presente serviço também pode ser utilizado para priorizar vias com maior número de veículos de transporte público. O serviço de gerenciamento das sinalizações de trânsito pode ser automatizado, sendo os intervalos de abertura e fechamento dos sinais controlados por computadores [6].

2.3.4 Contagem Automática de Passageiros

Através de sensores dentro dos veículos, é possível contabilizar o número de pessoas dentro deles de forma automática. Esse serviço é interessante tanto para as empresas que oferecem transportes públicos quanto para o público que utiliza essa forma de transporte. Para o primeiro grupo, o serviço oferece uma forma de controlar os ganhos e obter estatísticas, como o número de pessoas que utilizam as diferentes linhas de ônibus da empresa nas diferentes horas do dia. Com isso a empresa poderia realocar seus ônibus para as linhas mais usadas no momento, aumentando sua qualidade de serviço. O aumento da qualidade beneficia diretamente o segundo grupo,

ou seja, os usuários. Além disso, se o número de passageiros dentro dos veículos for divulgado, os usuários poderão escolher esperar por ônibus mais vazios.

2.3.5 Sistema Eletrônico de Cobrança

O serviço de cobrança eletrônica já está disponível em todos os ônibus da cidade do Rio de Janeiro. Com esse serviço, o usuário compra ou recarrega um cartão inteligente com um determinado saldo, e ao entrar no veículo, passa o cartão em um dispositivo eletrônico que desconta o valor da passagem do saldo presente no cartão. Esse serviço facilita o controle financeiro das empresas de ônibus; torna os ônibus menos suscetíveis a assaltos para roubo do dinheiro em posse do trocador; aumenta a velocidade da cobrança dos passageiros, pois o passageiro não precisa receber o troco; viabiliza a existência de sistemas como o Bilhete Único Carioca, que provê descontos aos usuários que precisam utilizar mais de um ônibus num curto intervalo de tempo; e ainda permite ao usuário visualizar o histórico de operações que ele efetuou [7].

2.3.6 WiFi para o Passageiro

As pessoas desejam estar conectadas em rede a todo o momento e em qualquer lugar. Essa premissa não deve ser diferente no interior de meios de transporte públicos. Atualmente, alguns passageiros já acessam a Internet por meio de redes celulares 3G ou 4G. Porém, ao oferecer o serviço de internet sem-fio ao passageiro usando o padrão IEEE 802.11, pode-se aumentar a razão dado obtido por energia consumida [8], o que é de primordial importância para o usuário caso ele esteja usando um smartphone ou um tablet para acessar a Internet. Além da vantagem energética para o usuário, com a disponibilização desse serviço gratuitamente, o custo do acesso à Internet dos usuários diminuiria. Logo, oferecer serviço de acesso à Internet dentro dos veículos, aumenta a qualidade do serviço prestado e o conforto do passageiro, o que atrai mais pessoas para o transporte público.

2.3.7 TV nos Transportes

Outro serviço voltado para o aumento da qualidade de serviço e do conforto do passageiro é o uso de televisões nos meios de transporte. Inserir televisões nos meios de transportes é uma forma de manter o cliente informado, entretê-lo, tornar seu tempo gasto no trânsito uma experiência menos estressante. Esse serviço pode ser financiado por propagandas veiculadas no próprio televisor, inclusive algumas empresas de ônibus da cidade do Rio de Janeiro já oferecem esse serviço [9].

2.4 Conclusão

Neste capítulo foram apresentados os principais serviços dos ITS relacionados aos meios de transporte públicos. Este projeto propõe um serviço de informação ao passageiro, onde a previsão de chegada dos ônibus é divulgada. Além disso, como o cálculo da previsão de chegada de um veículo requer o conhecimento da localização dele, um sistema de localização automática de veículo também é desenvolvido. O próximo capítulo apresenta os trabalhos relacionados ao projeto.

Capítulo 3

Trabalhos Relacionados

Neste capítulo, os trabalhos relacionados ao projeto são apresentados.

3.1 Localização de Veículos

A localização de veículos é um tema estudado há muitos anos no contexto de ITS. Como já enunciado no capítulo anterior, existem diversas técnicas de rastreamento de veículos.

Muitos trabalhos na área de sistemas de transporte inteligentes utilizam a localização por meio de GPS, devido à grande acurácia alcançada por esses sistemas. Contudo, sistemas de GPS exigem equipamentos especializados, e no contexto de ITS, exigem que alguma tecnologia de comunicação seja acoplada ao módulo GPS para que a informação da posição do veículo seja divulgada. Essa combinação torna o sistema mais complexo e custoso.

Em oposição ao uso comum do GPS, técnicas de localização que utilizam as próprias tecnologias de comunicação para localizar os veículos vêm sendo estudadas. Uma das vantagens do uso desses sistemas é que, além de prover a localização dos veículos, as redes criadas poderiam ser utilizadas para outros fins. Os sistemas mais simples de localização desse tipo funcionam como sistemas de localização por proximidade, sendo a localização do veículo dada apenas pela informação sobre qual nó da rede ele está conectado. Em [10], é proposto um sistema que utiliza a tecnologia de comunicação sem-fio ZigBee para localizar ônibus. A vantagem da proposta é

o custo da tecnologia ZigBee ser baixo. Porém, para disseminar a informação da localização dos ônibus utilizou-se outra rede, a GPRS, o que acaba por tornar o sistema mais complexo e custoso. O ideal seria utilizar a tecnologia de comunicação escolhida para localizar e disseminar a informação criada.

A precisão dos sistemas de localização que usam tecnologias de comunicação descritos até agora é inferior à apresentada pelos sistemas de posicionamento via satélite. Contudo, técnicas para aumentar a precisão desses sistemas vêm sendo propostas. Por exemplo, em [11], o veículo alvo mede a potência do sinal recebido de vários pontos de acesso em seu entorno, e através de uma rede neural criada e calibrada no trabalho, realiza o posicionamento do veículo. Os resultados do trabalho mostram que o sistema criado obteve erros médios quadráticos de 7,02 metros. Esse resultado se aproxima do alcançado por sistemas de posicionamento via satélite, que é de 4 metros [12]. O ponto fraco dessa forma de posicionamento está na calibração da rede neural, que é um processo longo para uma cidade inteira.

3.2 Previsão de Tempo de Chegada

A partir dos dados entregues por sistemas de localização, pode-se prever o tempo que um dado veículo chegará a um lugar específico. As técnicas para o cálculo das previsões vem sendo amplamente estudadas, já que essas previsões são usadas em sistemas inteligentes de transporte, aumentando a confiabilidade desses sistemas. As principais técnicas para calcular as previsões podem ser vistas na Tabela 3.1.

Os métodos baseados em dados de histórico preveem o tempo de chegada de acordo com informações de dias anteriores, semanas e até meses [13]. A partir dessas informações, uma média das informações passadas é realizada, de dados que pertencem ao mesmo período de um mesmo dia da semana. Essas estimativas funcionam bem em circunstâncias esperadas, mas não modelam situações não cíclicas, como acidentes. Além disso, um volume de informações passadas muito grande é necessário para que essa técnica seja implantada.

Tabela 3.1: Técnicas de previsão de tempo de chegada. Adaptado de [1].

Técnica	Autor(es)	Comentários
Métodos com base em dados de histórico	Manolis e Kwstis [13]	Prevê o tempo de viagem em um tempo particular como o tempo médio de viagem para o mesmo período historicamente.
Abordagem de tempo real	Lin e Zeng [14], Karbassi e Barth [15]	Assume que o tempo futuro de viagem será o mesmo que o do presente.
Análise de séries temporais	D'Angelo et al. [16], Ishak e Al-Deek [17]	Assume que padrões históricos se manterão no futuro segundo um modelo.
Modelos de regressão	Rice e van Zwet [18], Frechette e Khan [19]	Prevê a variável dependente baseada numa função formada por um conjunto de variáveis independentes.
Técnicas de inteligência artificial	Chien et al. [20], Vanajakshi e Rilett [21], Jeong e Rilett [22]	Previsão baseada em dados de exemplo. Necessário um grande banco de dados para previsão acurada.
Abordagem baseada em modelo	Vanajakshi et al. [23], Shalaby e Farhan [24]	Estabelece relacionamentos entre as variáveis e corrobora usando observações em campo. Não é específico para região ou dados.

Ao contrário dos métodos baseados em dados de histórico, as abordagens de tempo real utilizam apenas as informações do próprio dia, modelando melhor situações particulares do dia, como acidentes. As abordagens de tempo real basicamente assumem que o tempo que um dado veículo leva em um dado percurso é igual ao tempo que um outro veículo que já fez esse trajeto levou [14]. A desvantagem desses métodos é que eles são muito afetados por perdas dos dados, por exemplo, devido à falha de equipamentos ou perdas na recepção[1].

Assim como os métodos baseados em dados de histórico, modelos de séries temporais assumem que o padrão do trânsito se manterá historicamente. Contudo, essa forma de previsão relaciona os dados de histórico com as estimativas através de modelos [16], possivelmente não lineares, e não através de médias. Com isso, variações nos dados históricos ou mudanças no relacionamento entre os dados históricos e o tempo real podem causar pouca precisão nas previsões resultantes [16].

Diferentemente das técnicas citadas anteriormente, as técnicas que utilizam modelos de regressão não assumem que o trânsito depende de dados passados. Essas técnicas realizam a previsão através de um conjunto de variáveis independentes escolhidas para modelar o trânsito. O problema desses modelos é que nos sistemas

de transporte as variáveis são muito intercorrelacionadas, limitando a aplicação da técnica [20].

Outros métodos baseados em modelos tem sido estudados bastante por não serem específicos para uma região. Dentre esses métodos, se destacam os que utilizam filtros de Kalman, pois estes tem o potencial de se adaptar as flutuações do trânsito com parâmetros dependentes do tempo [25]. Contudo quando os dados de localização são temporalmente esparsos, essa abordagem não funciona bem [15].

Por ser difícil modelar matematicamente o comportamento do trânsito, métodos que usam inteligência artificial vêm sendo desenvolvidos. Esses métodos geralmente utilizam redes neurais criadas para realizar as previsões. Isso é feito devido às suas habilidades de resolver relacionamentos não lineares complexos e emular o processo de aprendizagem do cérebro humano [25]. O principal problema dessas técnicas é a necessidade de enormes quantidades de informação no processo de aprendizagem das redes [1].

Neste projeto de conclusão de curso, os dados de localização são esparsos, já que a posição dos veículos somente é conhecida nos pontos de acesso espalhados pela via. Além disso, não há uma base de dados com informações de localização antigas. Devido a essas restrições, neste projeto, a técnica usada para previsão de tempo de chegada é a de tempo real.

Capítulo 4

Arquitetura do Sistema

4.1 Visão Geral

O trabalho desenvolvido divide-se em duas partes. A primeira é um serviço de ITS de localização automática, já que não precisa de uma ordem explícita para localizar um veículo. A segunda é um serviço de ITS de estimativa de chegada de ônibus aos pontos de parada que se baseia em informações da primeira parte.

A localização automática de veículos desenvolvida tem foco em veículos de transportes públicos, mais especificamente em ônibus, mas pode ser estendida para outros veículos. O cenário da aplicação é o de redes veiculares e o serviço se baseia na técnica de localização por proximidade, onde a localização por proximidade é realizada em relação a pontos de acesso espalhados pela via. O ônibus ao se aproximar de um desses pontos de acesso, se conecta a ele. Em seguida, ocorre uma troca de mensagens, onde o ônibus se identifica para o ponto de acesso e vice-versa. Terminada a troca de mensagens anterior, o ônibus envia uma mensagem a um computador central informando a qual ponto de acesso ele está conectado.

Ao localizarem-se os ônibus em relação aos pontos de ônibus e sabendo a rota que os ônibus de uma determinada linha percorrem, pode-se oferecer o serviço de estimativa de horário de chegada de ônibus em todos os pontos de ônibus. Basta, para isso, criar um mapa das linhas de ônibus, a partir das informações da posição dos ônibus em relação aos pontos de parada, e usar esse mapa em conjunto com o histórico de períodos que os ônibus demoram entre dois pontos de parada consecutivos da linha.

O conhecimento do mapa da linha de ônibus é essencial para o serviço de estimação de hora de chegada. A abordagem inicial, para suprir essa necessidade, é supor que a rota dos ônibus não se altera nunca, e com isso a rota das linhas de ônibus é armazenada em um arquivo que é lido sempre que necessário. Porém, essa suposição pode falhar, caso mudanças temporárias na rota ocorram, por exemplo, em caso de fechamento de uma via devido a um acidente. Esse problema já havia sido percebido nos trabalhos [26] e [27] anteriores e não havia sido abordado. Por isso, neste trabalho, os mapas das linhas de ônibus serão criados dinamicamente, com o intuito de evitar os problemas levantados nos trabalhos anteriores.

4.2 Entidades

No sistema, existem quatro entidades, são elas:

- Central;
- Unidade de Acostamento (UA);
- Ônibus;
- Cliente.

Pode-se estabelecer uma relação entre as entidades do sistema, mencionadas acima, e as entidades dos APTS citadas no Capítulo 2. As entidades do sistema Central e Unidade de Acostamento são gerenciadas por um Operador, por exemplo, a secretaria de transporte da cidade onde o sistema está sendo instalado, outra organização governamental ou ainda uma empresa privada. A entidade Ônibus do sistema corresponde à entidade Veículo caracterizada no Capítulo 2, e a entidade Cliente do sistema corresponde à entidade Cliente do Capítulo 2. As entidades do sistema estão ilustradas na Figura 4.1.

4.2.1 Central

A Central é um computador que executa o programa principal do sistema. A Central possui todas as informações do sistema, os mapas criados dinamicamente, os dados da localização de cada veículo e os históricos dos tempos entre dois pontos

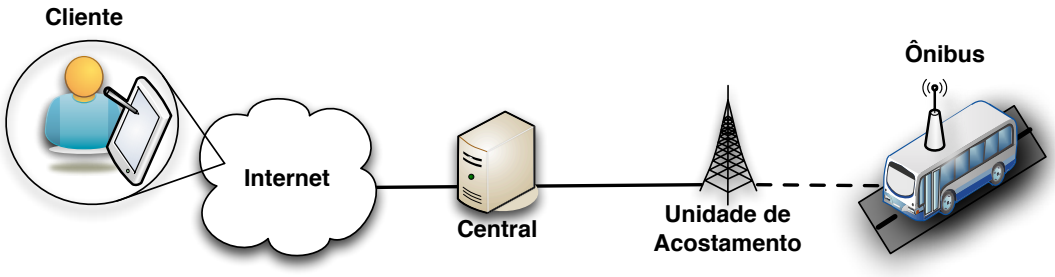

Figura 4.1: Entidades do sistema na arquitetura.

consecutivos. Históricos esses que são usados para oferecer estimativas sobre as chegadas dos ônibus aos Clientes. O pseudocódigo simplificado do programa executado na Central pode ser visto a seguir:

```
Entrada: mensagem "m" recebida do ônibus "b" servindo a linha "l"
Lista de Ônibus: B
// Cada linha de ônibus contém o seu mapa, que é uma sequência de UAs.
Lista de Linhas de Ônibus: L
// T associa um trecho da linha de ônibus ao tempo gasto para percorrê-lo
Mapa Associativo <Trecho, Tempo> T
Busca b em B
Se achou b E b não mudou de linha E mensagem anterior foi recebida
   Armazena tempo gasto no trecho entre a UA anterior e a UA atual em T
Se não achou b
   Cria ônibus e adiciona em B
Atualiza informação de localização do ônibus b com as UAs de m
Busca linha de b em L
   Se achou a linha l
      Atualiza o mapa da linha l usando as UAs de m
   Senão
      Cria mapa da linha l usando a UA atual como ponto inicial
Se a linha l não é circular
   Para cada ônibus cadastrado b' servindo a linha l faça
      Enquanto o ônibus b' não está no ponto final faça
         Acumule o tempo gasto por b' no trecho da última UA até a atual
         // No fim do cálculo a estimativa de cada UA deve ser a mínima,
         // representando o tempo para o ônibus mais próximo chegar na UA.
         Se estimativa do ônibus b' chegar < estimativa da UA atual
            estimativa da UA atual = estimativa do ônibus b' chegar
         Move o ônibus b' para a próxima UA da linha l
```

4.2.2 Unidade de Acostamento

No sistema existem várias Unidades de Acostamento. As UAs são os pontos de acesso posicionados na cidade, oferecendo redes sem-fio com identificadores (Service Set Identifier - SSID) pré-fixados e iguais. Essa premissa é necessária para que os roteadores inseridos dentro dos ônibus possam facilmente identificar e se conectar à rede de acesso. As UAs têm papel fundamental tanto na localização dos ônibus, quanto no envio dessa informação até a Central. No sistema proposto, os ônibus são localizados por proximidade em relação às UAs, e a informação da posição dos ônibus é encaminhada à Central através das UAs. Para o caso específico dos ônibus, os pontos de parada são locais interessantes para a instalação de UAs, fazendo com que as estimativas sejam calculadas para esses locais.

4.2.3 Ônibus

Os ônibus são os veículos automotores de transporte público abordados neste trabalho. Para que o veículo possa ser rastreado, ele deve possuir um roteador ou outro dispositivo com interface IEEE 802.11. Tal roteador deve ser instalado no interior do veículo e deve executar as aplicações criadas, que são responsáveis pela localização do ônibus em relação às UAs. O dispositivo instalado deve ser capaz de se conectar às redes sem-fio criadas pelas UAs.

4.2.4 Cliente

O Cliente é caracterizado como uma pessoa, que pede informações da hora de chegada dos ônibus a um dado ponto de ônibus ao serviço de estimativas do sistema. A requisição é realizada pelo Cliente por meio de uma página da Internet, e a resposta ao pedido alcança o Cliente também através de uma página da Internet, utilizando a tecnologia de acesso à Internet disponível para o Cliente.

4.3 Comunicações entre Entidades

As quatro entidades se comunicam de formas distintas, usando diferentes tecnologias para tal. Todas as entidades e as comunicações entre elas podem ser visualizadas na Figura 4.2.

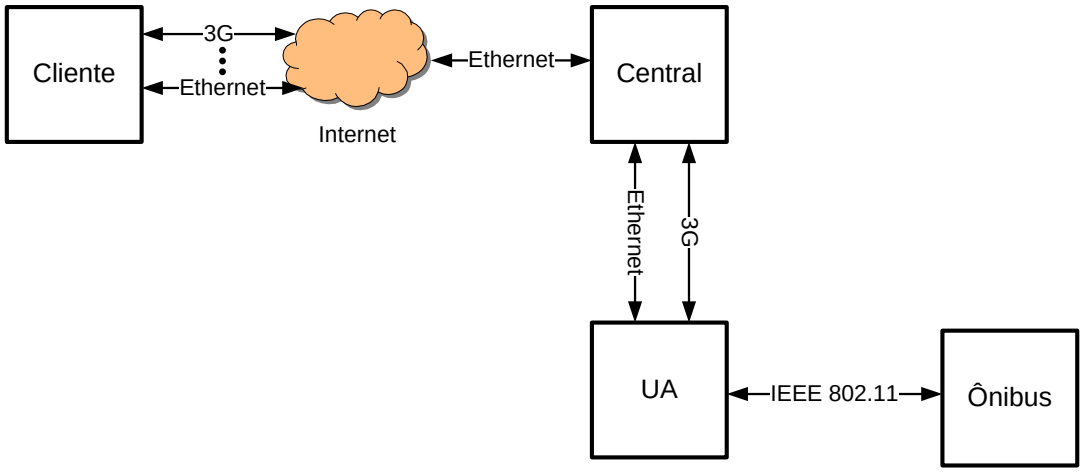

Figura 4.2: Comunicações entre entidades na arquitetura.

A comunicação entre o Cliente e a Central, ocorre através da Internet, o que garante compatibilidade, esteja o cliente usando um celular com redes móveis, esteja ele usando um computador de mesa conectado a um provedor de banda larga. A comunicação entre as UAs e a Central pode ser realizada de diferentes maneiras: através de cabos Ethernet, através de um modem 3G instalado na porta USB da UA, ou ainda, através de múltiplos saltos entre as UAs até que um contato direto com a Central seja alcançado. Os saltos seriam realizados através de uma interface de rede sem-fio secundária, instalada na porta USB da UA.

A comunicação entre os ônibus e a Central, na realidade, é realizada através das UAs. Portanto, resta apenas descrever a comunicação entre um ônibus e uma UA. Essa comunicação pode ser caracterizada como uma comunicação do tipo V2I, veículo para infraestrutura, das redes veiculares. Assim, esse tipo de comunicação está sujeita a desafios e problemas inerentes a essas redes, por exemplo, pode acontecer de a comunicação entre o ônibus e a UA não ocorrer, devido ao período de

curto de contato entre ônibus e UA.

Os principais problemas das redes veiculares decorrem do curto intervalo de contato entre os nós da rede, pois os veículos se movem rapidamente. Contudo, estudos realizados observaram que um veículo, trafegando a aproximadamente 96 quilômetros por hora, é capaz de enviar a uma UA 19 megabytes de informação usando as redes sem-fio do padrão IEEE 802.11b [28]. Isso indica que mesmo a uma velocidade superior a de um ônibus, veículo alvo do trabalho, uma quantidade relevante de informação pode ser transmitida.

No trabalho [26], mediu-se o tempo médio que os ônibus permanecem ao alcance de uma UA. Esse tempo médio é igual a 65 segundos, caso o ônibus pare no ponto para pegar passageiros, ou 36 segundos, caso contrário. Já no trabalho [29], os autores mostraram que o tempo de associação dos roteadores inseridos nos veículos com as UAs foi de até 9 segundos. Com base nesses valores, e considerando que os pacotes usados na localização são da ordem de uma centena de bytes, pode-se supor que o movimento dos ônibus não é problemático para a aplicação, podendo ser usado o padrão IEEE 802.11b nas comunicações entre os ônibus e as UAs.

4.4 Modelo em Rede de Petri

Como o sistema envolve a comunicação de diversas entidades, optou-se por modelar o sistema, utilizando métodos formais simplificadamente, antes da implementação ser iniciada. A modelagem em rede de Petri foi escolhida, pois possui a capacidade de representar bem sistemas distribuídos discretos, o que é o caso do sistema desenvolvido.

Como o sistema é formado por dois serviços diferentes, a rede de Petri completa do sistema foi dividida em duas partes para facilitar a visualização: uma parte para o serviço AVL, na Figura 4.3, e a outra para o serviço de previsão de chegada de ônibus, na Figura 4.4. Pelas figuras, nota-se que o serviço de localização faz uso de comunicações entre todas as entidades, excetuando-se a Cliente, enquanto o de previsão de chegada de ônibus depende apenas da comunicação entre a Central e o

Cliente. Além disso, observa-se que a Central é a entidade que une ambos os serviços no sistema, sendo a única que aparece nas duas partes da rede de Petri completa.

O comportamento do sistema está descrito nas redes de Petri. O serviço de localização se inicia com o ônibus se identificando à UA e vice-versa. Em seguida, o ônibus envia uma mensagem à Central informando a qual UA ele está conectado, a última UA que ele esteve conectado, a linha de ônibus que ele está servindo e o seu próprio identificador. A partir dessa mensagem, a Central cria ou atualiza o mapa da linha de ônibus que o ônibus está realizando. Após o processamento do mapa da linha de ônibus, as estimativas de horário de chegada dos próximos ônibus a todas as UAs do mapa são calculadas. O serviço de previsão de chegada de ônibus é oferecido aos Clientes. Neste serviço os Clientes solicitam as estimativas à Central enviando uma mensagem com as informações da solicitação. A Central, ao receber a solicitação, envia uma mensagem ao Cliente contendo a estimativa solicitada.

A descrição dos lugares da rede de Petri completa criada encontra-se abaixo:

- P1 - representa o repouso da UA, e possui uma ficha, significando que a UA sempre está apta a receber e enviar mensagens para os ônibus, desde que seja uma por vez.
- P2 - denota as mensagens já enviadas pelo ônibus, porém ainda não recebidas pela UA.
- P3 - representa as mensagens enviadas pela UA, porém ainda não recebidas pelo ônibus.
- P4 - é o estado inicial do comportamento dos ônibus, pode conter N fichas, cada uma representando um ônibus, ainda desconectado.
- P5 - representa quando o ônibus já está apto a enviar uma mensagem à UA, quando ele se associou a UA.
- P6 - denota a espera do ônibus pela chegada da mensagem enviada pela UA, antes de enviar uma mensagem à Central.

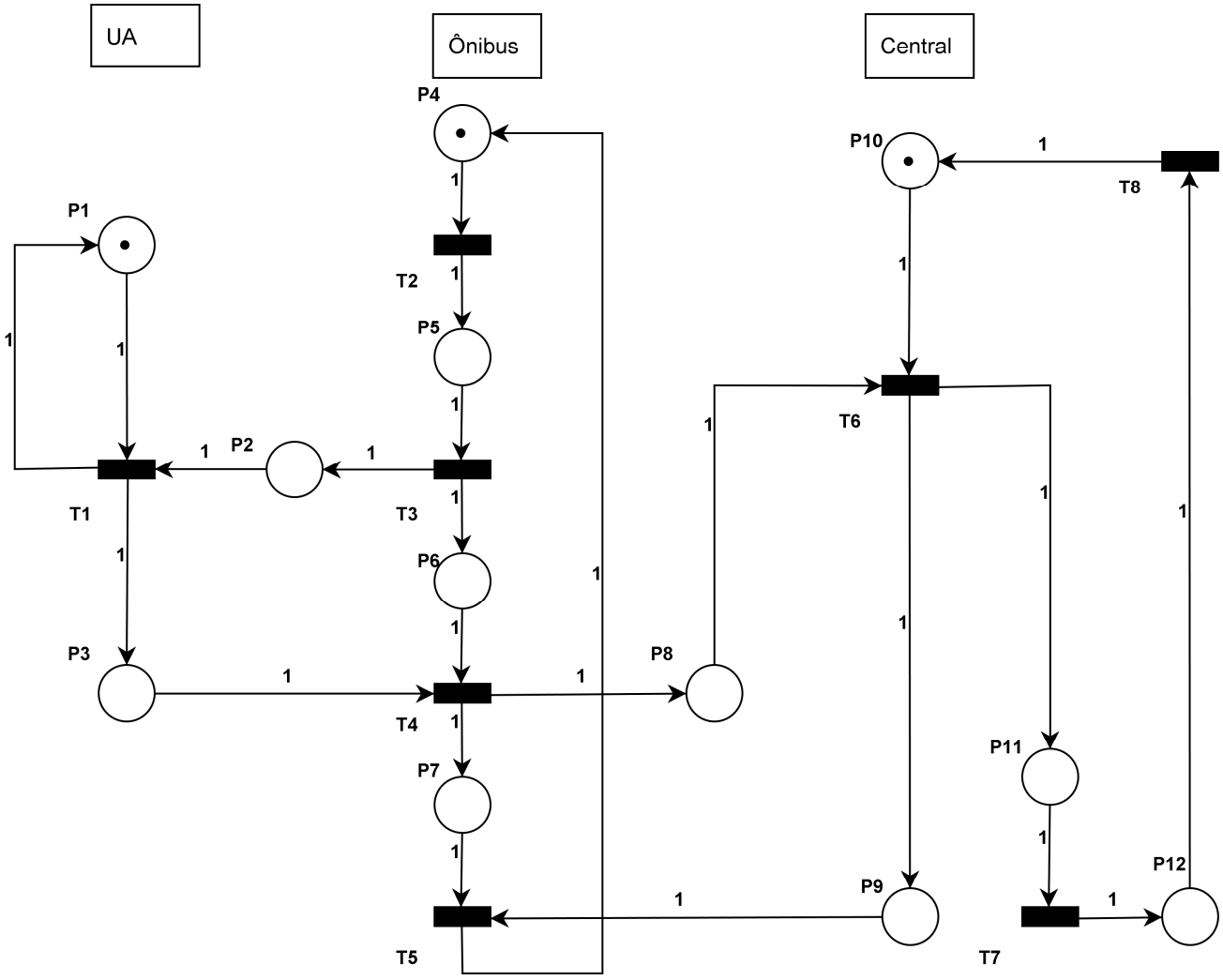

Figura 4.3: Rede de Petri do serviço de localização automática de veículos.

- P7 - é o estado onde o ônibus espera a resposta da Central, antes de retornar ao estado inicial.
- P8 - contém as mensagens enviadas pelo ônibus à Central ainda não entregues.
- P9 - contém as mensagens enviadas pela Central ao ônibus ainda não recebidas por este.
- P10 - é o estado inicial da Central, possui duas fichas para representar que uma mensagem de localização e uma requisição de estimativa do Cliente podem ser atendidas em simultâneo.
- P11 - representa o estado em que a mensagem recebida pela Central está em tratamento para localizar o veículo e construir ou atualizar o mapa da linha

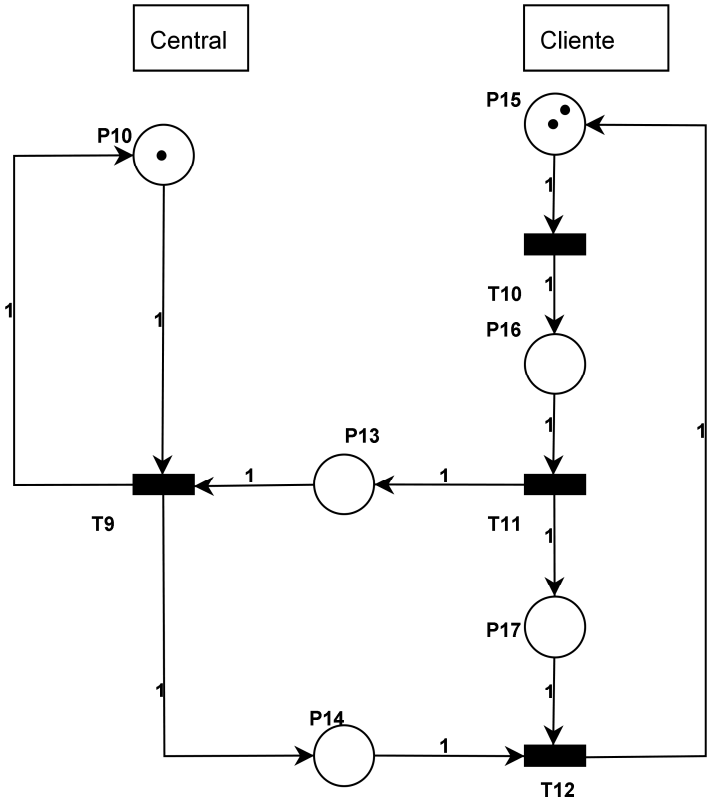

Figura 4.4: Rede de Petri do serviço de previsão de chegada de ônibus.

de ônibus.

- P12 - demonstra que a estimativa da linha de ônibus da mensagem esta sendo calculada pela Central.
- P13 - indica o pedido de estimativa do Cliente em trânsito para a Central.
- P14 - indica a mensagem da Central com a estimativa pedida indo para o Cliente.
- P15 - é o estado inicial do comportamento dos Clientes, pode conter N fichas, esse estado representa o universo de Clientes.
- P16 - é o estado que indica que o Cliente está no site de requisição de estimativas.
- P17 - indica o Cliente aguardando a resposta a sua requisição.

A descrição das transições da rede de Petri completa desenvolvida pode ser vista a seguir:

- T1 - indica as ações de recebimento na UA de uma mensagem vinda do ônibus e envio de uma mensagem da UA para o ônibus.
- T2 - indica a associação do ônibus à UA.
- T3 - corresponde ao envio de uma mensagem do ônibus para a UA.
- T4 - corresponde ao recebimento de uma mensagem proveniente da UA pelo ônibus e envio de uma mensagem do ônibus à Central.
- T5 - indica o recebimento pelo ônibus de uma mensagem enviada pela Central.
- T6 - denota as ações de recebimento na Central de uma mensagem vinda do ônibus e envio de uma mensagem da Central para o ônibus.
- T7 - indica o fim do tratamento da mensagem enviada pelo ônibus.
- T8 - indica o fim do cálculo da estimativa da linha de ônibus indicada na mensagem recebida.
- T9 - corresponde ao recebimento de um pedido de estimativa vindo do Cliente pela Central e envio da resposta desse pedido da Central ao Cliente.
- T10 - mostra a entrada do Cliente na página de requisição de estimativas.
- T11 - indica o envio da requisição de estimativa do Cliente à Central.
- T12 - denota o recebimento da resposta a requisição enviada pela Central ao Cliente.

A rede completa criada foi inserida em um simulador de redes de Petri, o Analisador de Redes de Petri (ARP) [30], para que as propriedades da rede fossem levantadas e possíveis erros de lógica fossem retirados. Para inserir a rede no simulador, foi necessário descrever a rede em forma de texto. A descrição em texto da rede e a saída da análise da rede feita pelo simulador se encontram no Apêndice A.

Dentre as propriedades obtidas, as mais importantes foram:

- A rede é viva, ou seja, não ocorrem bloqueios na arquitetura.
- A rede é limitada, o que significa que a arquitetura não cria um número infinito de mensagens a partir de um número finito.
- A rede é reinicializável, o que mostra que a arquitetura após rastrear um ônibus, ou responder uma estimativa de um cliente, estará apta a realizar qualquer das duas tarefas novamente.

Como as propriedades indicaram que a rede funcionava como esperado, então o desenvolvimento da aplicação pôde ser iniciado. O próximo capítulo apresentará a implementação realizada do sistema.

Capítulo 5

Implementação do Sistema

Este capítulo descreve o funcionamento do sistema do ponto de vista da implementação. Neste projeto final foram criados os quatro programas, cujas tarefas são apresentadas, simplificadamente, abaixo:

- `vbcsBusClientUA` - Este programa recebe a mensagem enviada pelas UAs com a identificação delas e envia a mensagem com a identificação do próprio ônibus às UAs.

- `vbcsBusClientCentral` - Este programa recebe a mensagem enviada pela Central para criação de histórico de comunicações e envia a mensagem com as informações sobre o próprio ônibus e sobre sua localização na rede.

- `vbcsUAServer` - Este programa envia a mensagem com a identificação da própria UA para o ônibus e recebe a mensagem com a identificação do ônibus enviada por ele para criação de histórico de comunicações.

- `vbcsCentralServer` - Este programa envia uma mensagem ao ônibus para criação de histórico de comunicações e recebe a mensagem enviada pelo ônibus com as informações sobre o próprio ônibus e sua posição na rede. Este programa, a partir das mensagens recebidas vindas dos ônibus, cria o mapa das linhas de ônibus, localiza os ônibus e calcula as estimativas de tempo de chegada dos ônibus. Além disso, é este programa que recebe e responde os pedidos de estimativa dos Clientes.

5.1 Localização Automática de Veículos

Desde o início, cada UA executa um programa, chamado `vbcsUAServer`, que aguarda as conexões dos ônibus ao socket da UA. Esse programa possui um ciclo infinito e é caracterizado como um servidor socket serial. Isso é, ele só trata a próxima mensagem, quando termina de tratar a mensagem atual. Da mesma maneira, a Central executa um programa servidor, chamado `vbcsCentralServer`, que também foi desenvolvido com sockets, e aguarda as conexões dos ônibus ao socket específico do serviço de localização.

O processo de localização automática é iniciado pelo ônibus. No roteador sem-fio instalado no ônibus, um script busca continuamente pelas redes sem-fio criadas pelas UAs, se conectando à UA mais próxima, definida como aquela que possuir maior potência de sinal. Assim que o ônibus se conecta a uma UA, o script inicia um programa, `vbcsBusClientUA`, que se conecta ao socket da UA. Com isso, o programa servidor na UA, citado anteriormente, aceita a conexão e envia uma mensagem ao ônibus.

A mensagem enviada, mostrada na Figura 5.1, contém o identificador da UA. Neste trabalho, escolheu-se utilizar como identificador as coordenadas geográficas da UA, ou seja, longitude e latitude. No caso de um plano, como são as ruas, a latitude e a longitude identificam unicamente um objeto estático no espaço. Além disso, a localização passa a ter um significado mais próximo do geográfico, não sendo necessário verificar onde a UA está para localizar o ônibus.

```
<GTApfvbcsUFRJ>
    <UAID>
        #Identificador da Unidade de Acostamento
        #Latitude;Longitude
        -22.842709;-43.236139
    </UAID>
</GTApfvbcsUFRJ>
```

Figura 5.1: Mensagem enviada pela UA para o ônibus.

No sistema criado, o ônibus armazena tanto a informação da UA em que ele se encontra no momento, quanto a informação da última UA onde ele esteve. Sendo assim, quando o ônibus recebe a mensagem enviada pela UA, a informação antiga da UA atual é movida para o arquivo que representa a UA anterior, e o programa sobrescreve o arquivo da UA atual com a informação retirada da mensagem recebida da UA. Terminada a etapa anterior, o ônibus responde a UA com uma mensagem, mostrada na Figura 5.2. Essa mensagem contém as informações que identificam e caracterizam o ônibus, o identificador do ônibus em si e o identificador da linha de ônibus que ele está realizando atualmente. Posteriormente, a UA recebe a mensagem enviada pelo ônibus, armazenando-a em um arquivo de log.

```
<GTApfvbcsUFRJ>
    <BusLineID>
        #Identificador da Linha do Ônibus
        COPPEAD
    </BusLineID>
    <BusID>
        #Identificador do Ônibus
        BUS125
    </BusID>
</GTApfvbcsUFRJ>
```

Figura 5.2: Mensagem enviada pelo ônibus para a UA.

Assim que o programa `vbcsBusClientUA` termina sua execução, o script do ônibus inicia o programa `vbcsBusClientCentral`, que se conecta ao socket específico do serviço de localização da Central. Então, o programa presente na Central aceita a conexão, enviando, em seguida, uma mensagem para o ônibus. Essa mensagem contém apenas a data que a comunicação foi realizada, servindo para constituição de um histórico de comunicações entre o ônibus e a Central. Após receber e armazenar o conteúdo da mensagem no histórico, o ônibus envia uma mensagem, semelhante à mostrada na Figura 5.3, à Central. Essa mensagem contém todas as informações acerca do ônibus no momento, em qual UA o ônibus se encontra, a última UA que ele esteve, o identificador do ônibus e o identificador da linha de ônibus que ele está realizando. A Central ao receber essa mensagem, inicia o seu tratamento.

```
<GTApfvbcsUFRJ>
    <BusLineID>
        #Identificador da Linha do Ônibus
        COPPEAD
    </BusLineID>
    <BusID>
        #Identificador do Ônibus
        BUS125
    </BusID>
    <PrevUAID>
        #Identificador da UA anterior
        #Latitude;Longitude
        -22.840363;-43.237508
    </PrevUAID>
    <CurrUAID>
        #Identificador da UA atual
        #Latitude;Longitude
        -22.842709;-43.236139
    </CurrUAID>
</GTApfvbcsUFRJ>
```

Figura 5.3: Mensagem enviada pelo ônibus para a Central.

As seções a seguir dividem o tratamento das mensagens em função das estruturas presentes no programa e finalidade do tratamento. Apesar de não ser essencial para o serviço de localização, a criação e manutenção do mapa dinâmico das linhas de ônibus, assim como o cálculo do intervalo de tempo gasto para o ônibus percorrer o trecho entre duas UAs é realizado nessa parte. Para facilitar o entendimento, as principais estruturas presentes no programa da Central podem ser visualizadas, de forma simplificada, na Figura 5.4.

- Bus - cada objeto da classe Bus armazena as informações, identificador do ônibus, linha de ônibus que ele está servindo, UA onde o ônibus se encontra e última UA onde ele esteve, de um ônibus presente no sistema. A lista de objetos Bus contém as informações de todos os ônibus cadastrados no programa.

- BusLine - um objeto BusLine é a representação no programa de uma linha de ônibus real. As informações da linha de ônibus, como o mapa da linha de ônibus, identificador dela e o número de ônibus realizando-a, estão contidas nesse objeto. O mapa da linha de ônibus é representado pela lista de BusStops no interior do objeto BusLine. A lista de objetos BusLine armazena todas as

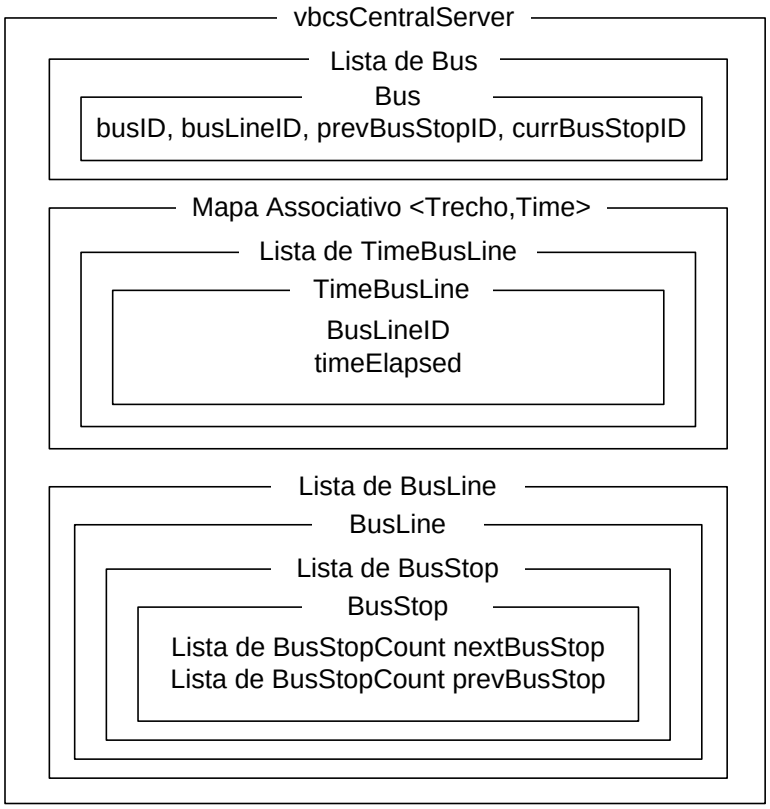

Figura 5.4: Estruturas principais simplificadas do programa da Central.

linhas de ônibus cadastradas no sistema.

- BusStop - cada objeto BusStop representa uma unidade de acostamento inserida no mapa de uma linha de ônibus, o que significa que uma UA física existirá mais de uma vez no programa se o mapa de mais de uma linha de ônibus contiver a UA. Cada objeto BusStop contém duas listas de objetos BusStopCount, sendo que uma delas indica os BusStops imediatamente após o atual no mapa da linha de ônibus e a outra os BusStops imediatamente antes do atual no mapa. Isso é necessário em situações como linhas com ciclo e mapas em processo de mudança, que será detalhado posteriormente.

- BusStopCount - cada objeto da classe BusStopCount contém um identificador, semelhante ao contido na classe BusStop; e um contador, que serve como peso. Os objetos da classe BusStopCount funcionam como arcos pesados, indicando objetos BusStop.

- Time - cada objeto Time está associado a um trecho UA de origem e UA de destino. Esse objeto contém uma lista de objetos TimeBusLine para armazenar os intervalos de tempo necessários para percorrer um dado trecho e uma variável que controla o número máximo de objetos nessa lista. A lista contém inclusive intervalos de tempo de linhas de ônibus diferentes que passem pelo mesmo trecho.

- TimeBusLine - um objeto TimeBusLine armazena a informação do intervalo de tempo necessário para percorrer um dado trecho e a linha de ônibus responsável por essa informação. A última informação serve para distinguir intervalos de tempo de linhas de ônibus diferentes que passem pelo mesmo trecho.

A princípio, pode não ser intuitivo a necessidade de uma lista de BusStopCount para representar os BusStops que vem antes e depois do atual, ao invés de apenas um ponteiro para o anterior e para o próximo. Contudo, deseja-se que o mapa da linha de ônibus seja gradualmente dinâmico. Isto é, o mapa não deve ser alterado instantaneamente a cada modificação que aparecer na linha de ônibus, pois podem haver mudanças devido a comportamento inadequado de um motorista, por exemplo. Esse dinamismo gradual exige que por certo período de tempo exista mais de uma possibilidade de BusStop anterior ou próximo na rota. Além disso, a estrutura em questão é absolutamente necessária para representar rotas de ônibus com ciclos, isto é, rotas que passam no mesmo ponto duas vezes.

5.1.1 Informação dos Ônibus

Para reunir as informações de todos os ônibus do sistema, uma lista de objetos da classe Bus existe no programa da Central. Cada objeto da classe Bus possui variáveis para armazenar as informações da mensagem mais recente recebida. Além disso, uma variável guarda a data que a última mensagem desse ônibus foi recebida e outra variável guarda a data que a penúltima mensagem foi recebida.

Quando uma mensagem proveniente de um ônibus é recebida, suas informações são extraídas e o programa busca na lista de objetos Bus pelo objeto que tem o mesmo identificador do ônibus que o contido na mensagem. Caso não encontre

nenhum objeto correspondente, um novo objeto Bus é criado e adicionado na lista contendo as informações da mensagem recebida e a data de recebimento. Porém, caso um objeto seja encontrado, o programa verifica possíveis situações que possam ter acontecido com o ônibus. Por exemplo, ele pode ter mudado de linha de ônibus ou ter saltado uma ou mais UAs. Essas verificações influenciam a criação do mapa das linhas de ônibus, e serão vistas posteriormente. Após as verificações, as informações do objeto encontrado são atualizadas com as da mensagem e as datas modificadas de acordo. Após esse processo, pode-se dizer que a lista de objetos da classe Bus armazena todas as informações necessárias para se localizar um ônibus.

5.1.2 Mapa das Linhas de Ônibus

Neste trabalho, como já foi dito, escolheu-se criar um mapa e mantê-lo dinamicamente. Em outras palavras, o sistema aprende a lista de pontos de ônibus de cada linha dinamicamente. Para isso ser possível, três classes são utilizadas: BusLine; BusStop; e BusStopCount.

Para facilitar o acompanhamento da descrição do procedimento de atualização dos mapas das linhas de ônibus, pode-se ver o fluxograma simplificado desse procedimento na Figura 5.5.

As verificações realizadas durante a atualização das informações dos ônibus influenciam nesse momento. As verificações indicam se o ônibus está sendo visto pela primeira vez, se ele mudou de linha em relação à última vez que foi visto ou se ele não conseguiu enviar a mensagem de localização em alguma UA anterior. Para saber se o ônibus mudou de linha, basta comparar o identificador da linha de ônibus contido na mensagem recebida com o contido no objeto Bus do ônibus em questão, antes da atualização. Para verificar se o ônibus foi capaz de enviar uma mensagem de localização à Central em um ponto anterior, deve-se comparar o identificador da UA anterior contido na mensagem com o identificador do BusStop atual do objeto Bus antes da atualização. Caso sejam diferentes, pode-se afirmar que alguma mensagem de localização não foi recebida pela Central.

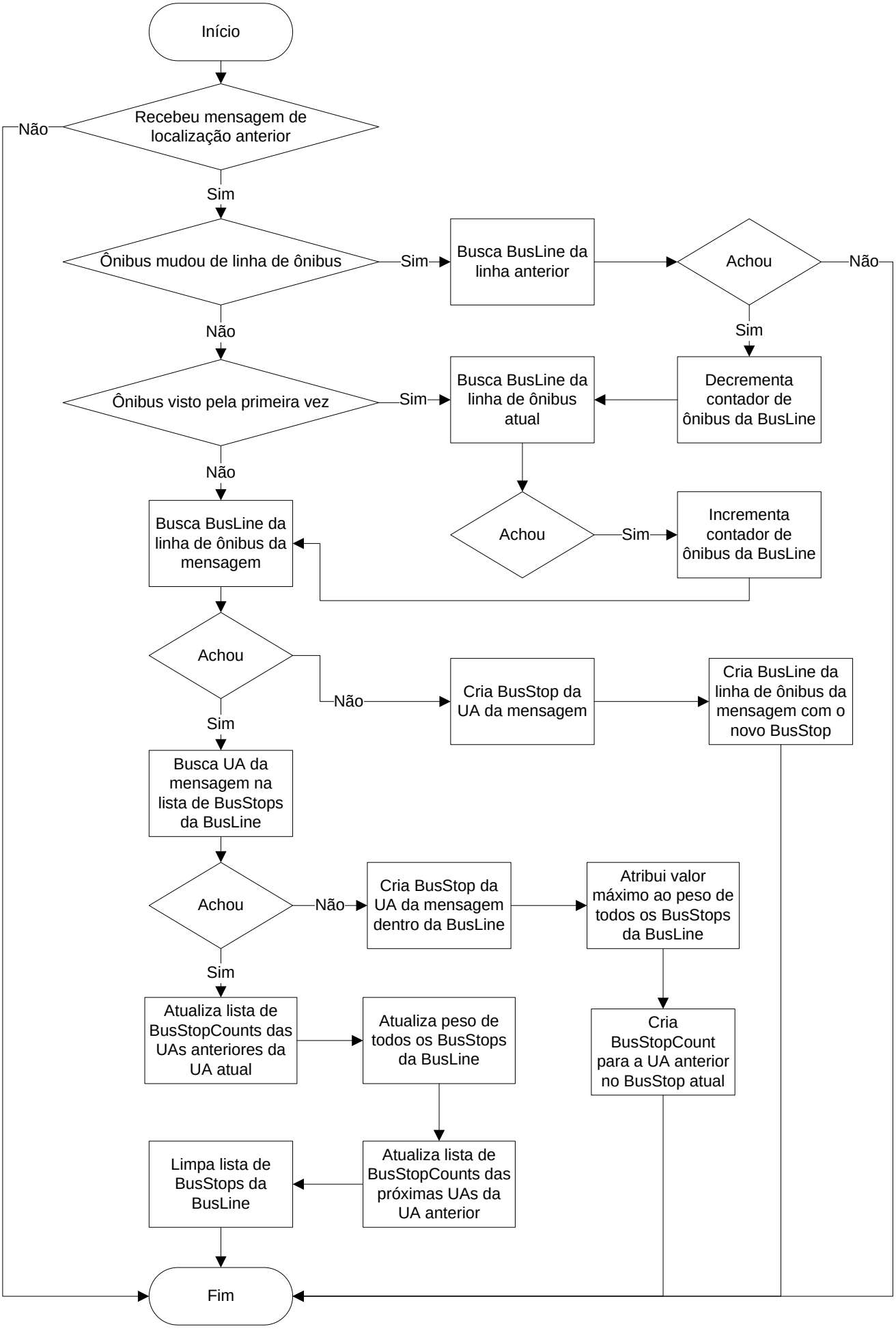

Figura 5.5: Fluxograma do procedimento de atualização dos mapas das linhas de ônibus.

Caso se verifique que a Central não recebeu alguma mensagem de localização anterior, o procedimento de atualização do mapa da linha não é executado e o tratamento da mensagem recebida termina. Caso contrário, diz-se que o ônibus age normalmente, o tratamento da mensagem recebida continua e o procedimento de atualização do mapa da linha será realizado.

Se tiver sido verificado que o ônibus mudou de linha, o programa procurará a linha de ônibus antiga na lista de BusLines e retirará uma unidade do contador de ônibus daquela linha. Em seguida, a nova linha de ônibus será procurada e, caso exista, uma unidade será adicionada ao contador de ônibus dessa linha. Da mesma maneira, se o ônibus estiver sendo visto pela primeira vez, a linha do novo ônibus será encontrada se existir, e uma unidade será adicionada ao contador de ônibus dessa linha.

Após as verificações anteriores e com o ônibus agindo normalmente, busca-se a linha de ônibus através do identificador contido na mensagem recebida. Não havendo nenhum objeto BusLine referente à linha especificada na mensagem, um objeto BusStop é criado e inserido na nova linha sendo criada. Esse BusStop representa o ponto de partida da linha de ônibus, e é criado a partir do identificador da UA atual contido na mensagem. Já no caso de a linha de ônibus ter sido encontrada, o programa irá procurar dentro da lista de BusStops, da BusLine encontrada, pela UA atual da mensagem.

Se a UA não for encontrada na lista, um objeto BusStop é criado para ela e inicializado com o peso máximo. O valor máximo do peso dos BusStops varia com o número de BusStops na linha, conforme mostrado na Equação 5.1:

$$maxWeightBusStop = 2 * qtBusStopInBusLine * qtBusInBusLine, \qquad (5.1)$$

onde maxWeightBusStop é o valor máximo permitido para o peso dos BusStops, qtBusStopInBusLine representa a quantidade de BusStops no objeto BusLine e qtBusInBusLine representa o número de objetos Bus percorrendo a BusLine. Então,

para ser justo com os BusStops adicionados anteriormente, decidiu-se que quando um novo BusStop for acrescentado em uma BusLine, após o peso máximo ser atualizado, o peso de todas as UAs cadastradas na linha é igualado a ele. Terminado o ajuste dos pesos dos BusStops, cria-se um objeto BusStopCount para referenciar a UA anterior da mensagem com peso do arco máximo, o peso do arco máximo é regido pela Equação 5.2:

$$maxWeightBusStopCount = 2 * qtBusStopInBusLine, \tag{5.2}$$

onde maxWeightBusStopCount é o valor máximo permitido para o peso do arco e qtBusStopInBusLine representa a quantidade de BusStops no objeto BusLine. Então, esse objeto BusStopCount é inserido na lista de UAs que aparecem antes da UA atual na rota, no interior do objeto BusStop.

Contudo, se a UA for encontrada na lista de BusStops da linha, a lista de BusStopCounts que representa os pontos anteriores ao BusStop é atualizada. Na atualização, subtrai-se uma unidade do peso do arco de todos os elementos da lista. Em seguida, somam-se duas unidades no peso do arco referente à UA anterior da mensagem recebida, se o valor após a soma for superior ao máximo, ele será igualado ao máximo. Caso algum objeto BusStopCount alcance peso do arco igual a zero após esse processo, ele será removido, significando que aquele BusStop não é mais utilizado como anterior do atual. O passo seguinte é a atualização do peso do BusStop. Nessa atualização, o peso de todos os BusStops da lista é decrementado de uma unidade. Após a subtração, soma-se ao peso do BusStop o valor da Equação 5.3:

$$weightBusStopIncrement = 2 * qtBusStopInBusLine, \tag{5.3}$$

onde weightBusStopIncrement é o valor do incremento somado ao peso do BusStop e qtBusStopInBusLine representa a quantidade de BusStops no objeto BusLine. Caso após a soma o valor do peso do BusStop ultrapasse o valor máximo estipulado pela Equação 5.1, o peso do BusStop será igualado ao máximo.

Como visto, dois tipos distintos de pesos são usados no trabalho, esses pesos têm funções diferentes. O peso do arco associado ao BusStopCount é o peso que rege a

dinamicidade do mapa, ou seja, quão rápido o mapa aceita uma modificação na rota como verdadeira e intencional. Esse peso no trabalho é uma constante independente que pode ser ajustada no código. Entretanto, o valor escolhido deve respeitar a restrição de ser no máximo duas vezes o número de ônibus servindo a linha, como visto na Equação 5.2. Essa restrição é necessária, pois em uma mudança de rota, obrigatoriamente, o peso do arco, que indica que o BusStop em processo de remoção faz parte da rota, deve alcançar o valor zero antes do peso do BusStop em remoção. Caso isso não seja respeitado, o objeto do BusStop em remoção é removido antes de ser retirado do mapa da linha de ônibus, gerando uma falha no programa. Nos testes, o valor usado foi o máximo permitido.

Já o peso dos BusStops é usado apenas para fins de liberação de memória, o que fica claro no exemplo de funcionamento de mudança de rota. O valor desses pesos deve ser grande o suficiente para que nenhum BusStop seja excluído antes da hora, o que levaria a inconsistências na rota de ônibus. Por isso, o valor inicial atribuído é o máximo contido na Equação 5.1. A ideia por trás do valor máximo desses pesos é que seria necessário que todos os ônibus realizando a linha percorressem todos os pontos da linha pelo menos duas vezes, para que um ponto não utilizado fosse excluído. Da mesma maneira, o incremento dado quando o BusStop é usado deve garantir que o BusStop utilizado não será excluído enquanto o ônibus percorre normalmente os outros BusStops da linha. Por isso, o incremento é função apenas do número de BusStops contido na linha de ônibus, como visto na Equação 5.3.

O próximo passo na atualização do mapa da linha é atualizar a lista de próximas UAs da UA anterior. Para isso, busca-se na lista de BusStops da BusLine pelo BusStop com identificador igual ao da UA anterior contido na mensagem. Quando o BusStop for encontrado, atualiza-se a lista de BusStopCounts que representa as próximas UAs do BusStop seguindo os mesmos critérios da atualização feita para a lista de UAs anteriores.

Após todas as atualizações nos pesos terem sido realizadas, o mapa da linha de ônibus está pronto. Porém, pode haver UAs não utilizadas na lista de UAs da linha de ônibus. É importante notar que apesar da existência das UAs não utilizadas, de

fato elas já não pertencem à rota da linha de ônibus, pois partindo do ponto inicial, elas não são alcançáveis. Para remover essas UAs inúteis, inicia-se o processo de limpeza da rota da linha de ônibus, ou seja, da lista de BusStops dentro da BusLine. Nesse processo, os BusStops podem ser eliminados por dois critérios, são eles:

1. O BusStop não é mais referenciado como anterior ou próximo BusStop por nenhum outro BusStop contido na linha.

2. O peso do BusStop alcançou o valor zero.

Inicialmente, utilizava-se apenas o primeiro critério, mas em casos onde várias UAs em sequência deixam de ser usadas na rota, essas UAs continuam se referenciando, não sendo possível removê-las pelo primeiro critério, por isso o segundo critério é necessário.

Realizados todos os procedimentos acima, o mapa da linha de ônibus é construído e atualizado dinamicamente, e as UAs não utilizadas são excluídas da lista das linhas de ônibus. Um exemplo de mapa construído pelo programa pode ser visualizado na Figura 5.6. Nas figuras que representam os mapas de linhas de ônibus desse projeto final, as circunferências representam as UAs, ou seja, os BusStops. O número no interior do retângulo em cada UA é o peso da UA usado em um critério de exclusão. Já os números nas setas representam os pesos dos arcos associados às informações de próxima UA e UA anterior da UA de origem da seta.

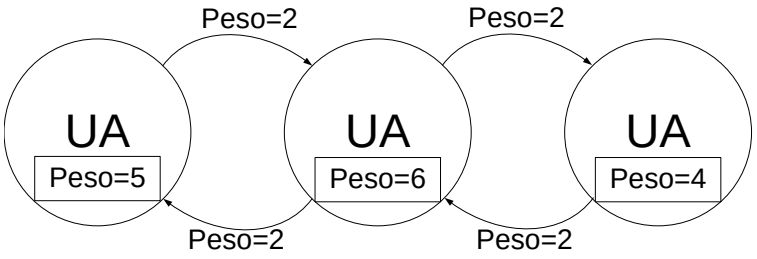

Figura 5.6: Visualização de mapa criado pelo programa.

5.1.3 Intervalos entre UAs

Os intervalos de tempo gastos entre duas UAs devem ser conhecidos para, posteriormente, as estimativas de previsão de chegada serem calculadas. Para criar esses tempos, são usadas as informações armazenadas nos objetos da classe Bus. Nesses objetos foram armazenados os horários de chegada das duas últimas mensagens, bem como os identificadores das duas últimas UAs que o ônibus passou. Porém, somente quando o ônibus estiver agindo normalmente, segundo as verificações já citadas, essas informações serão usadas para o cálculo do intervalo de tempo entre as UAs.

As classes Time e TimeBusLine são usadas para armazenar e lidar com os dados relativos às estimativas. No programa principal, uma estrutura map contém objetos da classe Time. Nessa estrutura, a concatenação dos identificadores das UAs do trecho separados por um asterisco é utilizada como índice identificador de cada objeto, seguindo a forma origem*destino. Por exemplo, o trecho da mensagem 5.3 é identificado como: -22.840363;-43.237508*-22.842709;-43.236139.

Pode-se observar que da maneira que a estrutura de armazenamento é construída todas as linhas que passam pelo mesmo par de UAs origem*destino têm seus tempos armazenados no mesmo objeto Time. Isso acontece pois os tempos são diferenciados quanto à linha de ônibus apenas no objeto TimeBusLine. A estrutura de armazenamento dos tempos é construída dessa forma para que a média dos intervalos gastos no trecho possa ser calculada de duas formas distintas. Uma das formas de cálculo utiliza as informações de todas as linhas de ônibus que passam no trecho enquanto a outra usa apenas as informações pertencentes à linha de ônibus sendo estimada. Os dois métodos para o cálculo de média estão presentes no objeto Time, e são discutidos na seção de estimativas.

O cálculo do tempo gasto para percorrer o trecho consiste em subtrair os dois horários contidos no objeto Bus, obtendo como resultado o intervalo de tempo gasto entre as duas UAs. Após a subtração, cria-se um objeto TimeBusLine com o tempo calculado e com o identificador da linha de ônibus do objeto Bus. Em seguida, o identificador do trecho é criado usando os identificadores das UAs anterior e atual

contidas no objeto Bus. Então, realiza-se uma busca no map pelo objeto Time correspondente ao identificador criado. Caso o objeto do trecho não exista na estrutura, ele é inicializado com o primeiro TimeBusLine inserido e incluído no map. Entretanto, caso o objeto do trecho exista, o objeto TimeBusLine criado é inserido no final da lista de TimeBusLines do objeto Time encontrado. Caso o número de elementos nessa lista ultrapasse o máximo permitido pela variável de controle do objeto Time, remove-se o primeiro elemento, como uma FIFO. A lista ser FIFO garante que os elementos existentes nela são sempre os mais recentes. A exclusão leva em consideração o número de elementos totais e não o número de elementos de cada linha de ônibus. Com isso, é possível que uma linha de ônibus monopolize os tempos gastos armazenados para um determinado trecho. Por exemplo, caso duas linhas de ônibus com números diferentes de ônibus compartilhem um determinado trecho em suas rotas. Para reduzir o problema, fez-se o tamanho máximo da lista ser variável e proporcional ao número de linhas de ônibus total no programa, como mostrado na Equação 5.4:

$$maxTimeBusLineListSize = qtBusLine * K, \tag{5.4}$$

onde maxTimeBusLineListSize é o tamanho máximo da lista de objetos TimeBusLine, qtBusLine representa o número de objetos BusLine existentes no sistema e K é um parâmetro que controla a proporção entre as variáveis maxTimeBusLineListSize e qtBusLine. Neste projeto K foi ajustado para 10, podendo ser alterado se necessário.

Não impedir um monopólio está associado à ideia de se armazenar as informações mais recentes para os trechos. Contudo, o monopólio pode ser maléfico caso existam linhas de ônibus que passam pelas mesmas duas UAs seguidas, caracterizando o trecho origem*destino, mas por caminhos geográficos muito distintos. Esse problema é um problema de amostragem. A solução proposta é a inserção entre as UAs que o problema ocorre de uma nova UA na rota da linha de ônibus que percorre o maior caminho do trecho.

5.2 Previsão de Chegada de Veículos

A previsão de chegada de veículos é um serviço de ITS oferecido aos usuários dos meios de transporte públicos. De posse do mapa da linha de ônibus e da informação dos tempos gastos pelos ônibus entre duas UAs, pode-se calcular o tempo que um ônibus leva para alcançar os pontos seguintes da sua rota. Esse serviço é oferecido pela Central aos Clientes.

5.2.1 Página de Requisição de Estimativas

O Cliente precisa informar uma linha e um ponto de ônibus para receber uma estimativa de hora de chegada do próximo ônibus. Para esse fim, criou-se uma página na Internet que pode ser vista na Figura 5.7.

Na página criada, o Cliente deve selecionar uma das linhas de ônibus disponíveis no sistema. Em seguida, o Cliente deve clicar em uma das figuras dos pontos de ônibus no mapa. Com isso, o ponto de ônibus selecionado ficará vermelho, indicando a escolha feita, como pode ser visto na Figura 5.8. Após o Cliente selecionar as informações necessárias, ele deve clicar no botão "Estimar". Caso o Cliente se esqueça de selecionar o ponto ou a linha de ônibus, uma janela o informa e pede para que os campos sejam preenchidos corretamente. Caso contrário, o Cliente é direcionado para uma página onde ele recebe a resposta à sua requisição, como pode ser visualizado na Figura 5.9.

A página onde o Cliente recebe a resposta possui um código em PHP que cria uma mensagem similar a da Figura 5.10. Em seguida, a página envia a mensagem criada para o socket específico do serviço de estimativa da Central. A mensagem contém as informações da requisição do usuário: linha e ponto de ônibus escolhidos. A Central recebe o pedido e responde com a estimativa desejada via socket.

5.2.2 Cálculo de Estimativas

O cálculo das estimativas é realizado após o serviço de localização automática de veículos ter terminado o tratamento da mensagem recebida. É importante notar que apesar do serviço de previsão depender do Cliente, o cálculo das estimativas é

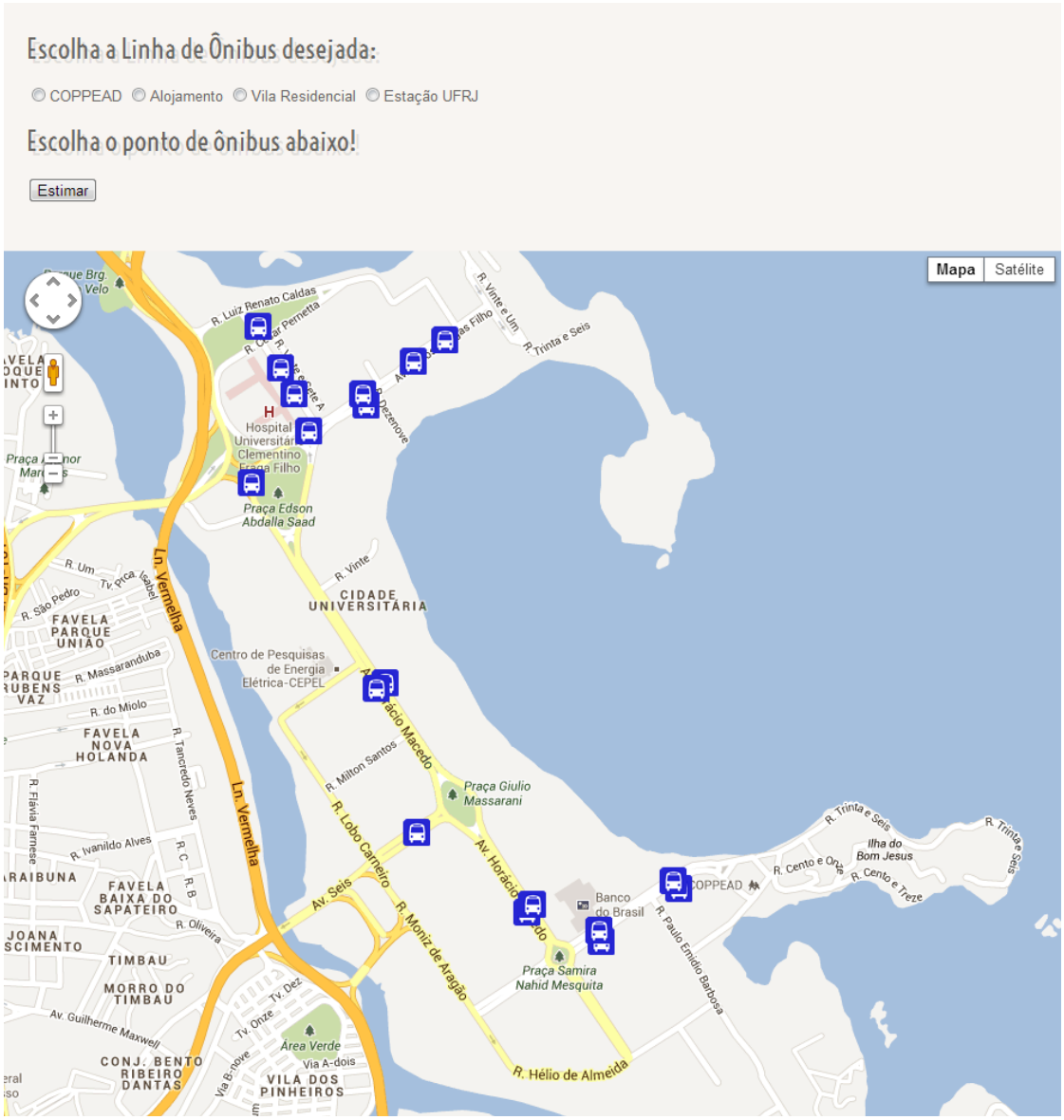

Figura 5.7: Página de solicitação de estimativa.

independente das requisições do Cliente. Essa escolha é feita para evitar a sobrecarga do processador. A sobrecarga aconteceria caso os cálculos fossem efetuados a cada requisição de Cliente e muitos Clientes pedissem estimativas simultaneamente. O resultado caso a sobrecarga acontecesse seria uma espécie de negação de serviço.

Toda vez que a Central recebe uma mensagem, após o serviço de localização automática de veículos, a linha de ônibus identificada na mensagem tem as estimativas calculadas para todos os BusStops. A única restrição para o cálculo das estimativas ser realizado é que a linha de ônibus em questão deve ter os pontos inicial e final diferentes, isso é, não deve ser uma linha circular. Isso se deve pois para o caso de

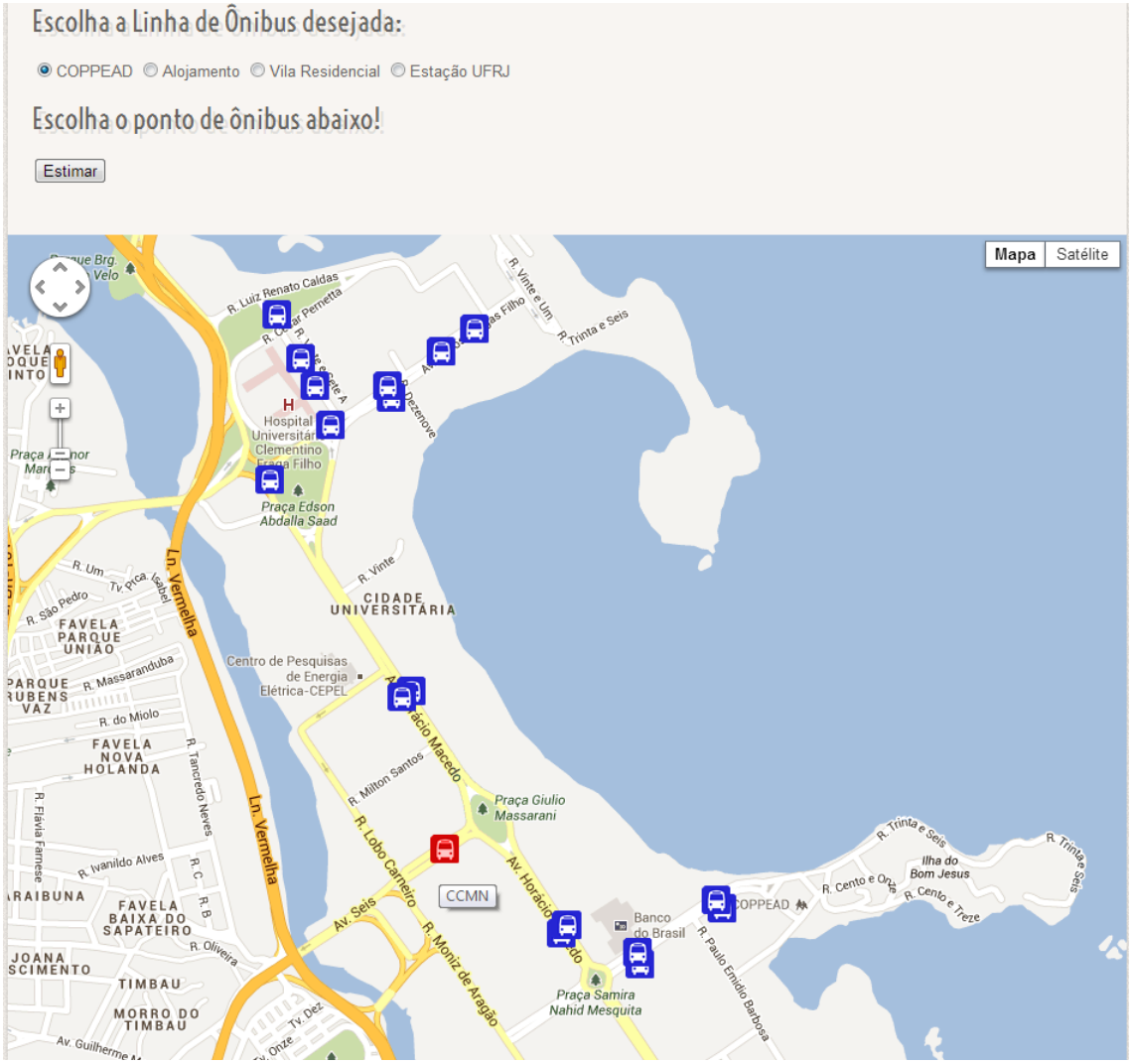

Figura 5.8: Ponto de ônibus e linha de ônibus selecionados.

linhas circulares o algoritmo utilizado no cálculo de estimativas segue o mapa da linha de ônibus eternamente. Entretanto, para usar o serviço em casos de linhas circulares basta dividir a linha de ônibus em duas. Por exemplo, de acordo com o sentido que a linha está sendo percorrida em relação a algum ponto de referência, como, em direção a qual bairro o ônibus está indo. Apesar de linhas circulares não segmentadas não serem estimadas, linhas que contenham ciclos em seu interior são estimadas normalmente.

Além das variáveis descritas na seção anterior, a classe BusStop possui ainda quatro variáveis para armazenar as estimativas de tempo. Cada variável guarda estimativas calculadas de maneiras diferentes. As quatro estimativas são resul-

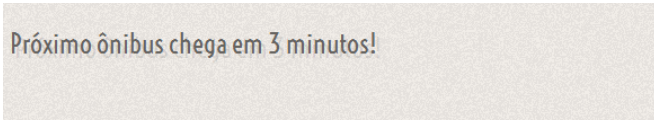

Figura 5.9: Estimativa devolvida ao Cliente.

```
<GTApfvbcsUFRJ>
    <ReqBusLineID>
        #Identificador da Linha do Ônibus requisitada
        COPPEAD
    </ReqBusLineID>
    <ReqUAID>
        #Identificador da UA requisitada
        #Latitude;Longitude
        -22.842709;-43.236139
    </ReqUAID>
</GTApfvbcsUFRJ>
```

Figura 5.10: Mensagem de requisição enviada pelo Cliente para a Central.

tado da existência de duas formas diferentes de calcular as médias dos trechos origem*destino. Uma das formas calcula a média de todos os elementos TimeBusLine da lista contida na classe Time do trecho, não importando se os tempos usados na média pertencem à mesma linha de ônibus. A outra forma calcula a média considerando apenas os objetos TimeBusLine que pertencem à linha de ônibus sendo estimada.

A primeira forma é necessária pois podem existir situações onde os tempos da linha de ônibus sendo estimada são antigos. Com isso, eles já não representam as atuais condições de trânsito do trecho, o que resulta em uma estimativa errada. Como a primeira forma não restringe os tempos usados na média, misturando tempos de diferentes linhas, o uso de tempos mais recentes no cálculo da média é privilegiado. Caso o problema, já citado, de duas rotas que passam pelas mesmas UAs em sequência, mas por caminhos muito discrepantes, não ocorra, esse método pode ser usado sem nenhuma ressalva.

Com os dois tipos de média citados para cada trecho, escolheu-se incrementar as estimativas de quatro maneiras distintas para cada BusStop. A primeira maneira utiliza como incremento entre a estimativa do ponto anterior e do ponto atual a média apenas dos intervalos de tempo do trecho que pertençam à própria linha. A segunda utiliza a média de todos os tempos no trecho como incremento. Já a terceira usa como incremento o valor mínimo entre as duas médias anteriores, enquanto a quarta usa como incremento o valor máximo entre elas. O uso da terceira maneira resulta em uma estimativa otimista do tempo de chegada do próximo ônibus, ou seja, resulta no menor tempo possível que o sistema prevê a chegada do próximo ônibus. Por outro lado, o uso da quarta maneira resulta em uma estimativa pessimista do tempo de chegada do próximo ônibus, ou seja, resulta no maior tempo possível que o sistema prevê a chegada do próximo ônibus. No sistema, as quatro maneiras de incrementar as estimativas são realizadas. A escolha da maneira de estimar que será enviada aos usuários fará parte de um trabalho futuro.

No cálculo da estimativa, busca-se o menor tempo necessário para que um ônibus qualquer da linha sendo estimada alcance cada um dos pontos do mapa. Como condição inicial, todas as variáveis de estimativa de todos os BusStops armazenados no mapa da linha são igualadas a um valor de referência negativo. Em seguida, para cada ônibus realizando a linha, procura-se o BusStop onde ele foi visto pela última vez, e faz-se com que esse BusStop receba o valor de estimativa zero. A seguir inicia-se um procedimento que simula o percurso do ônibus pela linha, a partir do ponto onde o ônibus estava.

Devido à forma como o mapa é construído, escolheu-se fazer o procedimento que percorre a linha ser recursivo. O procedimento desenvolvido tem como única condição de parada que o ponto sendo avaliado não possua nenhum próximo ponto cadastrado.

Durante o percurso, um BusStop em análise pode ter mais de um BusStop próximo possível. Isso pode ocorrer devido a uma mudança de rota em andamento, como pode ser visto na Figura 5.11, ou devido à rota possuir um ciclo, como na Figura 5.12. No exemplo da Figura 5.11, o ônibus que normalmente percorre a rota UA-1, UA-2,

UA-3, UA-4 e UA-5 começa a realizar uma nova rota, onde vai diretamente da UA-2 para a UA-5. Já no exemplo da Figura 5.12, o trajeto da linha de ônibus é UA-1, UA-2, UA-3, UA-4, UA-2, UA-3 e UA5. O fato de um BusStop em análise poder ter mais de um BusStop próximo possível exige que um mecanismo de controle exista. Esse mecanismo deve existir para que o algoritmo que percorre a linha decida qual dos próximos pontos é o seguinte da rota. O mecanismo de controle deve funcionar tanto para o caso de uma mudança de rota em curso quanto para o caso de uma linha com ciclos na rota.

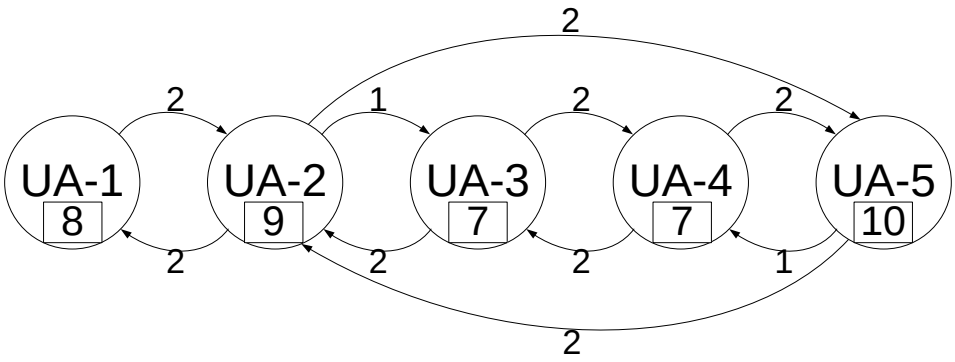

Figura 5.11: Mudança de rota em curso.

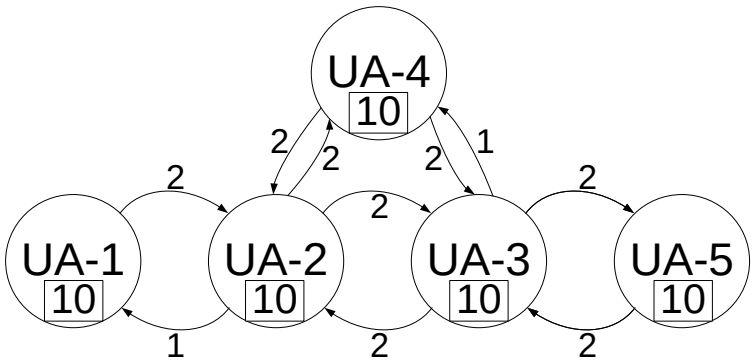

Figura 5.12: Linha com ciclo.

O mecanismo de controle só atua caso o BusStop em análise possua mais de um próximo BusStop. Para definir se o ônibus percorrendo a linha está passando no BusStop pela primeira ou segunda vez, cria-se uma lista dentro de cada objeto BusStop. Assim que um ônibus chega ao BusStop em questão, verifica-se se o identificador dele existe na lista do BusStop. Se não existir, o identificador do

ônibus é inserido na lista. Nesse caso o ônibus está passando pela primeira vez no BusStop. Já caso o identificador exista na lista, retira-se o identificador da lista para restaurar a configuração inicial. Nesse caso o ônibus está passando pela segunda vez no BusStop. No primeiro caso, o ônibus é direcionado para o primeiro BusStop cadastrado na sua lista de próximos BusStops. Já no segundo caso, o ônibus é direcionado para o segundo BusStop cadastrado em sua lista de próximos.

Nos casos de mudança de rota que não constituem um ciclo, como no exemplo da Figura 5.11, o mecanismo de controle sempre indica para o ônibus seguir para o primeiro BusStop cadastrado na lista de próximos. Isso ocorre pois para esse caso em um mesmo cálculo de estimativa, o mesmo ônibus só passa por esses BusStops uma vez. Nos casos de rotas com ciclo, como o exemplo da Figura 5.12, o mecanismo faz o ônibus percorrer a rota corretamente. Com isso, o algoritmo desenvolvido soluciona o problema de definir qual o próximo BusStop para ambos os casos problemáticos.

Após a definição do próximo BusStop, o identificador origem*destino do trecho é gerado. Em seguida, busca-se por esse identificador na estrutura que armazena os intervalos de tempo dos trechos, encontrando as informações de intervalo de tempo do trecho.

De posse das informações, os dois tipos de médias citados anteriormente são realizados. Contudo, as quatro estimativas do próximo BusStop só são atualizadas caso os valores das estimativas do BusStop atual somados aos incrementos do trecho sejam menores que os valores das estimativas presentes no próximo BusStop. Isso é feito para garantir que as variáveis de estimativa guardem os tempos necessários para a chegada do ônibus mais próximo, ou seja, as estimativas mínimas.

Toda vez que as variáveis de estimativas do próximo BusStop são atualizadas, é modificada também uma variável que indica qual o ônibus mais perto do BusStop seguinte até o momento. Em seguida, o próprio procedimento que simula o percurso do ônibus é chamado, indicando como BusStop atual o próximo BusStop. Isso simula o movimento do ônibus na rota.

Após todos os ônibus terem o percurso simulado, recalcula-se as estimativas dos BusStops que possuem estimativa igual a zero. Isso é necessário pois a estimativa ser igual a zero significa que o ônibus acabou de passar ou que ele está nesse exato momento no ponto de interesse. Essas informações não são relevantes para o Cliente. Então, as estimativas desses BusStop são modificadas de maneira a conter as estimativas para o próximo ônibus alcançá-lo. Isso é realizado, retornando pela rota até os BusStops anteriores a eles e somando as estimativas dos anteriores com as médias do respectivo trecho BusStop anterior BusStop atual. Os resultados dos cálculos são armazenados como as novas estimativas dos BusStops que tinham estimativa zero.

5.2.3 Resposta aos Clientes

No programa que executa na Central, há um thread responsável pelo servidor socket que recebe as requisições de estimativas dos Clientes. Esse servidor é semelhante ao do recebimento de mensagens, todavia utiliza outra porta. Quando uma mensagem de requisição é recebida, as informações contidas nela, ponto de ônibus alvo e linha de ônibus desejada, são retiradas. Em seguida, busca-se o objeto correspondente à linha de ônibus requisitada na lista de linhas de ônibus do programa. Caso esta não exista uma mensagem é enviada indicando o problema ao Cliente. De posse do objeto da linha de ônibus, o ponto de ônibus é buscado no mapa da linha. Caso não seja encontrado, uma mensagem é enviada ao Cliente indicando que a rota não passa pelo ponto indicado. Caso o ponto de ônibus seja encontrado no mapa da linha, obtém-se a estimativa do ponto em questão. Em seguida, a estimativa é atualizada e, após a atualização, enviada ao Cliente através de uma mensagem.

A referida atualização é necessária pois, como dito anteriormente, o cálculo das estimativas é independente das requisições dos Clientes. Portanto, uma vez que as estimativas de uma linha de ônibus tenham sido calculadas, elas somente são recalculadas quando um ônibus daquela linha enviar outra mensagem. Ou seja, as estimativas presentes nos BusStops não decrescem com o tempo, mas somente com novas mensagens de ônibus da linha. Isso significa que as estimativas caem em saltos ao invés de continuamente, o que pode ser ruim para o Cliente.

Para que as estimativas caiam continuamente para o Cliente, guarda-se no momento do cálculo das estimativas, a data que elas foram alteradas. As estimativas só são alteradas de fato nos BusStops em que o ônibus mais próximo é também o ônibus que enviou a mensagem que iniciou o cálculo das estimativas. Então, usando a data de alteração da estimativa presente no BusStop, e a data de recebimento da requisição, calcula-se o tempo decorrido entre as duas situações. Em seguida, o tempo decorrido é subtraido da estimativa do BusStop, antes que esta seja enviada ao Cliente. Com isso o Cliente enxerga a estimativa decrescendo continuamente, o que lhe é mais útil e agradável.

5.3 Linguagens e Bibliotecas

Todos os programas usados nos roteadores, seja nas Unidades de Acostamento ou nos ônibus, são escritos em C, e a compilação desses programas é realizada através de compilação cruzada, ou seja, é gerado código executável para um processador diferente daquele em que a compilação era realizada. Compila-se em um PC código a ser executado em um processador ARM, presente nos roteadores sem-fio utilizados. O procedimento para realizar essa compilação encontra-se em [31].

A Central é escrita parte em C e parte em C++. Isso ocorre pois o C++ não lida diretamente com sockets como o C faz. Por isso o C é usado somente na parte de manipulação de sockets e no thread responsável por responder as requisições de estimativa do Cliente, enquanto o C++ é utilizado no restante do programa.

As páginas acessadas pelo Cliente são desenvolvidas em PHP com parte em Javascript. Além disso, é utilizada a API versão 3 do Google Maps [32] para o usuário visualizar no mapa da cidade os pontos de ônibus cadastrados no sistema. Isso é feito para facilitar ao Cliente a escolha do ponto de ônibus. O PHP é usado devido à sua semelhança com o C e o suporte a sockets. Já o Javascript é usado devido à API do Google Maps ser em Javascript.

5.4 Exemplos do Funcionamento do Sistema

Nessa seção dois exemplos do funcionamento do sistema são apresentados passo a passo. Nas figuras dessa seção, a linha da circunferência que representa uma UA estar mais espessa significa que há um ônibus nessa UA no momento.

5.4.1 Construção Inicial de uma Linha de Ônibus

Nesse teste, considerou-se uma linha fictícia de ônibus que realiza uma rota percorrendo UAs numeradas de um a cinco em ordem ascendente. O ponto inicial da rota é a UA-1 e o ponto final a UA-5. Para facilitar a compreensão, apenas um ônibus percorre a rota sendo construída, mas essa limitação não existe na aplicação desenvolvida.

No início, a linha de ônibus não existe. A linha de ônibus, e consequentemente seu mapa, só é criada quando, ao menos uma vez, um ônibus informa à Central que está realizando aquela linha. Portanto, quando a Central recebe a mensagem de uma dada linha pela primeira vez, ela cria essa linha. A linha ao ser criada já contém a UA onde o ônibus se encontra como UA inicial da linha de ônibus. A UA inicial recebe o peso máximo, cujo valor pode ser obtido pela Equação 5.1; ou seja, duas vezes o número de UAs no mapa da linha de ônibus multiplicado pelo número de ônibus servindo a linha de ônibus. O resultado da operação descrita pode ser visualizado na Figura 5.13.

Figura 5.13: Criação do mapa da linha com primeira UA.

O próximo passo na construção da rota é tomado quando o ônibus alcança a próxima UA da rota da linha de ônibus. Ao alcançar a próxima UA da rota o ônibus envia uma mensagem à Central. A Central ao receber essa mensagem, verifica que a linha de ônibus já existe, mas que a UA informada pelo ônibus ainda não existe

na rota. A Central, então, cria um objeto associado a essa nova UA e o insere na rota. Ao uma nova UA ser adicionada na rota, os pesos de todas as UAs da rota são atualizados para o novo valor máximo, dado pela Equação 5.1, já considerando a nova UA na rota. Como a UA-2 não é a primeira da linha, a Central associa a UA-2 como próxima UA da rota em relação à UA inicial (UA-1). Da mesma maneira, a Central indica na UA-2 que a UA-1 vem antes dela na rota. Essas informações são inicializadas com peso do arco igual a dois, valor obtido através da Equação 5.2. O mapa após essas operações pode ser visto na Figura 5.14.

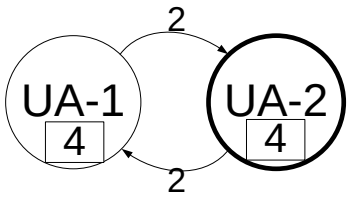

Figura 5.14: Mapa após inserção da segunda UA.

Em seguida, o ônibus alcança a terceira UA da rota, avisando a Central. A Central verifica que a UA informada não existe na rota, precisando ser criada e inserida. Com isso, a Central cria um objeto associado a essa nova UA e o insere na rota. Como já visto anteriormente, os pesos de todas as UAs são atualizados segundo a Equação 5.1 já considerando a nova UA na rota. Em seguida, a Central associa a nova UA como próxima UA da UA anterior. Da mesma forma, a Central indica na nova UA que a segunda UA é anterior a ela. Essas informações são inicializadas com valor dado pela Equação 5.2. O mapa atualizado pode ser visto na Figura 5.15.

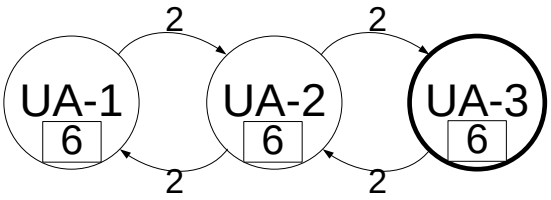

Figura 5.15: Mapa após inserção da terceira UA.

O processo descrito anteriormente é repetido para as quarta e quinta UAs da rota. Com os pesos das UAs sendo alterados e as informações de próxima UA e UA

anterior das UAs sendo preenchidas toda vez que uma nova UA é inserida na rota. O mapa após a inserção da quarta UA pode ser visualizado na Figura 5.16, e após a inserção da quinta e última UA da linha na Figura 5.17.

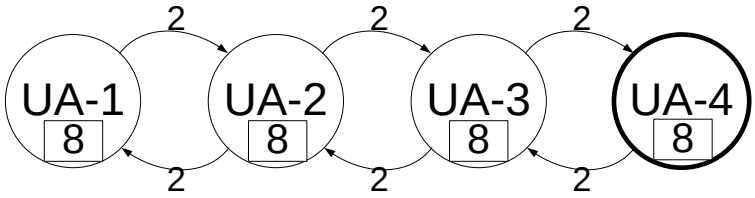

Figura 5.16: Mapa após inserção da quarta UA.

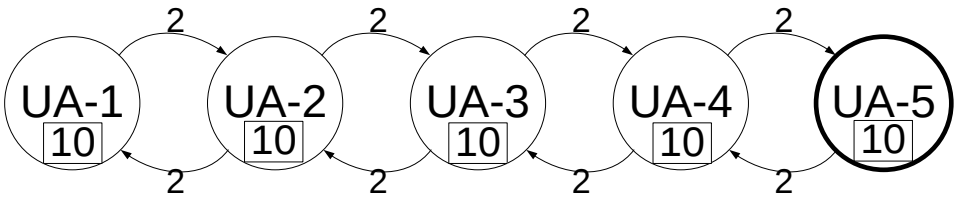

Figura 5.17: Mapa final da linha de ônibus.

5.4.2 Mudança de Rota

O exemplo de funcionamento atual demonstra como o sistema lida com mudanças na rota de uma linha de ônibus. Mudanças de rota podem ser devido a diversos fatores, por exemplo, vias interditadas ou mau funcionamento de UAs na rota. Esses eventos podem gerar uma diminuição do número de UAs na rota ou um aumento de UAs. Não importa se a rota aumenta ou diminui, o sistema lida com a mudança das rotas da mesma forma, alterando as informações de próxima UA e UA anterior das UAs da linha.

Neste exemplo de funcionamento é demonstrado o exemplo em que a rota da linha de ônibus diminui. Fez-se essa escolha pois ela permite demonstrar a necessidade da existência dos dois tipos de peso. Um para controlar a dinamicidade do mapa e outro para controlar a exclusão de UAs da rota. A dinamicidade do mapa é controlada pelo peso dos arcos e a exclusão de UAs da rota é controlada pelo peso das UAs.

Neste exemplo, considerou-se a linha fictícia de ônibus criada no exemplo anterior. Apenas um ônibus percorre a rota sendo modificada durante o teste. O teste se inicia com o ônibus se movendo até a UA-1. Isso atualiza o peso da própria UA para o máximo, e reduz em uma unidade o peso de todas as outras UAs da linha. Essa situação se encontra na Figura 5.18.

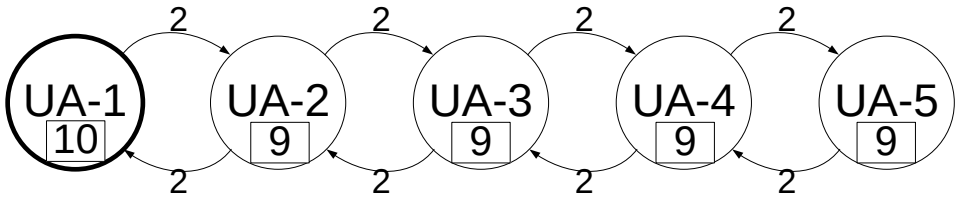

Figura 5.18: Rota inicial da linha de ônibus.

Em seguida, o ônibus se movimenta até a UA-2, o que não causa alteração nos pesos dos arcos entre as UAs 1 e 2, que continuam com valor máximo permitido. Entretanto, o peso de todas as outras UAs da linha de ônibus é decrementado de uma unidade, enquanto o peso da UA-2, onde o ônibus se encontra, é incrementado de acordo com a Equação 5.3. Após o incremento o peso da UA-2 atinge o peso máximo permitido, regido pela Equação 5.1. O mapa atualizado pode ser visto na Figura 5.19.

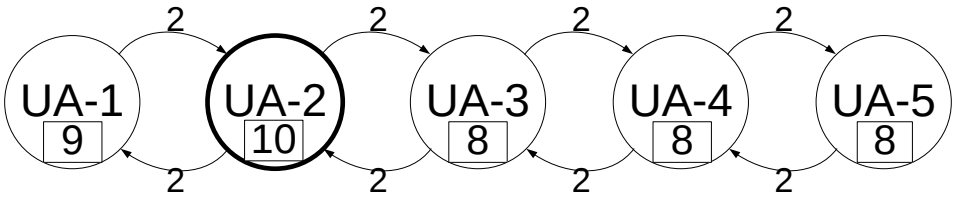

Figura 5.19: Ônibus se move até a UA-2.

Se a rota fosse seguida normalmente, o próximo passo seria o ônibus ir até a UA-3. Todavia, nesse exemplo de funcionamento, supõe-se que as UAs 3 e 4 estão defeituosas. Com isso, a Central não recebe as mensagens referentes a esses dois pontos de acesso. Portanto, a próxima UA funcionando na rota é a UA-5. Ao chegar na UA-5, o ônibus envia uma mensagem à Central, informando que ele está

lá e que antes esteve na UA-2. Quando a Central recebe essa mensagem, ela atualiza os pesos de todas as UAs da linha.

Após a atualização dos pesos das UAs da linha, a UA-5 é cadastrada na lista de próximas UAs da UA-2. Da mesma maneira, a UA-2 é cadastrada na lista de UAs anteriores da UA-5. Ambos os arcos são inicializados com peso igual a Equação 5.2. Em seguida, as outras informações da lista de próximas UAs da UA-2 e da lista de UAs anteriores da UA-5 são decrementadas de uma unidade. O resultado desse processo pode ser observado na Figura 5.20.

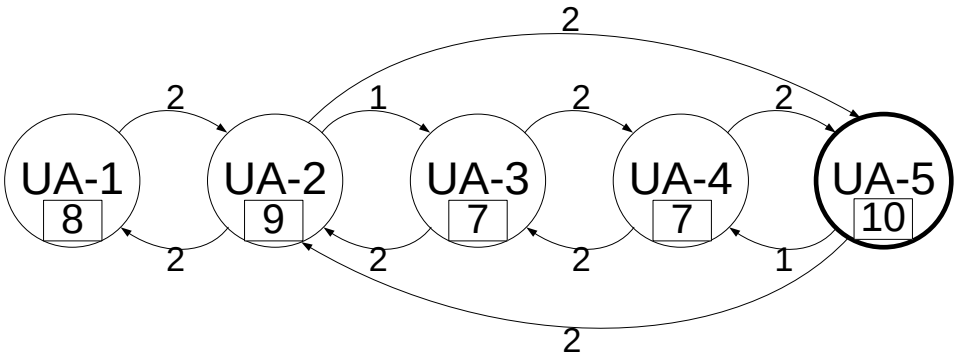

Figura 5.20: Ônibus alcança a UA-5.

Continuando o exemplo, o ônibus percorre a rota alcançando a UA-1 e depois a UA-2. Com isso, as atualizações nos pesos das UAs da linha são realizadas. As alterações no mapa podem ser observadas nas Figuras 5.21 e 5.22, respectivamente. Pode-se notar na Figura 5.22 que os pesos das UAs 3 e 4 já estão na metade do peso máximo permitido na linha de ônibus.

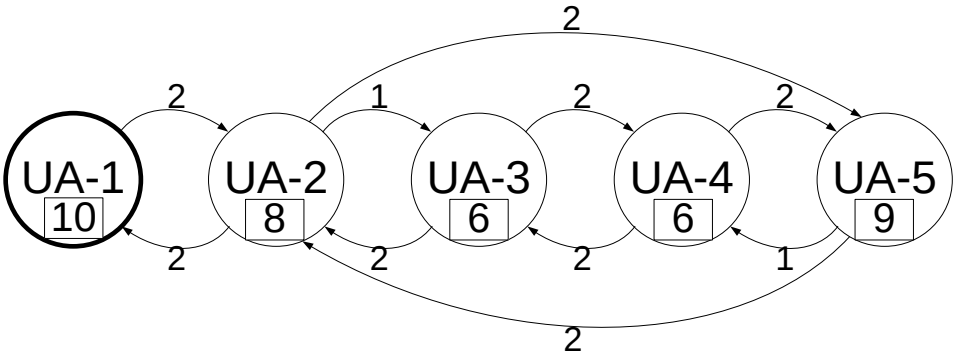

Figura 5.21: Ônibus retorna ao ponto inicial.

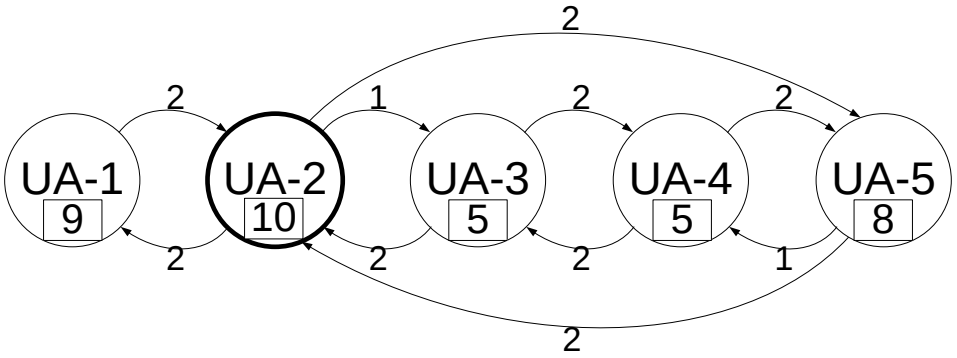

Figura 5.22: Ônibus chega na UA-2.

No momento que o ônibus se aproxima da UA-5 novamente, a atualização nos pesos se encarrega de decrementar novamente os pesos de todas as UAs. Em seguida o peso da UA-5 é incrementado e atinge o máximo permitido. Realizada a atualização dos pesos das UAs, o sistema verifica que, novamente, a UA anterior a UA-5 é a UA-2. Por consequência a UA seguinte a UA-2 é a UA-5. Com isso, o peso desses arcos se mantém, enquanto o peso dos outros arcos decresce novamente de uma unidade.

Quando os arcos que indicam que a próxima UA da UA-2 é a UA-3 e que a anterior a UA-5 é a UA-4, são decrementados, eles atingem o valor zero. Por isso, esses dois arcos são excluídas da lista. A exclusão desses arcos representa a concretização da modificação da rota. Isso pode ser afirmado pois apesar de as UAs 3 e 4 existirem na lista de UAs da linha, elas já não são alcançáveis a partir do ponto inicial da linha de ônibus. As modificações realizadas podem ser vistas na Figura 5.23.

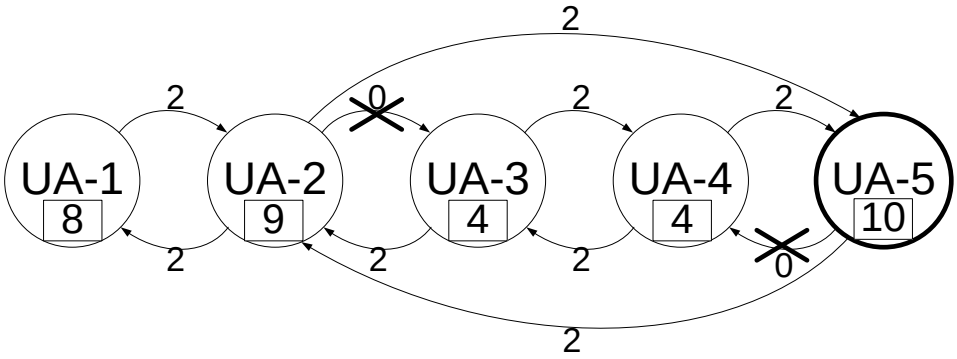

Figura 5.23: Ônibus se move até UA-5. E a mudança na rota é concretizada.

Com a rota do ônibus concretizada, não ocorre mais nenhuma alteração nas informações de próxima UA e UA anterior de nenhuma UA da rota. Todavia, o ônibus continua realizando a sua rota normalmente, e os pesos das UAs da linha são atualizados de acordo. Na Figura 5.24 está o resultado da movimentação do ônibus até a UA-1, na Figura 5.25 o resultado da continuação do movimento até a UA-2 e na Figura 5.26 o mapa resultante do movimento até o ponto final da rota.

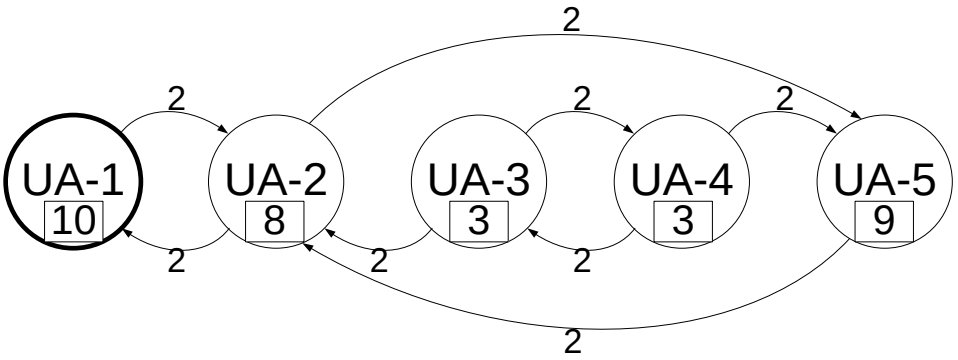

Figura 5.24: Ônibus na UA-1.

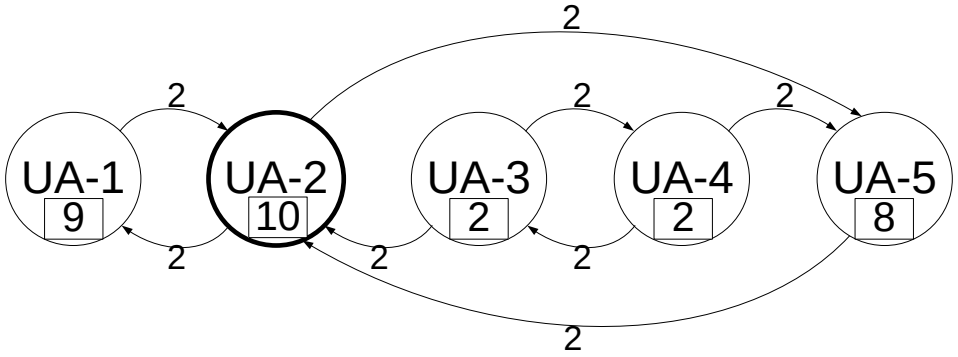

Figura 5.25: Ônibus na UA-2.

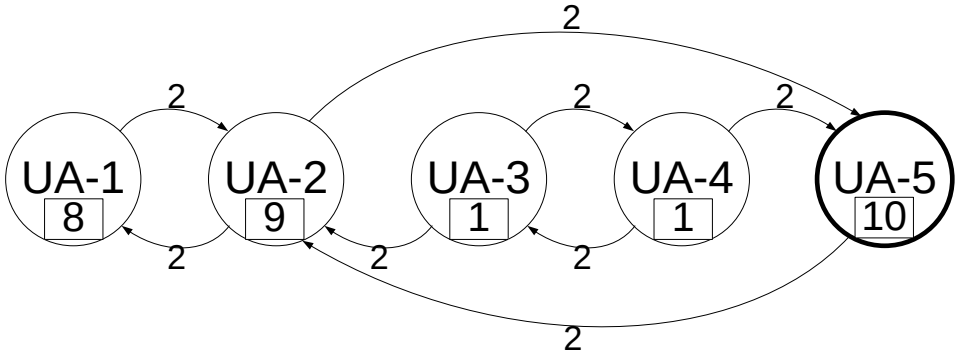

Figura 5.26: Ônibus na UA-5.

Ao continuar o seu trajeto, o ônibus chega na UA-1, e os pesos das UAs 3 e 4 ao serem subtraídos chegam a zero. Isso faz com que o mecanismo de limpeza da linha de ônibus exclua esses objetos da lista de UAs da linha. Portanto, pôde-se observar a função dos dois tipos de peso na aplicação criada. O processo descrito pode ser visto na Figura 5.27.

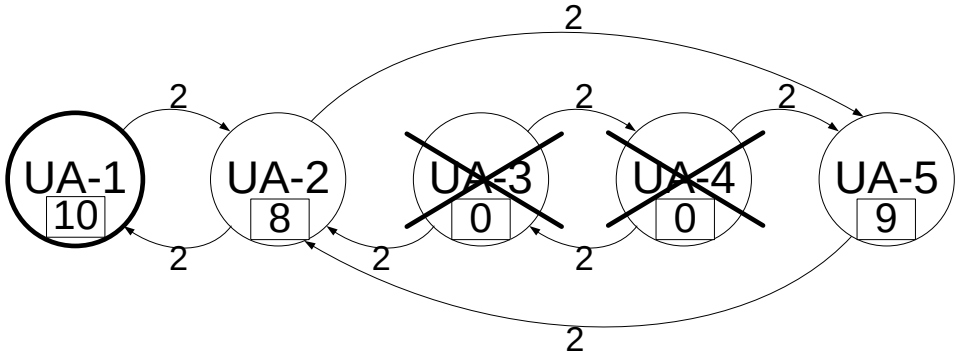

Figura 5.27: Ônibus na UA-1. E exclusão de UAs 3 e 4.

Com a exclusão das UAs 3 e 4 da lista de UAs da linha de ônibus, o valor máximo dos pesos das UAs diminui, segundo a Equação 5.1. Porém, o peso de cada UA só será atualizado, obedecendo o novo limite, quando o ônibus alcançar cada UA. Esse processo pode ser visto na Figura 5.28, onde o peso da UA-2 é atualizado para o novo máximo. Após os pesos de todas as UAs terem sido atualizados, a rota final da linha de ônibus, que pode ser vista na Figura 5.29, é alcançada.

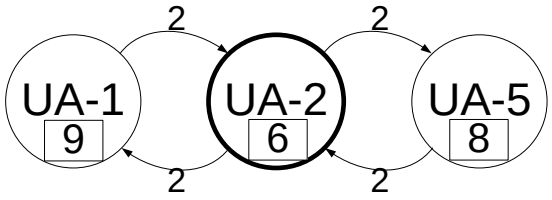

Figura 5.28: Ônibus na UA-2. Atualização de peso para novo máximo.

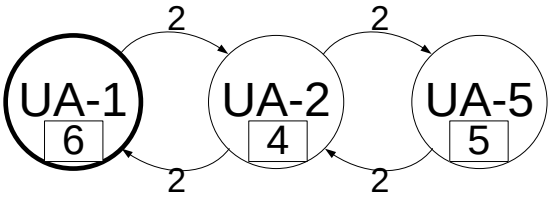

Figura 5.29: Rota final da linha de ônibus.

Com esse exemplo mostra-se que os mapas das linhas de ônibus podem ser alterados dinamicamente. Vale lembrar que a velocidade de concretização das alterações pode ser configurada escolhendo o valor máximo dos pesos dos arcos das UAs.

Capítulo 6

Resultados

Nesse capítulo os resultados obtidos através de emulação são apresentados.

6.1 Ambiente de Testes

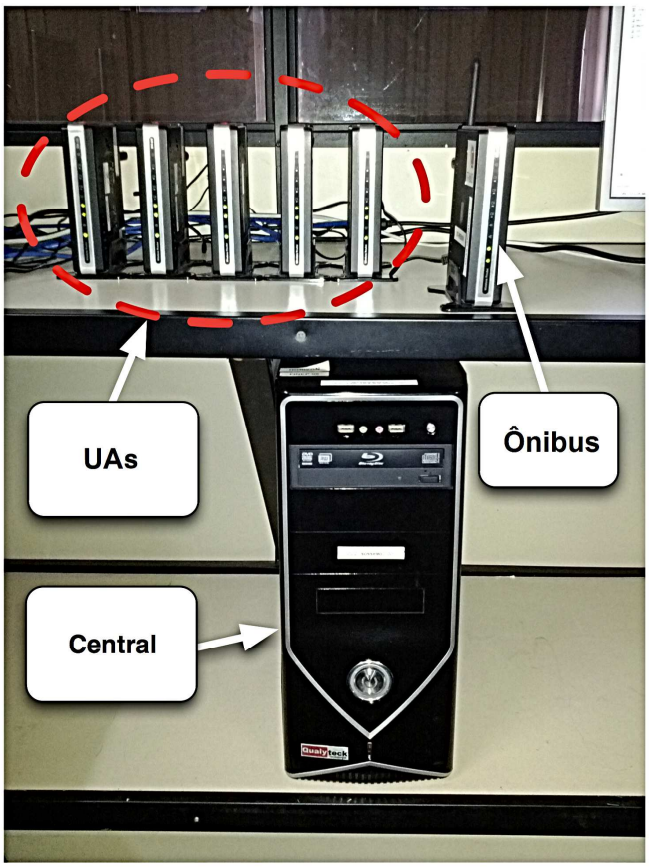

Figura 6.1: Ambiente de testes.

A Central é um computador de mesa, que possui o sistema operacional Debian instalado, 8 GB de memória RAM e processador Intel Core i7 860. Por outro lado, as UAs e os ônibus são representados por roteadores sem-fio da marca D-Link, modelo DIR-320. A Central e as UAs estão conectadas através de um roteador, também da marca D-Link e modelo DIR-320, que foi configurado para funcionar como comutador. Já a conexão das UAs e da Central com esse comutador é realizada por cabo Ethernet. Esse modelo possui processador ARM de 240 MHz e 32 MB de memória RAM. Além disso, possui uma entrada USB que pode ser usada para aumentar a capacidade de armazenamento do roteador, que originalmente é de apenas 4 MB, ou para instalar uma segunda interface de rede sem-fio. O sistema operacional usado nos roteadores é o OpenWRT versão Backfire 10.03.1, uma distribuição Linux para dispositivos embarcados.

6.2 Linhas COPPEAD e Estação UFRJ

Nesse teste, as rotas das linhas de ônibus universitárias COPPEAD (Figura 6.2) e Estação UFRJ (Figura 6.3) foram simuladas. Essas linhas de ônibus são de circulação interna da UFRJ e servem alunos e funcionários dessa universidade. Para isso, definiu-se que no cenário do teste cada ponto de ônibus das rotas, tem uma UA instalada, não existindo outras UAs fora desses pontos. Para que os testes realizados se aproximassem da realidade, o roteador que representava os ônibus estava conectado via rede sem-fio ao roteador que fazia o papel das UAs, incluindo os possíveis problemas causados pela rede sem-fio no teste. Porém, sem o agravante de um dos roteadores estar em movimento.

A movimentação do ônibus pelas rotas foi simulada com o uso de um script no roteador que representava os ônibus. Esse script modificava os arquivos que armazenam as informações dos ônibus, UA atual, UA anterior, linha de ônibus e identificador do ônibus, de forma programada. Em seguida, o script chama a aplicação que envia a mensagem de localização à Central. A Central, então, recebe as mensagens dos ônibus como se eles realmente estivessem realizando as rotas.

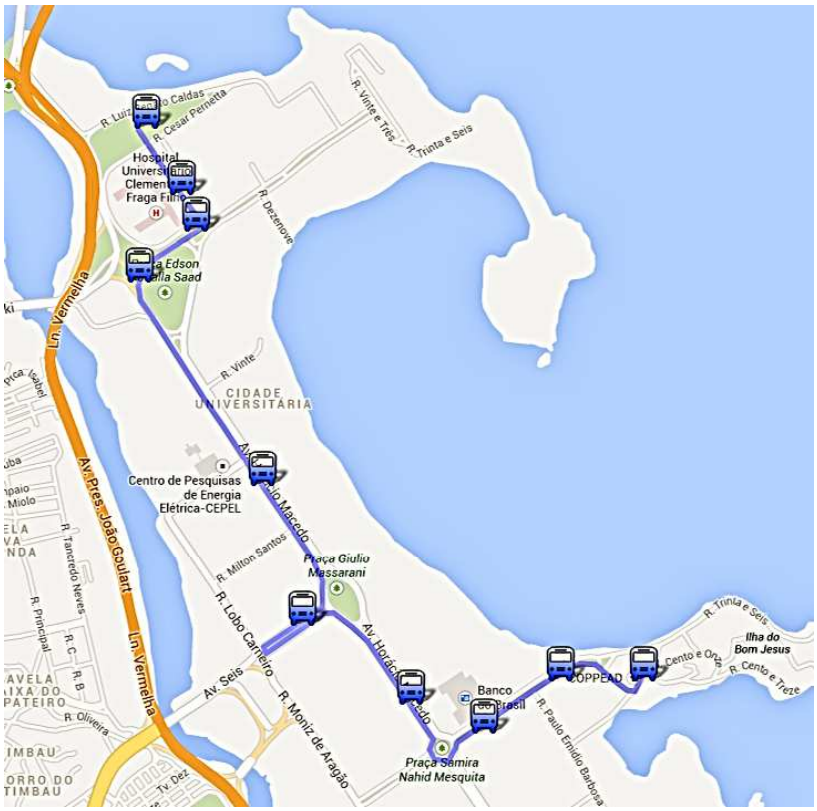

Figura 6.2: Rota da linha de ônibus universitária COPPEAD.

As duas linhas juntas possuem um total de 22 pontos de ônibus, e para o teste condizer com a realidade das linhas, o tempo que um ônibus gastava entre dois pontos de ônibus consecutivos das duas linhas foi medido experimentalmente com um cronômetro. Essa medição foi realizada apenas uma vez, servindo apenas como referência para o teste ter conexão com a realidade. Esses tempos medidos foram usados no script citado anteriormente, de maneira que a simulação levasse em conta não só as rotas das linhas, mas também os intervalos de tempo inerentes a elas.

No teste, considerou-se a existência de cinco ônibus, todos eles iniciavam o trajeto na linha COPPEAD, realizando a linha Estação UFRJ em seguida. Um ônibus entrava no teste 5 minutos após o anterior ter iniciado seu percurso. Uma vez que o ônibus iniciava o percurso do teste, este percurso era repetido indefinidamente. Os ônibus após terminarem a linha Estação UFRJ realizavam novamente a linha COPPEAD e assim sucessivamente. Isso é plausível, pois as duas linhas são circulares, isto é, o ponto final de uma é o ponto de origem da outra e vice-versa.

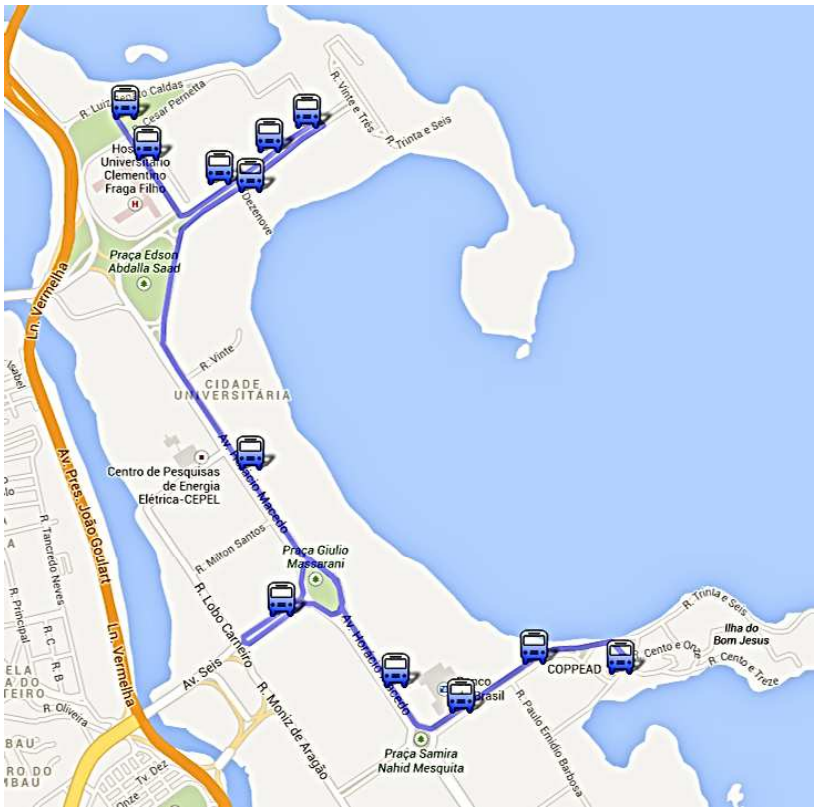

Figura 6.3: Rota da linha de ônibus universitária Estação UFRJ.

O teste foi interrompido após 11 mil mensagens terem sido enviadas pelos ônibus, o que equivale a cada um dos 5 ônibus ter percorrido o trajeto com 22 pontos de ônibus 100 vezes. O teste durou 64 horas e 28 minutos e todas as mensagens foram recebidas pela Central. As estimativas realizadas não continham erros, como era de se esperar, já que os tempos de percurso do ônibus não variavam no teste, pois eram previamente programados. As estatísticas coletadas do teste encontram-se na Tabela 6.1. É interessante observar que o fato do tamanho das mensagens ser pequeno evita os problemas decorrentes do movimento do ônibus e do tempo de contato entre ônibus e UA ser limitado. O tempo de tratamento de mensagem tem relação direta com o número de mensagens que o sistema desenvolvido é capaz de tratar por segundo, e essa métrica será melhor avaliada nos testes seguintes.

Tabela 6.1: Estatísticas do teste das linhas COPPEAD e Estação UFRJ.

Estatísticas	**Média**	**Desvio padrão**
Tamanho de mensagens de localização	152,59 bytes	4,52 bytes
Tempo de tratamento de mensagens	2,039 ms	0,769 ms

6.3 Tempo de Tratamento de Mensagens de Localização

Para avaliar a métrica de tempo de tratamento de mensagens de localização, vários testes foram realizados. Os testes tem a intenção de mostrar como essa métrica, do programa da Central, se comporta perante a variação dos parâmetros do cenário. Para que os resultados avaliassem somente o programa, as simulações foram feitas executando os scripts que realizavam o papel de ônibus dentro do próprio computador onde a Central executa, eliminando a influência da rede nos testes. Avaliar o tempo de tratamento de uma mensagem é importante pois esse tempo tem relação direta com o número de mensagens que o sistema desenvolvido é capaz de tratar por segundo. A partir desse número, é possível avaliar se o sistema é capaz ou não de atender as exigências de uma cidade, o que foi avaliado no último teste realizado.

Ao todo quatro simulações foram feitas, sendo três delas avaliando a influência das características do cenário de forma isolada e uma simulando as condições da cidade do Rio de Janeiro. As características do cenário que influenciam o sistema foram:

1. Quantidade de ônibus cadastrados no sistema;
2. Quantidade de linhas de ônibus cadastradas no sistema;
3. Quantidade de UAs em uma linha de ônibus do sistema.

Os gráficos de todas as simulações mostram, para cada variação do parâmetro da simulação, a média e o desvio padrão dos tempos de tratamento das mensagens.

6.3.1 Variação da Quantidade de Ônibus

No teste em que se avaliou a influência da quantidade de ônibus cadastrados no sistema sobre o tempo de tratamento de mensagem, variou-se o número de ônibus cadastrados no sistema de 1000 até 8000 ônibus, incrementando de 1000 em 1000. O valor máximo foi definido por se aproximar do número de ônibus médio em operação na cidade do Rio de Janeiro em 2011, que era de 8413 [33].

Os ônibus foram cadastrados por um script que enviava uma mensagem contendo o identificador do novo ônibus, a linha pré-estabelecida usada no teste e a primeira UA da linha como ponto atual. A Central ao receber essa mensagem cadastrava os ônibus no sistema. Após todos os ônibus terem sido cadastrados, escolheu-se um deles para realizar a rota da linha pré-estabelecida. A rota da linha possuía 5 UAs cadastradas e o ônibus percorreu a rota 10 vezes seguidas. Quando o ônibus chegava à próxima UA ele enviava a mensagem de localização à Central, que então tratava a mensagem e media o tempo do tratamento dela.

Todo o processo foi realizado três vezes para cada quantidade de ônibus escolhida, resultando em 150 mensagens para cada valor. É interessante notar que o teste visa avaliar a influência do parâmetro variado no tempo de tratamento de mensagens dos ônibus já cadastrados, e não o tempo para cadastrar um novo ônibus no sistema. Os resultados do teste podem ser observados na Figura 6.4.

Pode-se notar pelo gráfico na Figura 6.4 que a média e o desvio padrão do tempo de tratamento por mensagem aumentam com o número de ônibus cadastrados. Além disso, a curva obtida se assemelha a uma reta com as variações nos tempos médios entre dois pontos consecutivos simulados em torno de 5 ms. Esse teste apesar de demonstrar o comportamento do sistema, não é de senso prático, pois os mais de 1000 ônibus foram cadastrados na mesma linha de ônibus, o que nunca acontece na vida real. O resultado foi o alongamento do cálculo da estimativa da linha de ônibus, já que a estimativa ao ser calculada durante o tratamento das mensagens avalia todos os ônibus cadastrados na linha sendo estimada.

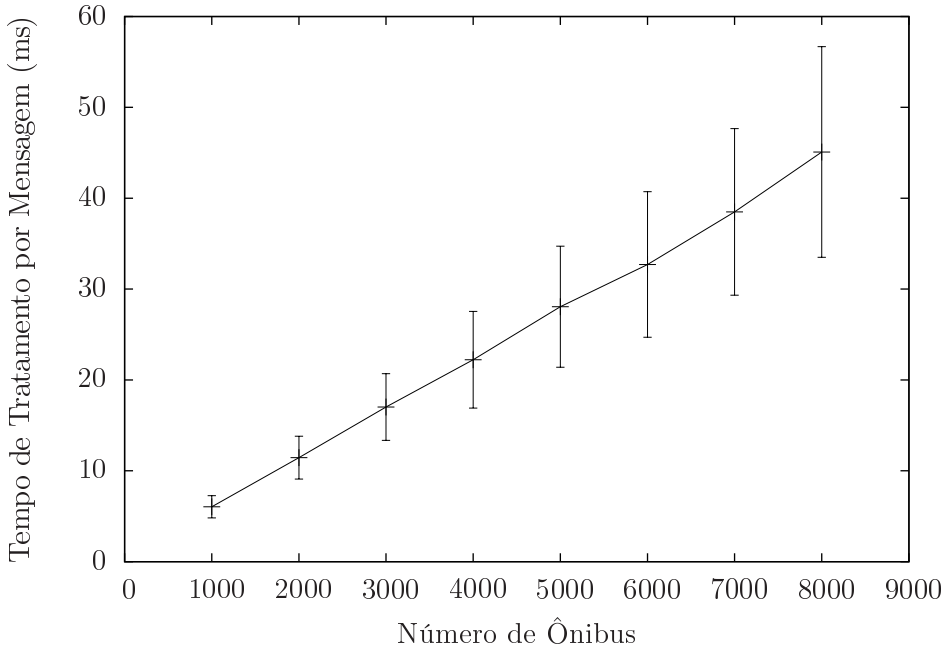

Figura 6.4: Variação da quantidade de ônibus no sistema.

6.3.2 Variação da Quantidade de Linhas de Ônibus

No teste em que se avaliou a influência da quantidade de linhas de ônibus cadastradas no sistema, variou-se o número de linhas de ônibus cadastradas de 100 até 800, incrementando de 100 em 100. O valor máximo foi definido por se aproximar do número de linhas de ônibus em operação na cidade do Rio de Janeiro em 2011, que era de 822 [33].

Durante esse teste, somente um ônibus foi cadastrado no sistema para eliminar a influência desse parâmetro. As linhas foram cadastradas por um script que enviava uma mensagem contendo o identificador do ônibus pré-estabelecido, a nova linha que o ônibus estava realizando e a primeira UA da nova linha como ponto atual. A Central ao receber essa mensagem identificava que o ônibus havia mudado de linha e cadastrava as novas linhas de ônibus no sistema. Após todas as linhas terem sido cadastradas, escolheu-se uma para o ônibus percorrer. A rota da linha possuía 5 UAs cadastradas e o ônibus realizou a rota 10 vezes consecutivas. Da mesma forma que no teste anterior, quando o ônibus alcançava a próxima UA ele enviava a mensagem de localização à Central, que tratava a mensagem e media o tempo do tratamento dela.

Assim como no teste anterior, todo o processo foi realizado três vezes para cada quantidade de linhas de ônibus, resultando em 150 mensagens para cada valor. Assim como o teste anterior, o teste tem a intenção de avaliar como a alteração do número de linhas de ônibus inseridas no sistema afeta o tempo de tratamento de mensagens dos ônibus percorrendo a rota, e não o tempo para cadastrar uma nova linha de ônibus no sistema. Os resultados do teste podem ser observados na Figura 6.5.

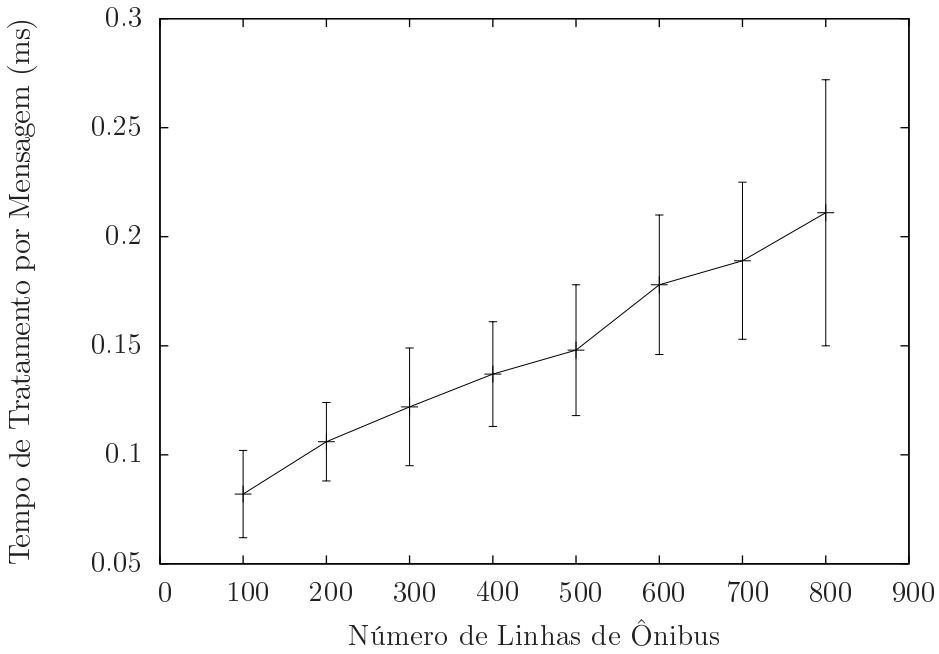

Figura 6.5: Variação da quantidade de linhas de ônibus no sistema.

Pelo gráfico na Figura 6.5, observa-se que a média e o desvio padrão aumentam com o número de linhas de ônibus cadastradas no sistema, contudo para os valores simulados o crescimento apresentado não impediu a escalabilidade.

6.3.3 Variação da Quantidade de UAs em uma Linha de Ônibus

Nesse teste, variou-se o número de UAs numa linha de 10 até 100, com incrementos de 10 em 10. Durante o teste, para eliminar da melhor forma possível a influência dos outros parâmetros, apenas uma linha de ônibus e um ônibus foram cadastrados.

As UAs foram inseridas na linha através do envio das mensagens do ônibus percorrendo a rota pela primeira vez. Após a rota da linha ficar pronta, fez-se o ônibus percorrer a rota criada 10 vezes seguidas. Quando o ônibus aparecia na próxima UA ele enviava a mensagem de localização à Central, que tratava a mensagem e cronometrava seu tempo do tratamento.

Assim como nos dois testes anteriores, todo o processo foi realizado três vezes para cada variação de UAs por linha. Como o número de UAs na rota percorrida varia, o número de mensagens, e consequentemente de tempos medidos, para cada valor também se modifica, aumentando com o número de UAs na rota. Assim como os testes anteriores, o objetivo do teste é avaliar a influência da variação do parâmetro em questão no tempo de tratamento de mensagens dos ônibus percorrendo a rota, e não o tempo necessário para inserir uma UA numa linha de ônibus. Os resultados do teste podem ser observados na Figura 6.6.

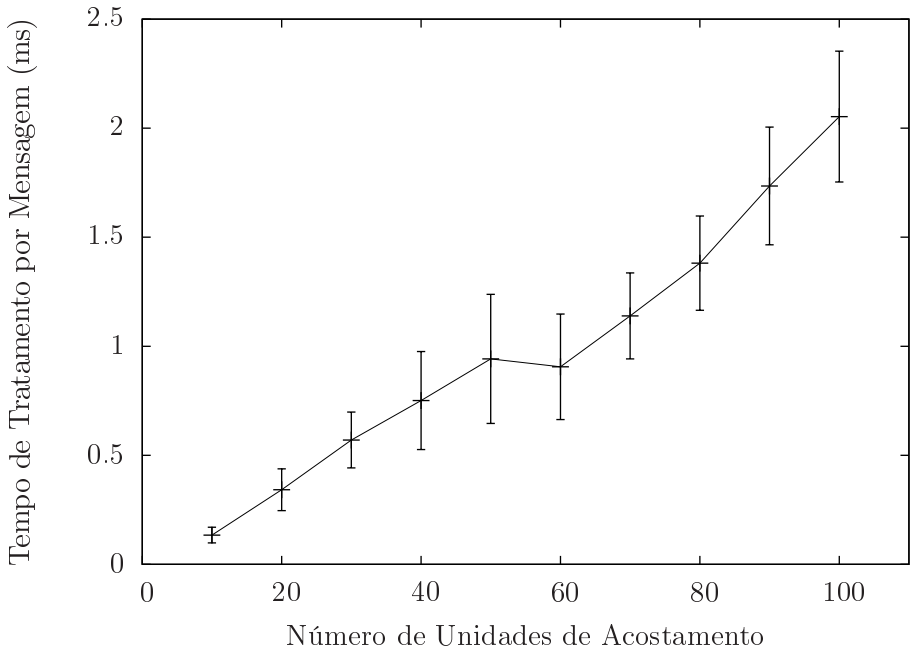

Figura 6.6: Variação da quantidade de UAs no sistema.

Na Figura 6.6, observa-se que a média dos tempos de tratamento de mensagem aumenta com o número de UAs presentes em uma dada linha de ônibus. Entretanto, para os valores simulados, o crescimento apresentado não impediu a escalabilidade,

com o tempo médio de tratamento por mensagem permanecendo abaixo de 2,5 ms.

6.3.4 Cenário Rio de Janeiro

O último teste simula as condições da cidade do Rio de Janeiro, sendo cadastradas no sistema 800 linhas de ônibus, cada uma com 10 ônibus, totalizando os 8000 ônibus que se aproximam do número de ônibus da cidade. Como o número de UAs inseridas em uma linha pode variar, por exemplo, devido à variação de pontos de ônibus nas linhas, decidiu-se variar o número de UAs como no teste anterior, de 10 a 100 UAs em uma linha de ônibus variando de 10 em 10.

Como nos testes anteriores, cadastraram-se as linhas de ônibus e os ônibus, e em seguida uma das linhas de ônibus teve sua rota criada com o número de UAs estabelecido. Após a criação da rota, um dos ônibus percorreu a mesma 10 vezes consecutivas, enviando mensagens para a Central ao trocar de UA. A Central tratou as mensagens e mediu o tempo gasto no tratamento delas.

O processo foi repetido três vezes para cada variação de UAs. Os resultados do teste podem ser observados na Figura 6.7.

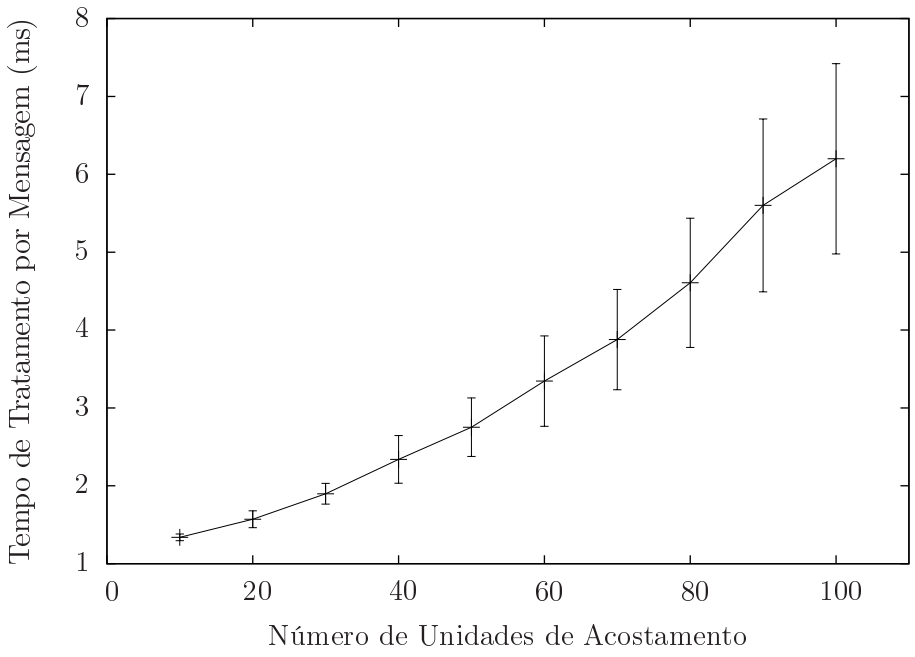

Figura 6.7: Cenário Rio de Janeiro.

Pelo gráfico observa-se que a média dos tempos de tratamento de mensagem aumenta com o número de UAs presentes numa dada linha de ônibus. Com os dados obtidos no gráfico, pode-se inferir que apenas um computador semelhante ao usado executando a aplicação desenvolvida é suficiente para monitorar e estimar toda a cidade do Rio de Janeiro dado que o tempo de tratamento por mensagem não passou de 7 ms. Essa inferência é realizada através da suposição de que os aproximadamente 8000 ônibus da cidade levam pelo menos um minuto para alcançar a próxima UA da rota. Com isso, o programa teria 60 segundos para tratar as 8000 mensagens. Considerando o pior caso do gráfico da Figura 6.7, ou seja, cada linha de ônibus com 100 UAs, o tratamento de cada mensagem levaria em média, aproximadamente, 7 milissegundos. Dividindo-se os 60 segundos por esse tempo de tratamento, obtém-se uma taxa de aproximadamente 8500 mensagens tratadas por minuto. Portanto, todas as mensagens que fossem enviadas à Central conseguiriam ser tratadas, mesmo considerando o pior caso para todas as linhas de ônibus da cidade, o que mostra a capacidade da aplicação criada.

Capítulo 7

Conclusões e Trabalhos Futuros

O objetivo deste projeto final foi propor e desenvolver um sistema que, utilizando roteadores sem-fio instalados nos ônibus e nos pontos de ônibus, fosse capaz de rastrear e estimar os horários que os ônibus passam nos pontos requisitados pelos usuários. Para isso, foi desenvolvido um serviço de ITS de localização automática de veículos para rastrear os ônibus, no qual se utilizou a técnica de localização por proximidade, considerando os pontos de acesso do padrão IEEE 802.11 como referência. Apesar de o sistema AVL criado ser focado no rastreamento de veículos de transporte público, a aplicação de localização projetada também pode ser usada para veículos de uso pessoal.

O serviço proposto de estimativa de chegada de ônibus aos pontos de ônibus pode ser desenvolvido a partir de quatro formas de cálculo, sendo todas elas baseadas em médias de informações anteriores. Uma página na Internet foi criada para disponibilizar as estimativas calculadas para os usuários dos meios de transporte público.

Neste projeto de fim de curso, uma forma de criação e manutenção de mapas digitais dinâmicos, que se adaptam a modificações nas rotas monitoradas, para cenários semelhantes também foi proposta e desenvolvida. O funcionamento do mecanismo proposto foi descrito passo a passo em dois exemplos, um descrevendo a criação de um mapa digital dinâmico e outro descrevendo a reação do mapa a uma modificação na rota.

As simulações realizadas incluíram um teste de longa duração, mostrando a integridade do sistema como um todo. Além disso, o tempo de tratamento das mensagens de localização, que inclui desde a localização do veículo, passando pela criação e manutenção do mapa, até o cálculo das estimativas, foi avaliado frente às principais variáveis do sistema. Para concluir os testes, uma simulação com parâmetros similares aos encontrados na cidade do Rio de Janeiro foi realizada. Nessa simulação havia 8000 ônibus e 800 linhas de ônibus cadastrados no sistema, e variou-se entre 10 e 100 o número de unidades de acostamento por linha de ônibus. Os resultados obtidos mostraram que a aplicação criada atenderia, mesmo no pior caso, as exigências de uma cidade grande como o Rio de Janeiro.

Como trabalhos futuros, seria interessante obter mais dados práticos dos tempos gastos pelos ônibus para percorrer trechos das linhas de ônibus. De posse de mais dados seria possível avaliar a precisão dos diferentes mecanismos de estimativa criados, e também como esses mecanismos reagem a diferentes condições de trânsito, como um engarrafamento. Além disso, seria interessante colocar o sistema em funcionamento, instalando UAs nos pontos de ônibus e roteadores sem-fio nos ônibus internos da UFRJ para, além de avaliar o funcionamento do sistema criado em um cenário real, aumentar a qualidade de serviço dos ônibus internos, beneficiando os usuários.

Referências Bibliográficas

[1] PADMANABAN, R. P. S., DIVAKAR, K., VANAJAKSHI, L., *et al.*, "Development of a real-time bus arrival prediction system for Indian traffic conditions", *Intelligent Transport Systems, IET*, v. 4, n. 3, pp. 189–200, 2010.

[2] Instituto Municipal de Urbanismo Pereira Passos, "Total de veículos terrestres automotores (frota ativa) - Município do Rio de Janeiro - 1994 - 2011 (Tabela Nº 1253)", Armazém de Dados - Informações sobre a cidade do Rio - `http://www.armazemdedados.rio.rj.gov.br/`, 2011, (Acesso em 12 Junho 2013).

[3] WEILAND, R. J., PURSER, L. B., "Intelligent transportation systems", *Transportation in the New Millennium*, p. 3, 2000.

[4] Bus Standards Program and Bus Rapid Transit Working Group, *Recommended Practice for Implementing BRT Intelligent Transportation Systems*, Report, American Public Transportation Association, 2010.

[5] RITER, S., MCCOY, J., "Automatic vehicle location - An overview", *Vehicular Technology, IEEE Transactions on*, v. 26, n. 1, pp. 7–11, 1977.

[6] SKABARDONIS, A., "Traffic Signal Control Systems". In: *Assessing the Benefits and Costs of ITS*, v. 10, *Transportation Research, Economics and Policy*, Springer US, pp. 131–144, 2004.

[7] AMPELAS, A., "Automatic fare collection". In: *Intelligent Transportation Systems, 2001. Proceedings. IEEE*, pp. 1164–1166.

[8] BALASUBRAMANIAN, N., BALASUBRAMANIAN, A., VENKATARAMANI, A., "Energy consumption in mobile phones: a measurement study and

implications for network applications". In: *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, IMC '09, pp. 280–293, 2009.

[9] BusTV, "A Rede BusTV", `http://bustv.com.br/portal/a-rede-bustv`, 2013, (Acesso em 9 Julho 2013).

[10] FENG, H., LULU, L., HENG, Y., *et al.*, "Bus Monitoring System Based on ZigBee and GPRS". In: *Computer Distributed Control and Intelligent Environmental Monitoring (CDCIEM), 2012 International Conference on*, pp. 178–181, 2012.

[11] CACERES, M., SOTTILE, F., SPIRITO, M., "WLAN-Based Real Time Vehicle Locating System". In: *Vehicular Technology Conference. VTC Spring 2009. IEEE 69th*, pp. 1–5.

[12] DOD, U., "Global positioning system standard positioning service performance standard", 2008.

[13] MANOLIS, K., KWSTIS, D., "Intelligent transportation systems - travelers' information systems the case of a medium size city". In: *Mechatronics. ICM '04. Proceedings of the IEEE International Conference on*, pp. 200–204, 2004.

[14] LIN, W.-H., ZENG, J., "Experimental study of real-time bus arrival time prediction with GPS data", *Transportation Research Record: Journal of the Transportation Research Board*, v. 1666, n. 1, pp. 101–109, 1999.

[15] KARBASSI, A., BARTH, M., "Vehicle route prediction and time of arrival estimation techniques for improved transportation system management". In: *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, pp. 511–516, 2003.

[16] D'ANGELO, M. P., AL-DEEK, H. M., WANG, M. C., "Travel-time prediction for freeway corridors", *Transportation Research Record: Journal of the Transportation Research Board*, v. 1676, n. 1, pp. 184–191, 1999.

[17] ISHAK, S., AL-DEEK, H., "Performance Evaluation of Short-Term Time-Series Traffic Prediction Model", *Journal of Transportation Engineering*, v. 128, n. 6, pp. 490–498, 2002.

[18] RICE, J., VAN ZWET, E., "A simple and effective method for predicting travel times on freeways", *Intelligent Transportation Systems, IEEE Transactions on*, v. 5, n. 3, pp. 200–207, 2004.

[19] FRECHETTE, L. A., KHAN, A. M., "Bayesian regression-based urban traffic models", *Transportation Research Record: Journal of the Transportation Research Board*, v. 1644, n. 1, pp. 157–165, 1998.

[20] CHIEN, S., DING, Y., WEI, C., "Dynamic Bus Arrival Time Prediction with Artificial Neural Networks", *Journal of Transportation Engineering*, v. 128, n. 5, pp. 429–438, 2002.

[21] VANAJAKSHI, L., RILETT, L., "Support Vector Machine Technique for the Short Term Prediction of Travel Time". In: *Intelligent Vehicles Symposium, 2007 IEEE*, pp. 600–605, 2007.

[22] JEONG, R., RILETT, L., "Bus arrival time prediction using artificial neural network model". In: *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*, pp. 988–993, 2004.

[23] VANAJAKSHI, L., SUBRAMANIAN, S., SIVANANDAN, R., "Travel time prediction under heterogeneous traffic conditions using global positioning system data from buses", *IET intelligent transport systems*, v. 3, n. 1, pp. 1–9, 2009.

[24] SHALABY, A., FARHAN, A., "Bus travel time prediction model for dynamic operations control and passenger information systems", *Transportation Research Board*, p. 16, 2003.

[25] CHENG, S., "Bus arrival time prediction model based on APC data", *6th Advanced Forum on Transportation of China (AFTC 2010)*, pp. 165–169(4), 2010.

[26] DA SILVA, V. B., DA SILVA, F. O., CAMPISTA, M. E. M., *et al.*, "Roteamento Baseado na Trajetória para Redes Veiculares Desconectadas com Multiplos Gateways", *XXXI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC*, pp. 1038–1051, 2013.

[27] DA SILVA, V. B., DA SILVA, F. O., CAMPISTA, M. E. M., *et al.*, "A Trajectory-Based Approach to Improve Delivery in Drive-Thru Internet Scenarios", *Workshop on Emerging Vehicular Networks: V2V/V2I and Railroad Communications - IEEE International Conference on Communications*, pp. 499–504, 2013.

[28] SIM, M. L., NEKOVEE, M., KO, Y. F., "Throughput analysis of Wi-Fi based broadband access for mobile users on the highway". In: *Networks, 2005. Jointly held with the 2005 IEEE 7th Malaysia International Conference on Communication., 2005 13th IEEE International Conference on*, v. 1, p. 6, 2005.

[29] JÚNIOR, J. G. R., QUINTANILHA, I. M., CAMPISTA, M. E. M., *et al.*, "Sistema para Monitoramento Descentralizado de Trânsito Baseado em Redes Veiculares Infraestruturadas", *XXXI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC*, pp. 863–876, 2013.

[30] Carlos A. Maziero, "The ARP tool", `http://dainf.ct.utfpr.edu.br/~maziero/doku.php/software:arp_tool`, 2009, (Acesso em 3 Julho 2013).

[31] apaloczi, "OpenWRT c,c++ programing", `http://fleshandmachines.wordpress.com/2011/08/22/openwrt-cc-programing/`, 2011, (Acesso em 3 Julho 2013).

[32] Google, "Google Maps Javascript API V3", `https://developers.google.com/maps/documentation/javascript/reference`, 2013, (Acesso em 03 Julho 2013).

[33] Instituto Municipal de Urbanismo Pereira Passos, "Total de linhas, frota operante, passageiros transportados, viagens realizadas,quilometragem coberta, combustível utilizado e pessoal ocupado pelo sistema de ônibus - Município do Rio de Janeiro - 1994 - 2011 (Tabela Nº 1736)", Armazém de Dados - Informações sobre a cidade do Rio - `http://www.armazemdedados.rio.rj.gov.br/`, 2011, (Acesso em 12 Junho 2013).

Apêndice A

Propriedades da Rede de Petri

A.1 Descrição da Rede de Petri para Simulador ARP

Para inserir a rede de Petri no simulador utilizado, é necessário representar a rede em forma de texto. Essa representação encontra-se abaixo:

```
NET TRABALHO;

NODES

   P2, P3, P5, P6, P7, P8, P9, P11, P12, P13, P14, P16, P17 : PLACE;
   P1: PLACE(1);
   P4: PLACE(1);
   P10: PLACE(2);
   P15: PLACE(2);
   T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12: TRANSITION;

STRUCTURE

   T1:  (P1,P2)  ,(P1,P3);
   T2:  (P4)    ,(P5);
   T3:  (P5)     ,(P2,P6);
   T4:  (P3,P6)  ,(P7,P8);
   T5:  (P7,P9)  ,(P4);
   T6:  (P8,P10) ,(P9,P11);
   T7:  (P11)   ,(P12);
```

```
   T8:  (P12)    ,(P10);
   T9:  (P10,P13) ,(P10,P14);
   T10: (P15)    ,(P16);
   T11: (P16)    ,(P13,P17);
   T12: (P14,P17) ,(P15);

ENDNET.
```

A.2 Saída da Análise da Rede

A rede é limitada:

```
Net under analysis is limited.
  Null places (M = 0): {}
  Binary places     : {P2, P3, P5, P6, P7, P8, P9, P1, P4}
  k-Bounded places : {2* P11, 2* P12, 2* P13, 2* P14, 2* P16, 2* P17, 2*
      P10, 2* P15}
  Unbounded places : {}
```

A rede não é estritamente conservativa.

A rede é viva, ou seja, não possui bloqueios.

```
Net under analysis is live.
  Live Tr.        : {all}
  "Almost-live" Tr.: {all}
  Non-fired Tr.   : {}
No live-locks detected.
No deadlocks detected.
```

A rede é reinicializável, isto é, partindo de qualquer marcação válida, ela sempre pode voltar para a marcação inicial.

Invariantes de transição:

```
Transition invariant basis:
BI1  :{T1, T2, T3, T4, T5, T6, T7, T8}
BI2  :{T10, T11, T12, T9}
```

Invariantes de lugar:

```
Place invariant basis:
BI1  :{P2, P3, -P6}
BI2  :{P13, P14, -P17}
BI3  :{P7, -P8, -P9}
BI4  :{P1}
BI5  :{P2, P3, P5, P7, P4}
BI6  :{P11, P12, P10}
BI7  :{P13, P14, P16, P15}
Minimum positive place invariants for this net:
IL1  :{P11, P12, P10}
IL2  :{P13, P14, P16, P15}
IL3  :{P16, P17, P15}
IL4  :{P2, P3, P5, P8, P9, P4}
IL5  :{P5, P6, P8, P9, P4}
IL6  :{P5, P6, P7, P4}
IL7  :{P1}
IL8  :{P2, P3, P5, P7, P4}
```