

## 5 Experimentos

Este capítulo descreve o procedimento experimental para a avaliação dos algoritmos TLAR e U-GPSR e a comparação de seus desempenhos com o de outras abordagens. Os detalhes do ambiente de simulação, incluindo as ferramentas computacionais utilizadas, são apresentados na Seção 5.1. A lista dos experimentos conduzidos e seus resultados se encontram na Seção 5.2, que, a seguir, são discutidos na Seção 5.3.

### 5.1 Ambiente de Simulação

Tendo em vista a indisponibilidade do equipamento necessário para efetuar testes de campo com o TLAR, a avaliação de seu desempenho foi feita por meio de simulação. As ferramentas utilizadas são apresentadas na Seção 5.1.1, enquanto o procedimento de simulação é explicado na Seção 5.1.2 e a Seção 5.1.3 discute quais algoritmos foram selecionados para testes, justificando a escolha.

#### 5.1.1 Ferramentas Utilizadas

Uma boa ferramenta para a simulação de VANETs deve, preferencialmente, permitir a modelagem tanto da movimentação dos veículos quanto da comunicação sem fio. Até o momento inicial de desenvolvimento desta tese, nenhuma ferramenta que atendesse a esse requisito foi encontrada, motivando a utilização de dois simuladores: um de tráfego (SUMO) e outro de redes sem fio (NS-2). Essas ferramentas são apresentadas resumidamente a seguir.

#### SUMO

O SUMO (*Simulation of Urban MObility*) (KRA02) é um simulador de tráfego de escala microscópica, ou seja, que calcula constantemente a posição e a velocidade instantânea de cada veículo. Sua implementação é baseada no modelo de Stefan Krauß (KRA98) (discutido na Seção 4.1.2), e simula manobras complexas a exemplo de ultrapassagens e mudanças de

```

<nodes>
  <node id="A" x="300" y="300" type="traffic_light"/>
  <node id="B" x="600" y="300" type="traffic_light"/>
  <node id="C" x="900" y="300" type="traffic_light"/>
  <node id="D" x="900" y="600" type="traffic_light"/>
</nodes>

```

Figura 5.1: Exemplo de arquivo de cruzamentos do SUMO.

```

<edges>
  <edge id="AB" fromnode="A" tonode="B" nolanes="3" speed="20.0"/>
  <edge id="BC" fromnode="B" tonode="C" nolanes="3" speed="20.0"/>
  <edge id="CD" fromnode="C" tonode="D" nolanes="3" speed="20.0"/>
</edges>

```

Figura 5.2: Exemplo de arquivo de *segmentos* do SUMO.

faixa. Além disso, trata-se de uma ferramenta gratuita, portátil, de código aberto (*open source*), e que já foi utilizada com sucesso em vários projetos (LAH07, WEG06), sendo esses os motivos de ele ter sido adotado para os experimentos desta tese.

Uma simulação no SUMO requer como entrada a especificação da malha viária e dos fluxos de veículos em formato XML. O resultado das simulações consiste em um relatório (*log*), também em formato XML, em que a posição e velocidade de cada veículo é discriminada para cada instante de tempo.

A descrição de uma malha viária consiste em uma lista de cruzamentos e uma lista de *segmentos* entre pares de cruzamentos. As Figuras 5.1 e 5.2 contém a descrição da topologia utilizada como exemplo na Figura 5.3, que ilustra três cruzamentos (*nodes*) *A*, *B*, *C* e *D* conectados por *segmentos* (*edges*) *AB*, *BC* e *CD*. Esses cruzamentos têm semáforos, e as ruas entre eles têm três faixas (atributo *nolanes*) e admitem uma velocidade máxima de 20 m/s.

No instante inicial de simulação, a topologia das ruas se encontra vazia. Os locais e os instantes de tempo em que os veículos serão injetados são especificados por meio de *fluxos veiculares*. Esses fluxos também especificam a seqüência de *segmentos* que os veículos percorrerão e algumas de suas características, a exemplo da aceleração e desaceleração máximas. Um exemplo de arquivos de fluxos está listado na Figura 5.4.

O exemplo da Figura 5.4 descreve um fluxo de veículos chamado *my\_flux* que consiste na entrada de um veículo a cada 5 segundos (atributo *period*) a partir do instante de tempo inicial 0 (atributo *depart*), sendo que no máximo 1000 veículos serão inseridos no cenário de simulação através desse fluxo (atributo *repno*). Os veículos gerados dessa forma percorrerão os *segmentos*

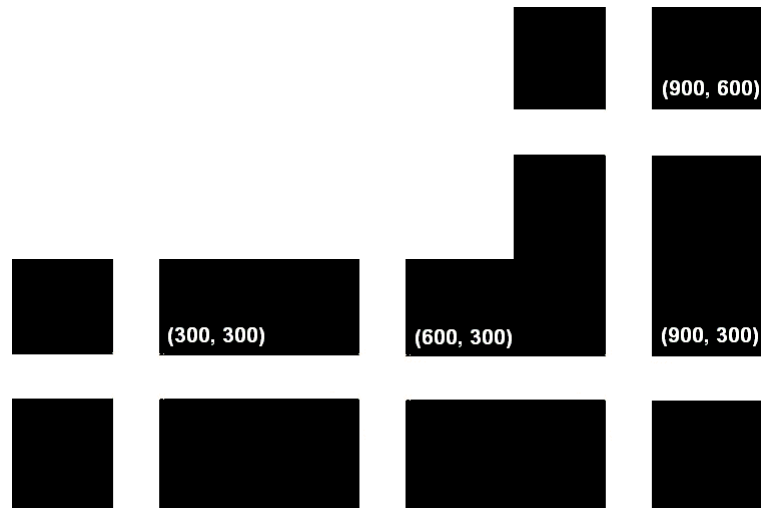


Figura 5.3: Exemplo de topologia no SUMO.

```

<routes>
  <vtype id="my_type" accel="0.7" decel="4.2" sigma="0.5"
    length="5" maxspeed="16"/>
  <vehicle id="my_flux" type="my_type" depart="0" period="5"
    repno="1000">
    <route>AB BC CD</route>
  </vehicle>
</routes>

```

Figura 5.4: Exemplo de descrição de fluxos de veículos.

*AB*, *BC* e *CD*, nessa ordem, e serão retirados do cenário de simulação após percorrer esse último *segmento*. Nota-se que o valor do atributo *repno* foi superdimensionado, a fim de garantir que a chegada de veículos ocorra durante todo o tempo de simulação. Além disso, a taxa de entrada de veículos no cenário de simulação ocorre a uma taxa constante, não sendo possível, na versão utilizada do SUMO, especificar outra distribuição.

Finalmente, o atributo *type* permite especificar o *tipo* dos veículos que compõem o fluxo. Um *tipo*, por sua vez, consiste em uma combinação de parâmetros que permitem um ajuste mais fino do comportamento de um veículo. São eles a aceleração e desaceleração máximas (atributos *accel* e *decel*), o comprimento do veículo (incluindo a distância mínima a ser mantida ao veículo à frente quando em repouso), a velocidade máxima (atributo *maxspeed*) e um parâmetro que modela a componente não-determinística do modelo (atributo *sigma*).

O resultado da simulação é armazenado em um relatório (*log*) XML, que discrimina a posição e a velocidade de cada veículo ao longo do tempo. Esse relatório contém uma entrada (elemento *timestep*) no formato mostrado

```

<timestep time="20">
  <edge id="AB">
    <lane id="AB_0">
      <vehicle id="my_flux_4" pos="25.00" speed="2.00"/>
    </lane>
  </edge>
</timestep>

```

Figura 5.5: Trecho de um relatório gerado por simulação do SUMO.

na Figura 5.5 para cada instante de tempo simulado. Especificamente, esse exemplo relata que, no instante de tempo  $t = 20s$ , o quarto veículo do fluxo *my\_flux* ( $id = "my\_flux\_4"$ ) se encontra a 25 metros do início do *segmento AB* (ou seja, da extremidade *A*), na faixa mais à direita ( $lane\ id="AB\_0"$ ) e a uma velocidade de 2 m/s.

## NS-2

NS-2 (*Network Simulator*) (MCC09) é um simulador discreto de redes de computadores originalmente destinado a simulação de redes fixas e depois estendido com as funcionalidades necessárias à simulação de redes sem fio. Essas extensões incluem funções para a movimentação dos nós, modelos de propagação de sinal e a implementação dos principais protocolos de redes móveis. Contudo, o NS-2 não modela redes veiculares de forma explícita; em vez disso, as simulações são feitas a partir de redes *ad hoc* tradicionais configuradas de forma compatível às VANETs, utilizando, por exemplo, velocidades e modelos de tráfego compatíveis com veículos reais.

A maior parte do código do NS-2 foi escrito em C++, enquanto o restante foi implementado em OTcl, uma versão orientada a objetos da conhecida linguagem de *script* Tcl. O objetivo dessa opção de projeto é o de combinar a eficiência oferecida pela linguagem C++ com a flexibilidade típica de linguagens de *script*. Dessa forma, é possível executar uma mesma simulação sob diversas configurações de parâmetros, tais como o período com que um protocolo envia *beacons*, a frequência de rádio utilizada, entre outros, sem que seja necessário recompilar o código do simulador propriamente dito.

A movimentação das estações é uma parte da simulação que normalmente é implementada em OTcl. Para todos os instantes de tempo, cada estação ou está em repouso ou se movimentando na direção de um destino temporário (*waypoint*) a uma velocidade constante. A Figura 5.6 mostra um trecho de um *script* de mobilidade típico.

Nesse exemplo, *node* é uma estação que, no instante  $t = 5s$  se movimenta

```
$ns at 5 {$node setdest 420 300 10}  
$ns at 8 {$node setdest 450 300 8}
```

Figura 5.6: Exemplo de configuração de mobilidade no NS-2.

para as coordenadas (420, 300) a uma velocidade de  $10m/s$ . No instante  $t = 8s$ , ela passa a se movimentar na direção do ponto (450, 300) a  $8m/s$ . Dessa forma, pode-se construir um padrão de mobilidade arbitrário pelo uso de uma seqüência de comandos *setdest* agendados em sucessivos instantes de tempo.

A configuração de diversos parâmetros de simulação também costuma ser feita por *scripts* OTcl. É possível selecionar a sensibilidade da interface de comunicação sem fio, o modelo de propagação, o tamanho da fila de pacotes à espera de serem transmitidos, os protocolos MAC e de rede, etc. O NS-2 inclui em sua distribuição a implementação de vários protocolos, o que, combinado ao fato de se tratar de uma ferramenta gratuita e de código aberto, o tornou o simulador mais utilizado no meio acadêmico.

### 5.1.2

#### Procedimento de Simulação

De modo a utilizar o SUMO e o NS-2 em conjunto nos experimentos desta tese, foi necessário desenvolver alguns aplicativos para desempenharem certas tarefas menores, tais como criar de forma automatizada os arquivos de entrada para a simulação do SUMO, traduzir sua saída em um *script* de mobilidade para o NS-2, fazer a decodificação (*parsing*) dos resultados e calcular as estatísticas de interesse. A Figura 5.7 resume o procedimento de simulação adotado, discutido em maior detalhe nas seções seguintes.

#### Gerador de Topologias

O gerador de topologias, desenvolvido especificamente para os experimentos desta tese, cria de forma automatizada arquivos XML (como os dos exemplos das Figuras 5.1 e 5.2) que descrevem uma malha viária com topologia Manhattan (COU03). Parâmetros de linha de comando permitem especificar o tamanho da topologia e das quadras, bem como o número de faixas das ruas.

Todos os cruzamentos da topologia resultante são controlados por semáforos. Essa hipótese é aceitável para os experimentos sendo considerados nesta tese porque as ruas desprovidas de semáforos normalmente apresentam tráfego veicular bastante esparso. Somando-se a isso uma baixa *taxa de penetração*, é possível concluir que o roteamento através dessas ruas requer, quase sempre, o armazenamento temporário de pacotes (*store-and-forward*).

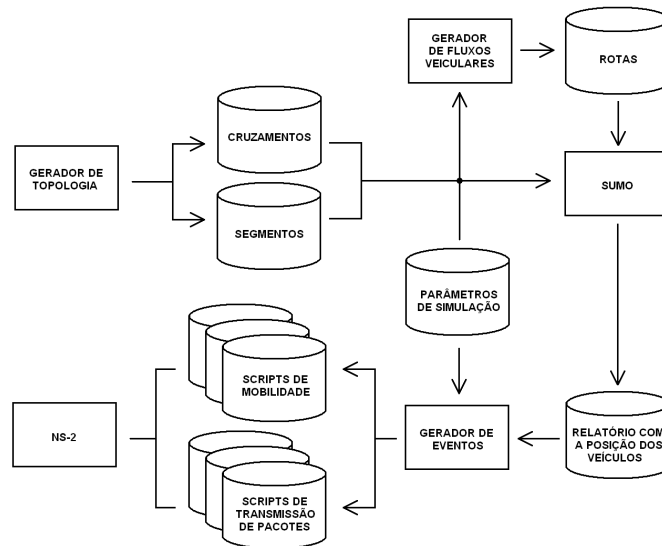


Figura 5.7: Procedimento adotado nas simulações.

Tendo em vista que essa abordagem se encontra além do escopo desta tese, e que as tais ruas não seriam boas opções para a construção de rotas, elas podem ser desconsideradas para efeito de roteamento. Dessa forma é possível reduzir uma topologia real, composta por *segmentos* cujas extremidades podem ou não possuir semáforos, a uma versão que contenha apenas aqueles que se encontrem entre dois cruzamentos controlados por semáforos.

Finalmente, o tamanho da área simulada é de  $1800m \times 1800m$ , com quadras de  $300m$  de lado, o que resulta em cinco ruas verticais e cinco horizontais, sendo todas bidirecionais e com três faixas em cada sentido. Os semáforos se mantêm sempre sincronizados (exceto nos experimentos da Seção 5.2.7) e passam 60 s no estado vermelho, 3 s no amarelo e 57 s no verde.

### Gerador de Fluxos Veiculares

A função desta ferramenta é criar a especificação dos fluxos veiculares a serem usados na simulação de tráfego. Conforme mostrado na Figura 5.4, esses fluxos são caracterizados pelo período com que os veículos são inseridos no cenário de simulação e a rota que devem seguir, bem como a velocidade, aceleração e desaceleração máximas que podem desenvolver. Todos esses parâmetros são obtidos de um arquivo de configuração, escrito individualmente para cada experimento.

As rotas criadas por este gerador consistem em fazer os veículos seguirem em frente até deixarem o cenário de simulação (quando são removidos) ou dobrarem à direita em algum cruzamento, após o que prosseguem em linha reta até saírem da área simulada. Essa escolha de rotas reflete a tendência de

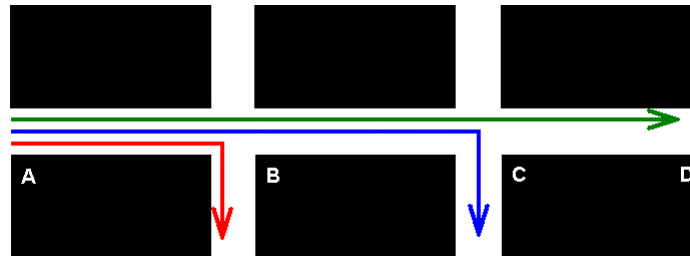


Figura 5.8: Rotas a serem seguidas por veículos injetados na topologia em *A*.

motoristas reais em dirigirem em frente na maior parte do tempo, ao invés de dobrar nos cruzamentos constantemente, em uma trajetória sinuosa.

A Figura 5.8 ilustra as rotas que seriam criadas, em um cenário reduzido, para veículos que entram nele pelo cruzamento *A*. Conforme indicado pelas setas, alguns viram à direita em *B* enquanto outros o fazem em *C* e os demais seguem adiante até deixarem o cenário, em *D*.

Para cada extremidade da área simulada (a interseção de cada rua com a fronteira do cenário de simulação) um *conjunto de rotas* similar ao ilustrado na Figura 5.8 é criado. A taxa de chegada de veículos  $f_R$  de um conjunto de  $n$  rotas  $R = \{R_1, R_2, \dots, R_n\}$  é a soma das taxas de chegadas de cada uma das rotas que o constitui, o que pode ser escrito como na Equação 5-1.

$$f_R = f_1 + f_2 + \dots + f_n \quad (5-1)$$

Contudo, em virtude de uma limitação do SUMO mencionada na Seção 5.1.1, cada rota  $R_1, R_2, \dots, R_n$  tem, obrigatoriamente, uma taxa de chegada constante, ao contrário do que ocorre em cidades reais. A simulação de taxas de chegada constantes em princípio seria menos realista do que a de distribuições arbitrárias, tal como a de Poisson, normal, etc. Contudo, o efeito causado no fluxo de veículos por essa limitação do SUMO tende a ser menos intensa para pontos mais próximos do centro da área simulada.

Esse efeito se deve principalmente aos semáforos: quando um semáforo fica vermelho para alguma rua, os veículos situados nela se aglomeram nas proximidades do cruzamento, independentemente da distribuição com que sua chegada foi modelada. Desse modo, ainda que fosse possível especificar no SUMO uma distribuição arbitrária para a chegada de veículos, ela se descaracterizaria à medida em que os semáforos fechassem para cada rua. Conseqüentemente, os efeitos da chegada de veículos com uma distribuição específica seria mais acentuado nas proximidades das bordas do cenário de simulação.

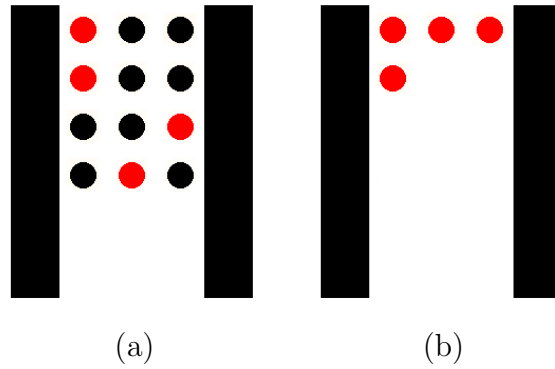


Figura 5.9: Aglomeração de veículos próximos de um cruzamento. *Veículos equipados* estão indicados em vermelho.

### Gerador de Eventos

Uma vez criados os arquivos que descrevem a topologia das ruas e os fluxos de veículos, passa-se à simulação de tráfego. Essa etapa foi realizada pelo simulador SUMO, versão 0.9.8, cuja saída é um relatório (*log*) no formato mostrado na Figura 5.5.

O gerador de eventos pode ser subdividido em um gerador de mobilidade e um gerador de pacotes. O gerador de mobilidade é responsável pela decodificação (*parsing*) do relatório produzido pelo SUMO e por converter a informação de posicionamento de cada veículo em comandos de mobilidade em OTcl que serão usados pelo NS-2 posteriormente. Contudo, tendo em vista que nem todos os veículos serão necessariamente *equipados*, o gerador de mobilidade seleciona aleatoriamente várias amostras do conjunto total de veículos para serem utilizados como estações móveis na simulação de rede. O número de veículos  $n$  de cada amostra é calculado pela Equação 5-2.

$$n = pN, \quad (5-2)$$

onde  $p$  é a *taxa de penetração* e  $N$  é o número total de veículos no experimento. Cada amostra de *veículos equipados* corresponde a uma repetição da simulação de redes sem fio, da qual os outros veículos não participam. Os detalhes referentes a essa parte da simulação são apresentados mais adiante.

Neste ponto, convém destacar que a simulação de veículos *não-equipados* é importante para se obter um padrão de tráfego mais realista dos *veículos equipados*. Em outras palavras, simular a movimentação de um total de  $N$  veículos, dos quais  $n$  são *equipados*, tende a produzir resultados diferentes dos que seriam obtidos pela simulação, exclusivamente, dos  $n$  *veículos equipados* em questão.

Essa diferença pode ser observada na Figura 5.9, que mostra um grupo de veículos próximos a um cruzamento, aguardando a abertura do semáforo. De



acordo com a Figura 5.9(a), quatro *veículos equipados* se encontram misturados a oito *não-equipados*. Quando o semáforo abrir, eles tenderão a se espalhar ao longo da rua, uma vez que os três veículos da primeira fileira poderão acelerar imediatamente ao passo em que os demais só o farão alguns instantes depois. Por outro lado, se os *veículos não-equipados* fossem eliminados, os demais se agrupariam como na Figura 5.9(b). Após a abertura do semáforo, eles tenderiam a se manter próximos uns dos outros, proporcionando resultados de simulação significativamente diferentes.

A outra subdivisão do gerador de eventos é o gerador de pacotes, responsável por criar um *script* OTcl que escalonará o envio dos pacotes que tráfegarão pela rede. Para tanto, é necessário determinar os instantes de tempo em que os pacotes serão enviados bem como seus remetentes e destinatários. Tendo em vista que o cenário de simulação se encontra inicialmente vazio, torna-se necessário determinar o instante de tempo em que o volume de tráfego na área simulada supera a etapa transiente e atinge o estado de equilíbrio.

O critério adotado nas simulações foi o de aguardar até que o primeiro veículo de cada fluxo tenha deixado o cenário de simulação. A partir desse instante, a simulação se estende por 130 s, referentes à soma do período dos semáforos (120 s) com uma prorrogação de 10 s, a fim de permitir que os últimos pacotes enviados tenham uma chance de chegar a seus destinatários. O remetente e destinatário de um pacote são sempre selecionados, respectivamente, como os veículos mais próximos do canto inferior esquerdo e do superior direito da área simulada. O objetivo desse procedimento é o de maximizar o espaço que os algoritmos de roteamento poderão utilizar para criar suas rotas.

Em uma situação real, é possível que os veículos possam se comunicar a distâncias menores, mas a simulação do roteamento em tais circunstâncias tenderia a revelar menos informações a respeito do desempenho dos algoritmos, uma vez que haveria menos combinações possíveis de *segmentos* entre remetente e destinatário para a construção de rotas. Esse problema dificulta particularmente a avaliação de desempenho da estratégia de reparo, que normalmente é acionada com maior frequência em rotas longas.

Além disso, os resultados de roteamento a distâncias curtas tendem a ser melhores do que aqueles obtidos quando remetente e destinatário estão separados por distâncias maiores. Dessa forma, se as estatísticas de desempenho dos algoritmos fossem calculadas a partir de amostras mistas (compostas tanto de resultados obtidos para rotas longas quanto para as curtas), eventuais benefícios ou deficiências dos algoritmos poderiam ser mascarados (diluídos) pelo cálculo da média e ainda contribuiriam para aumentar o desvio padrão da amostra.

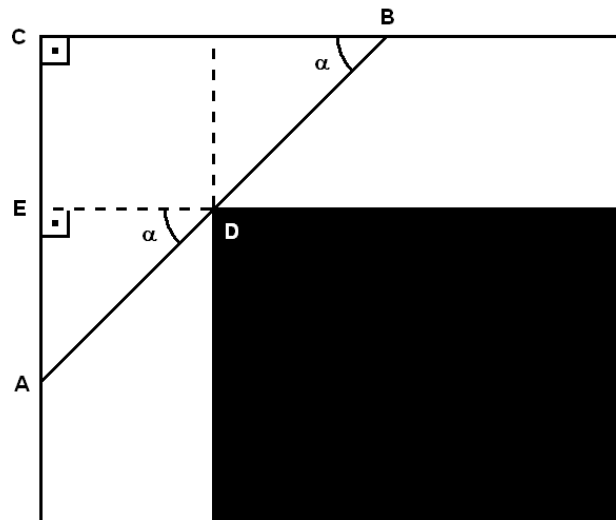


Figura 5.10: Cálculo da largura da calçada para efeito de propagação.

Por motivos similares, a maioria dos experimentos realizados consiste no envio de um único pacote por segundo, com uma seção de dados (*payload*) de 256 bytes, a fim de testar a eficácia de cada algoritmo em encontrar uma rota que atravessasse todo o cenário de simulação. Essa verificação poderia ser afetada pela existência de múltiplos pacotes na rede e as conseqüentes colisões no meio sem fio. Supondo que tal rota exista, o próximo pacote de cada fluxo poderia ser encaminhado pela rota seguida pelo seu antecessor, princípio este que poderia ser aplicado a todos os algoritmos de roteamento sendo comparados. Conforme discutido no parágrafo anterior, essa prática poderia mascarar algumas características do desempenho dos algoritmos.

Por outro lado, muitas aplicações podem precisar enviar vários pacotes a um mesmo destinatário em um curto espaço de tempo. Para verificar quais algoritmos de roteamento atenderiam a esse requisito, seria necessário simular o envio de múltiplos fluxos simultâneos de pacotes. Contudo, em virtude de limitações de tempo, esse experimento terá de ser conduzido em trabalhos futuros.

### Simulação da Rede sem Fio

A comunicação inter-veicular foi simulada pelo NS-2 versão 2.33, a partir dos *scripts* de mobilidade e de envio de pacotes discutidos nas seções anteriores. A saída do NS-2 é um relatório do qual podem ser extraídas as estatísticas de interesse, conforme discutido mais adiante.

O modelo de propagação adotado utilizado nas simulações foi aquele descrito na Seção 4.1.1. Tomando-se como referência um cruzamento entre duas ruas perpendiculares como na Figura 5.10, a largura total da calçada,

para efeito de propagação de sinal pelo modelo adotado, pode ser calculado como se segue.

Sejam  $A$  e  $B$  dois *veículos equipados* separados por uma distância igual ao *alcance efetivo de transmissão obstruída* (AETO), definido na Seção 4.1.1. No  $\triangle ABC$ , temos:

$$\overline{AC} = \overline{AB} \sin \alpha \quad (5-3)$$

E do  $\triangle ADE$  vem:

$$\overline{AE} = \overline{DE} \tan \alpha \quad (5-4)$$

Tendo em vista que:

$$\overline{AE} + \overline{CE} = \overline{AC}, \quad (5-5)$$

e substituindo as Equações 5-3 e 5-4 na Equação 5-5, tem-se:

$$\overline{DE} \tan \alpha + \overline{CE} = \overline{AB} \sin \alpha \quad (5-6)$$

Assumindo-se que as larguras de calçadas  $\overline{CE}$  e  $\overline{DE}$  tenham o mesmo valor  $w$ , a Equação 5-6 se reduz a:

$$w \tan \alpha + w = \overline{AB} \sin \alpha \quad (5-7)$$

$$w = \frac{\overline{AB} \sin \alpha}{1 + \tan \alpha} \quad (5-8)$$

De acordo com experimentos baseados em medições, utilizando interfaces de comunicação IEEE 802.11b (GON08), um valor razoável para o alcance de comunicação sem linha de visão ( $\overline{AB}$ ) é de 90 m. Contudo, tendo em vista a indisponibilidade de medições mais detalhadas a respeito do ângulo  $\alpha$ , seu valor foi arbitrado em  $45^\circ$ , resultando em  $w = 31,34m$ .

Porém, as distribuições originais do NS-2 não implementam a simulação da obstrução de sinal sem fio por prédios e nas esquinas dos cruzamentos. Esse problema foi parcialmente contornado fazendo cada veículo descartar, *explícita e deliberadamente*, qualquer pacote recebido *diretamente* de outro veículo com o qual não tiver, após o procedimento de alargamento das calçadas, a *linha de visão* desobstruída naquele momento.

Finalmente, a Tabela 5.1 resume a configuração dos parâmetros das camadas física e MAC.

### Pós-processamento

O resultado de uma simulação do NS-2 é um relatório que passa por um processo de decodificação (*parsing*) para a obtenção das informações de interesse, seguido pelo cálculo de estatísticas como média e desvio padrão. O

Tipo de antena	omnidirecional
Altura da antena	1,5 m
Potência de transmissão	0,28 W
Frequência	914 MHz
Comprimento de onda	32,82 cm
Sensibilidade da antena (CS — <i>carrier sense</i> )	$1,56 \times 10^{-11}$ W
Limiar de recepção (RxT — <i>receiving threshold</i> )	$8,92 \times 10^{-10}$ W
Alcance de transmissão	200 m
Taxa de transmissão	1 Mb/s
Alcance de interferência	549,93 m
Protocolo MAC	IEEE 802.11

Tabela 5.1: Parâmetros das camadas física e MAC.

intervalo de confiança, por sua vez, é determinado para um nível de confiança de 90% com amostras colhidas em vinte repetições.

O desempenho de cada algoritmo é resumido em um gráfico do número de pacotes entregues com sucesso em função de seus tempos de vida. Especificamente, se o ponto  $(x, y)$  pertence à curva referente ao desempenho de um algoritmo, isso significa que  $y$  pacotes chegaram ao destinatário passando por no máximo  $x$  veículos. Em outras palavras, se o tempo de vida dos pacotes fosse  $x$ , a quantidade de pacotes que seriam entregues seria  $y$ .

### 5.1.3

#### Algoritmos Simulados

Dentre os algoritmos discutidos no Capítulo 3, dois foram selecionados para a comparação de desempenho com o TLAR: o U-GPSR e o STAR. Entendemos que algoritmos elaborados como o TLAR devam oferecer um desempenho melhor do que o de um algoritmo simples, a exemplo do U-GPSR, que não armazenam nenhuma informação de estado nos veículos, e que não dispõem de dados sobre o volume de tráfego em cada rua. A comparação com o U-GPSR, portanto, permite verificar qual o ganho de desempenho obtido pelas abordagens computacionalmente mais caras em relação a uma mais simples.

Por outro lado, abordagens que empregam sensoriamento e disseminação do volume de tráfego, representadas pelo STAR, tendem a produzir resultados melhores em termos de taxa de entrega, retardo total e quantidade de veículos na rota. Desse modo, a comparação entre o STAR e o TLAR permitiria averiguar se a sobrecarga (*overhead*) causada pela disseminação da informação de tráfego compensaria uma eventual superioridade do STAR sobre o TLAR.

Finalmente, um quarto algoritmo, que dispõe da posição de *todos* os *veículos equipados* da simulação, também é analisado neste capítulo. Apesar de não ser implementável na prática, esse algoritmo, batizado *onisciente*, indicará

qual é o melhor desempenho que pode ser atingido em cada cenário de teste.

Exceto pelo algoritmo onisciente, todos os demais usam *beacons* para manterem uma lista de vizinhos atualizada. Vários experimentos foram conduzidos com diferentes períodos de envio de *beacons*, mas nenhum produziu resultados satisfatórios. Períodos curtos implicavam uma geração excessiva de *beacons*, acarretando muitas colisões de acesso ao meio sem fio e tornando a simulação lenta demais; períodos longos, por outro lado, proporcionaram uma imprecisão inaceitável à lista de vizinhos e reduziram sensivelmente o desempenho de todos os algoritmos que empregam *beacons*.

Desse modo, fica evidente que o desempenho dos algoritmos de roteamento depende fortemente desses aspectos da camada MAC. Contudo, a implementação de um mecanismo mais eficiente para a manutenção da lista de vizinhos não faz parte do escopo desta tese. Além disso, para evitar que esses problemas da camada de enlace interfiram na análise de desempenho dos protocolos da camada de rede, a abordagem baseada em *beacons* foi substituída por uma lista de vizinhos construída a partir das posições instantâneas de cada veículo na simulação.

Contudo, o uso de *beacons* tinha como efeito colateral a atualização constante da *cache* do protocolo ARP (Address Resolution Protocol), responsável pela tradução de endereços de rede em endereços MAC. Com a eliminação dos *beacons*, o protocolo ARP passou a ser acionado frequentemente, causando o envio de várias consultas (ARP\_REQUEST) a fim de descobrir o endereço MAC do próximo veículo de cada rota, cujo identificador havia sido obtido de uma lista de vizinhos construída artificialmente. Esse problema foi resolvido utilizando-se uma tabela global, que permite aos veículos obterem imediatamente o endereço MAC referente a qualquer de seus vizinhos.

Outro problema relativo aos *beacons* diz respeito ao STAR. Em sua versão original, esse algoritmo dissemina informações de tráfego fazendo-a “pegar carona” (*piggybacking*) nos *beacons*. Contudo, tendo em vista o alto custo computacional da simulação do envio regular de *beacons*, a alternativa adotada foi manter uma tabela de tráfego global única, compartilhada por todos os veículos. Apesar de não ser implementável na prática, acreditamos que essa abordagem simula condições muito favoráveis à disseminação da informação do tráfego veicular. Por esse motivo, seu desempenho serve de estimativa para o melhor desempenho possível que se poderia obter de uma implementação real do STAR.

Uma última simplificação da implementação dos experimentos foi a de assumir que o serviço de localização, empregado por todos os algoritmos de roteamento discutidos neste capítulo, possa oferecer a cada veículo a informação

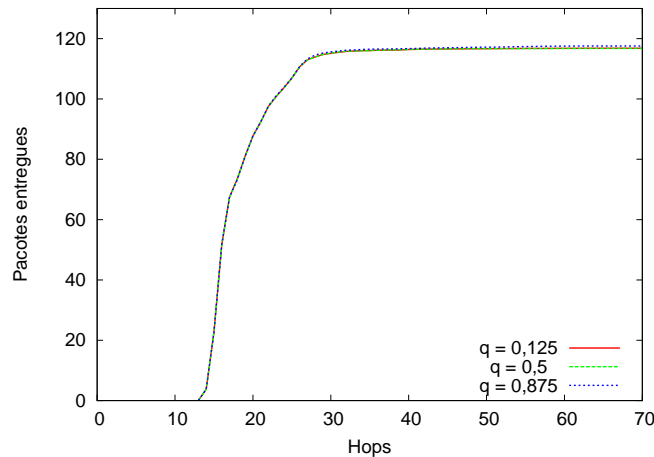


Figura 5.11: Quantidade de pacotes entregues pelo TLAR em função de seus tempos de vida para alguns valores da constante de penalidade. A *taxa de penetração* é de 40%, o intervalo de chegada de veículos é de 3s e o alcance de comunicação é de 200 m.

precisa, e com latência desprezível, sobre a localização do destinatário de cada pacote. Essa informação foi obtida consultando-se o estado global do simulador de redes sem fio, e teve por objetivo evitar que aspectos específicos dos serviços de localização influenciassem o desempenho dos algoritmos simulados.

## 5.2 Experimentos

Esta seção descreve os experimentos conduzidos com os algoritmos a fim de avaliar seus desempenhos. Primeiramente, experimentos preliminares foram conduzidos para a determinação dos principais parâmetros de simulação (5.2.1). Os demais experimentos visam avaliar o efeito da velocidade dos veículos e de outros fatores no desempenho dos algoritmos de roteamento.

### 5.2.1 Experimentos Preliminares

Dos quatro algoritmos simulados, apenas o TLAR e o STAR empregam parâmetros que precisam ser ajustados previamente. A determinação dos valores mais apropriados foi feita por experimentos preliminares conduzidos em um cenário de referência. Tendo em vista que os experimentos discutidos neste capítulo envolvem taxas de penetração de 30%–50% e que o intervalo de chegada dos veículos variou de 2 s–4 s, o cenário de referência adota uma taxa de penetração de 40% e um intervalo de chegada de 3 s.

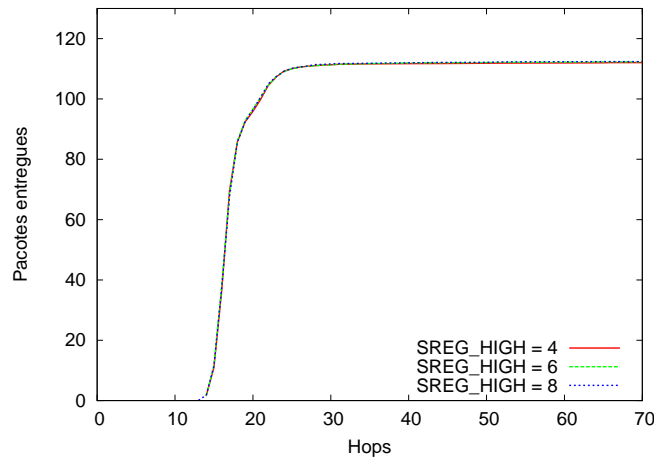


Figura 5.12: Quantidade de pacotes entregues pelo STAR em função de seus tempos de vida para alguns valores da constante SREG\_HIGH. A taxa de penetração é de 40%, o intervalo de chegada de veículos é de 3s e o alcance de comunicação é de 200 m.

## TLAR

No caso do TLAR, o único parâmetro a ser determinado é a *constante de penalidade* ( $q$ ), introduzida na Seção 4.2.6. Foram conduzidas 20 repetições para valores de  $q$  iguais a 0,125; 0,5 e 0,875. Os resultados dessas simulações se encontram na Figura 5.11, e mostram desempenho quase idêntico para as três constantes de penalidade, sendo que um valor de  $q = 0,875$  proporcionou resultados ligeiramente melhores.

## STAR

Os principais parâmetros do STAR especificam a quantidade mínima de vizinhos (constante SREG\_HIGH) que um veículo precisa possuir em determinada direção para provocar a disseminação dessa informação. Além disso, de modo a evitar flutuações de tráfego passageiras, a quantidade de vizinhos precisa permanecer acima desse limiar por um período de tempo mínimo (SPROP\_HIGH). Finalmente, cada entrada da tabela de tráfego permanece ativa por um determinado período (TIMEOUT), após o qual são removidas.

Mais uma vez, os resultados obtidos para todas essas configurações foram bastante semelhantes. Os valores adotados nas demais simulações desse capítulo foram: SREG\_HIGH = 6, SPROP\_HIGH = 5 e TIMEOUT = 10.

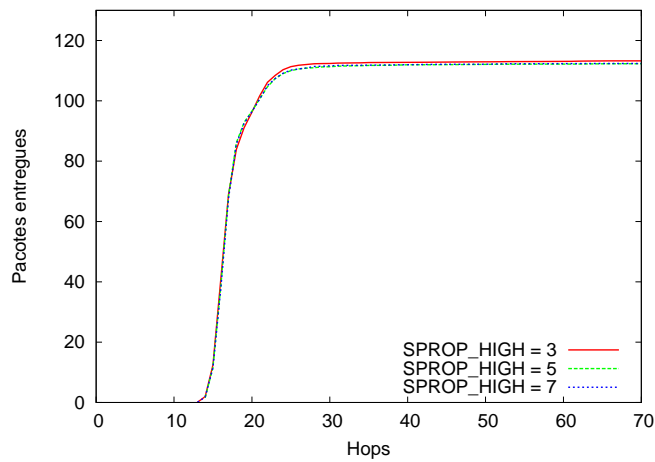


Figura 5.13: Quantidade de pacotes entregues pelo STAR em função de seus tempos de vida para alguns valores da constante SPROP\_HIGH. A taxa de penetração é de 40%, o intervalo de chegada de veículos é de 3s e o alcance de comunicação é de 200 m.

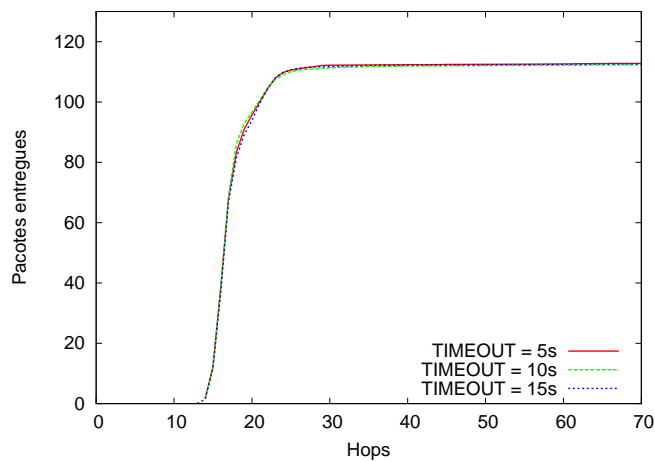


Figura 5.14: Quantidade de pacotes entregues pelo STAR em função de seus tempos de vida para alguns valores da constante TIMEOUT. A taxa de penetração é de 40%, o intervalo de chegada de veículos é de 3s e o alcance de comunicação é de 200 m.



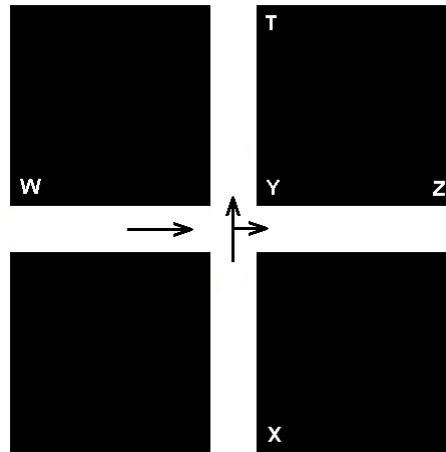


Figura 5.15: Cruzamento usado para análise das *conectividades*.

### 5.2.2

#### Cálculo das Conectividades

Conforme discutido na Seção 4.2.2, as *conectividades reta e cruzada* dependem do alcance de comunicação sem fio, da *taxa de penetração*, do volume de tráfego nas ruas e das fases dos semáforos. Primeiramente, o algoritmo para o cálculo aproximado das *conectividades* é apresentado. A seguir, alguns experimentos avaliam como a variação de cada parâmetro de simulação isoladamente as afeta.

O cenário utilizado (exclusivamente nos experimentos desta Seção) contém um único cruzamento, como mostra a Figura 5.15. Essa escolha visa eliminar a influência dos semáforos de cruzamentos próximos e limitar os fatores que poderiam interferir nas *conectividades* de interesse. Pelos mesmos motivos, o tráfego de veículos em ambas as ruas é unidirecional, fluindo da esquerda para a direita e de baixo para cima, sendo que 25% dos veículos da rua vertical (*XYT*) viram à direita no cruzamento *Y*. Desse modo, *todos* os veículos que passarem no *segmento YZ* terão vindo ora do *segmento WY* ora do *segmento XY*, conforme a fase do semáforo.

As subseções seguintes analisam a flutuação das *conectividades* do cruzamento *Y* e do *segmento YZ*. Nos gráficos apresentados, a curva do semáforo assume os valores 0 quando ele está aberto para a rua vertical (*XYT*) e 0,5 quando ele está aberto para a rua horizontal (*WYZ*). Finalmente, quando o semáforo passa para o estado amarelo nas ruas *XYT* e *WYZ*, sua curva atinge os valores 0,25 e 0,75, respectivamente.

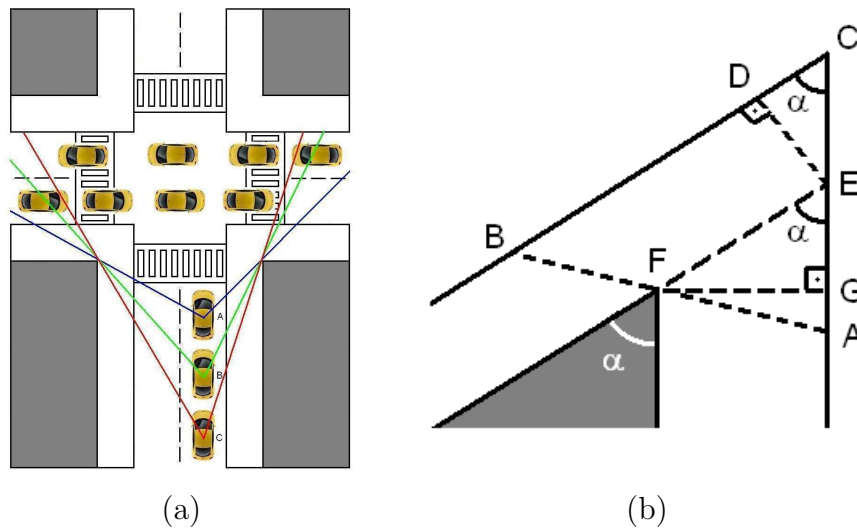


Figura 5.16: (a) Ângulo de visão dos veículos em um cruzamento. (b) Cálculo da *visibilidade cruzada*.

### Conectividade Cruzada

Conforme definido na Seção 4.2.2, a *conectividade cruzada* (CC) é a probabilidade de existirem pelo menos dois veículos *comunicáveis diretamente* entre si, mas localizados em ruas diferentes e concorrentes. Como ilustrado na Figura 5.16(a), quanto mais próximo do cruzamento um veículo estiver, maior será o número de veículos na rua transversal com os quais terá linha de visão desobstruída. Esta tese define como *visibilidade cruzada* (VC) de um veículo *A* como a distância máxima a que um veículo *B*, em uma rua transversal, possa estar do cruzamento para que esses veículos estejam em linha de visão um do outro.

Dada a distância de um veículo ao cruzamento, o ângulo entre as ruas ( $\alpha$ ) e a largura das calçadas (para efeito de propagação, conforme discutido na Seção 4.1.1), é possível calcular *VC* como na Figura 5.16(b). Nela, a região cinza representa prédios, e as linhas contínuas a trajetória dos veículos em cada rua. Seja *A* um veículo cuja *visibilidade cruzada* se deseja calcular. Por semelhança de triângulos e relações trigonométricas básicas, é possível demonstrar que:

$$VC_A = \frac{dw}{d \sin \alpha - w}, \quad (5-9)$$

onde  $VC_A$  é a *visibilidade cruzada de A*,  $w$  é a largura das calçadas (por simplicidade, suposta igual em todas as ruas), e  $d$  é a distância de *A* ao cruzamento. A Equação 5-9 só se aplica caso  $d \sin \alpha > w$ ; do contrário, *A* se encontrará entre *C* e *E*, situação na qual  $VC_A$  é definida como o comprimento da rua transversal àquela onde *A* se encontra. Além disso, a Figura 5.16(b) presume que o ângulo entre as ruas ( $\alpha$ ) é agudo; porém, por

cálculos semelhantes, é possível provar que a Equação 5-9 também é válida caso  $\alpha > 90^\circ$ .

Uma vez calculada  $VC_A$ , e conhecida a posição de cada veículo próximo ao cruzamento, seja  $V_A$  o conjunto dos que tenham linha de visão desobstruída com  $A$ , *independentemente do fato de serem ou não equipados*. Sendo  $p$  a probabilidade de um veículo ser *equipado*, pode-se calcular a probabilidade de que *pelo menos um* veículo em  $V_A$  o seja como:

$$p_A = 1 - (1 - p)^{\#V_A}, \quad (5-10)$$

onde  $\#V_A$  é a cardinalidade de  $V_A$ .

De posse da Equação 5-10, pode-se calcular a *conectividade cruzada* conforme descrito a seguir. Seja  $R$  uma rua em que existam vários veículos, aos quais são atribuídos identificadores  $V_k, k \in \{0, 1, \dots\}$  de acordo com a distância que os separa ao cruzamento com uma rua  $S$ , tal que  $V_0$  seja o veículo mais próximo a esse cruzamento,  $V_1$  o segundo mais próximo, e assim por diante. Se  $V_0$  for *equipado*, e existir *pelo menos um veículo comunicável diretamente* com ele em  $S$ , haverá conectividade nesse cruzamento. Contudo, se esse veículo não for *equipado*, a conectividade cruzada passa a depender da existência de outro veículo em  $R$  capaz de se comunicar com algum veículo em  $S$ . Generalizando esse raciocínio, se os  $i$  veículos mais próximos do cruzamento não forem *equipados*, a comunicação através do cruzamento em questão irá requerer a existência de um veículo  $V_k, k \geq i$  que seja *equipado* e possua pelo menos um vizinho em  $S$ , o que pode ser expresso pela seguinte recursão:

$$CC_k = \begin{cases} p \times p_{V_k} + (1 - p) \times CC_{k+1}, & \text{se } k < n \\ 0, & \text{caso contrário} \end{cases} \quad (5-11)$$

onde  $p$  é a probabilidade de um veículo ser equipado,  $p_{V_k}$  é a probabilidade de haver pelo menos um *veículo comunicável diretamente* com  $V_k$  em  $S$  (Equação 5-10),  $n$  é o número de veículos em  $R$  e  $CC_k$  é conectividade cruzada, dado que os  $k$  veículos de  $R$  mais próximos do cruzamento *não são equipados*. Assim, calcular  $CC$  em qualquer cruzamento equivale a calcular  $CC_k$ , conforme a Equação 5-11, com  $k = 0$ .

É importante notar que a definição de conectividade cruzada é relativa à condição onde não haja colisões no meio sem fio. Desse modo, em uma situação real, a transmissão de dados por um cruzamento pode falhar mesmo quando  $CC(t) = 100\%$ . Além disso, o algoritmo de cálculo apresentado na Equação 5-11 assume que o modelo de propagação utilizado é aquele discutido na Seção 4.1.1. O uso de outros modelos requer modificações na Equação 5-11, a fim de levar em consideração efeitos como o não-determinismo na propagação de

sinal, previsto em modelos como o de *shadowing*.

O comportamento da *conectividade cruzada* é mostrado para algumas combinações de parâmetros na Figura 5.17. É possível notar que todas as curvas têm o mesmo período que a do semáforo e que mostram uma melhor *conectividade* para intervalos de chegada ( $T$ ) mais curtos (Figura 5.17(a)) e taxas de penetração ( $p$ ) maiores (Figura 5.17(b)), uma vez que ambas as condições aumentam o número de *veículos equipados* nas proximidades do cruzamento. Contudo, em virtude da obstrução do sinal de rádio pelos prédios adjacentes ao cruzamento  $Y$ , o aumento do alcance da comunicação tem pouco efeito sobre a *conectividade cruzada* (Figura 5.17(c)), principalmente se comparado ao impacto causado na *conectividade reta*, discutida a seguir.

### Conectividade Reta

Reconsideremos a definição da *conectividade reta* apresentada na Seção 4.2.2, que consiste na probabilidade de dois veículos *equipados*, localizados nas extremidades opostas de um *segmento*, estarem *comunicáveis indiretamente* entre si, através, *exclusivamente*, de veículos situados no *segmento* em questão.

No caso trivial, se o comprimento desse *segmento* for inferior ao alcance de comunicação sem fio, pode-se concluir que sua *conectividade reta* será de 100%, independentemente da existência de outros veículos. Ou seja,

$$CR(A, B) = 100\%, \quad \text{se } \overline{AB} < r, \quad (5-12)$$

onde  $r$  é o alcance da comunicação sem fio e as extremidades do *segmento* em questão são  $A$  e  $B$ .

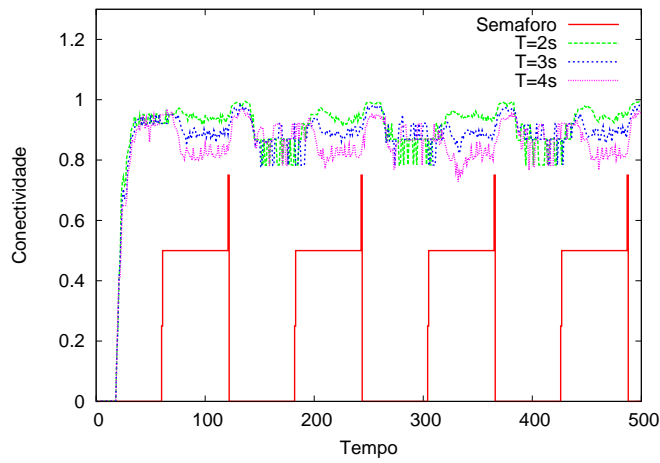
Geralmente, o comprimento dos *segmentos* é maior que o alcance de comunicação sem fio. Nesse caso, a *conectividade* desses *segmentos* depende da existência de veículos capazes de encaminhar pacotes através deles.

Consideremos um *segmento*  $AB$  (tal que  $\overline{AB} > r$ ) cuja *conectividade* se deseja calcular. Se  $Q$  é o veículo mais distante de  $A$  em  $AB$ , a *conectividade* de  $AB$  pode ser calculada pela Equação 5-13. Caso  $A$  não possua veículos *alcancáveis diretamente* em  $AB$ , a *conectividade* desse segmento será nula.

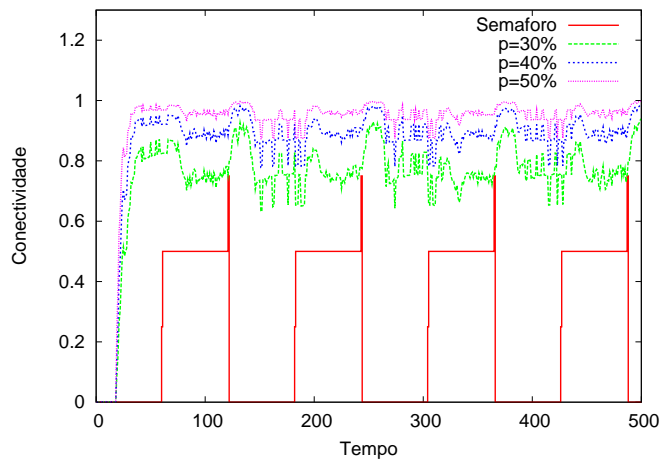
$$CR(AB, N) = p \times CR(QB, N) + (1 - p) \times CR(AB, N \cup \{Q\}), \quad (5-13)$$

onde  $CR(AB)$  é a *conectividade reta* de  $AB$  dado que nenhum veículo do conjunto  $N$  é *equipado* e  $p$  é a probabilidade de um veículo ser *equipado*.

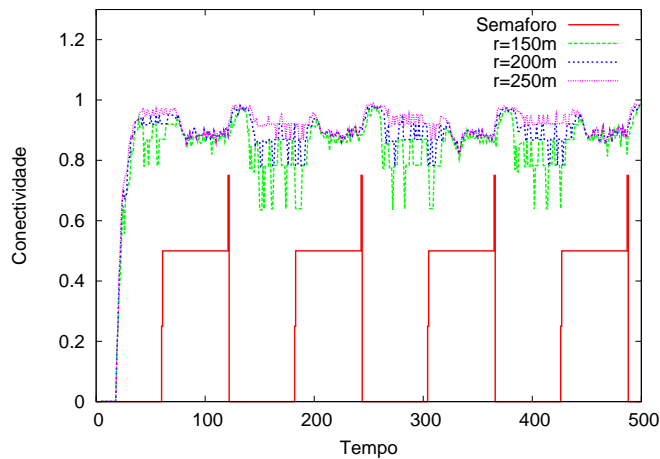
Ou seja, se  $Q$  for um *veículo equipado*, a *conectividade* de  $AB$  dependerá da *conectividade* do segmento  $QB$  (primeiro termo da Equação 5-13). Do



(a) Taxa de penetração: 40%, alcance de comunicação: 200m.



(b) Intervalo de chegada: 3s, alcance de comunicação: 200m.



(c) Taxa de penetração: 40%, intervalo de chegada: 3s

Figura 5.17: Efeito da variação dos parâmetros de simulação na *conectividade cruzada*: (a) Intervalo de chegada de veículos (b) Taxa de penetração (c) Alcance de comunicação.

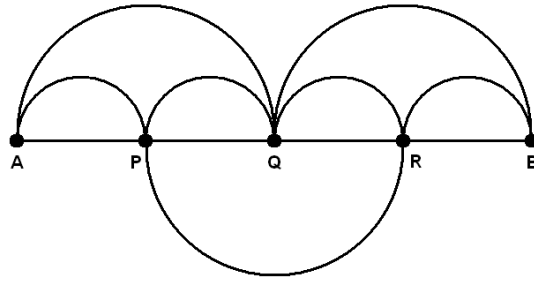


Figura 5.18: Calculando a *conectividade reta* de  $AB$ .

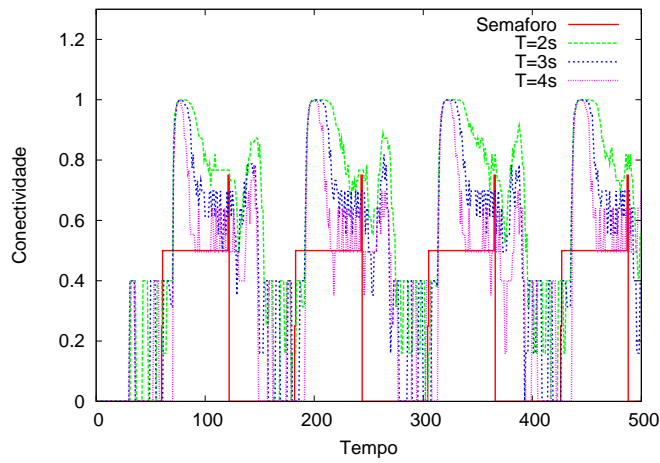
contrário,  $Q$  é desconsiderado e a Equação 5-13 é aplicada recursivamente (segundo termo). Finalmente, por definição,  $CR(AB) = CR(AB, \{\})$ .

A Figura 5.18 mostra um segmento onde cada par de veículos está ligado por um arco se, e somente se, estiverem *alcançáveis diretamente* entre si. Observa-se, por exemplo, que os veículos *alcançáveis diretamente* por  $A$  são  $P$  e  $Q$ . Se  $Q$  for *equipado*, haverá *conectividade reta* nesse segmento independentemente de  $P$  e  $R$ , uma vez que  $B$  é *alcançável* a  $Q$ . Por outro lado, se  $Q$  não for *equipado* mas  $P$  e  $R$  forem, a *conectividade reta* também será conseguida, uma vez que  $A$  é *alcançável diretamente* a  $P$  e este a  $R$ , que por sua vez é *alcançável diretamente* a  $B$ . Aplicando-se as Equações 5-12 e 5-13, conclui-se que  $CR(AB) = p + (1 - p)p^2$ .

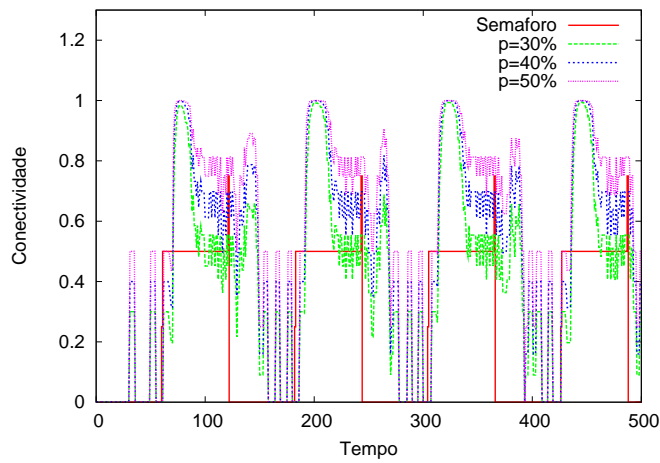
Mais uma vez, cabe a observação de que o conceito de *conectividade reta* não considera eventuais colisões no meio sem fio. Adicionalmente, o algoritmo de cálculo assume que o modelo de propagação de sinal seja o apresentado na Seção 4.1.1. A modelagem da propagação por abordagens alternativas normalmente irá requerer modificações nas Equações 5-12 e 5-13.

A dependência dos parâmetros de simulação na *conectividade reta* é mostrada na Figura 5.19. Especificamente, o aumento do intervalo de chegada dos veículos diminui a *conectividade* (Figura 5.19(a)), pois haverá menos veículos nas ruas a cada instante de tempo. Desses veículos, o percentual dos que são *equipados* determinará quais poderão participar do roteamento, o que se traduz em maior *conectividade* para maiores percentuais (Figura 5.19(b)). Todavia, ao contrário do que ocorre com a *conectividade cruzada*, o alcance de comunicação afeta sensivelmente a *conectividade reta*, pois seu aumento implica um maior número de veículos *comunicáveis diretamente* entre si em um mesmo *segmento*, como pode ser visto na Figura 5.19(c).

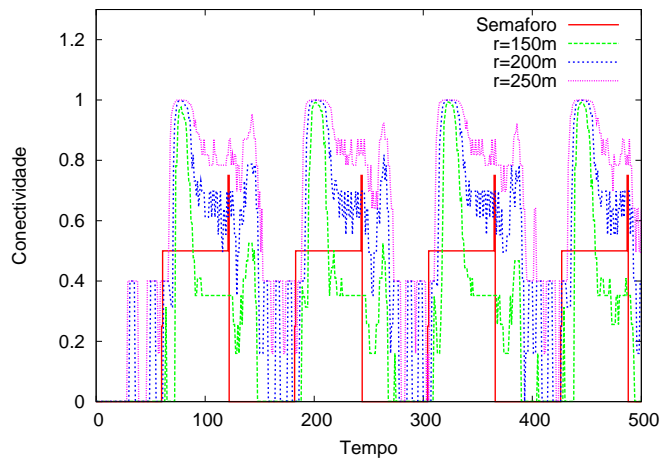
As Figuras 5.19(a)–5.19(c) mostram, ainda, que a *conectividade reta* sempre assume valores maiores quando o semáforo está aberto para a rua horizontal. Isso ocorre porque, nesse momento, todos os veículos que chegaram ao cruzamento vindos de  $W$  passam por  $YZ$ . Por outro lado, quando ele abre



(a) Taxa de penetração: 40%, alcance de comunicação: 200m.



(b) Intervalo de chegada: 3s, alcance de comunicação: 200m.



(c) Taxa de penetração: 40%, intervalo de chegada: 3s

Figura 5.19: Efeito da variação dos parâmetros de simulação na *conectividade reta*: (a) Intervalo de chegada de veículos (b) Taxa de penetração (c) Alcance de comunicação.

Algoritmo	Onisciente	U-GPSR	STAR	TLAR
Pacotes entregues				
média	119,95	110,60	96,90	108,90
desvio padrão	2,69	4,75	12,03	5,46
intervalo de confiança	[118, 91; 120, 99]	[108, 77; 112, 43]	[92, 26; 101, 54]	[106, 79; 111, 01]
Retardo total				
média [ms]	63,59	98,14	82,38	89,65
desvio padrão [ms]	7,18	56,97	19,13	31,82
intervalo de confiança [ms]	[63, 35; 63, 83]	[96, 15; 100, 13]	[81, 67; 83, 09]	[88, 53; 90, 77]
Número de hops				
média	15,18	22,68	18,90	20,74
desvio padrão	1,16	13,06	4,15	7,27
intervalo de confiança	[15, 14; 15, 22]	[22, 22; 23, 14]	[18, 74; 19, 06]	[20, 48; 21, 00]

Tabela 5.2: Desempenho dos algoritmos para taxa de penetração de 30%, alcance de transmissão de 200 m e intervalo de chegada de veículos de 3s.

para a rua vertical, somente os veículos que dobram à direita em  $Y$  (25%) passam pelo segmento  $YZ$ . Finalmente, em todos os experimentos da Figura 5.19, a variação de fase do semáforo é um dos fatores que mais influencia a conectividade reta.

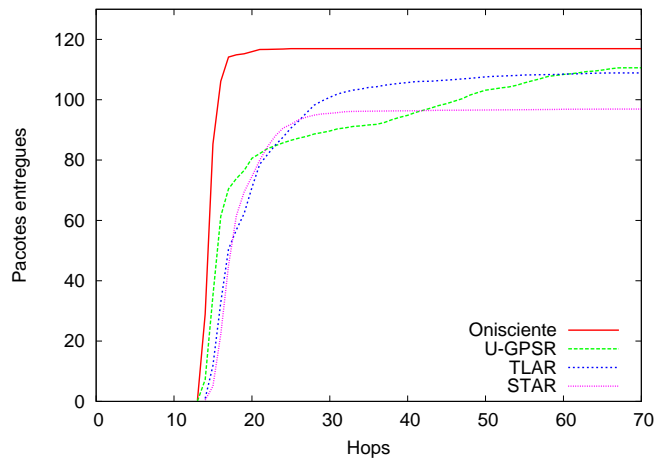
### 5.2.3

#### Taxa de Penetração

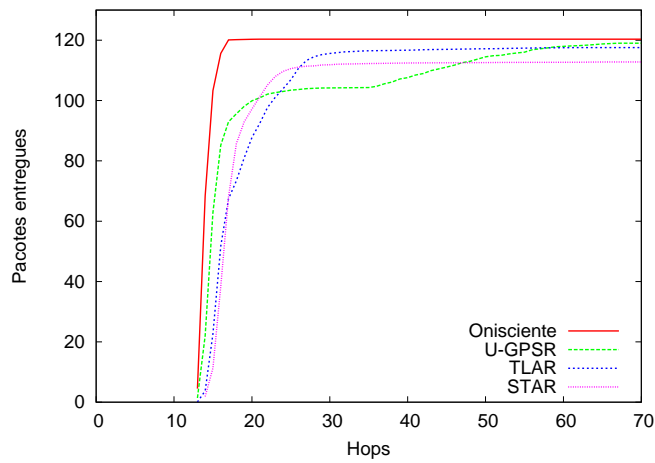
Conforme discutido ao longo desta tese, a taxa de penetração é um fator fundamental para o sucesso do roteamento em VANETs. Com o objetivo de investigar qual é o impacto da variação da taxa de penetração no desempenho dos algoritmos de roteamento, este experimento simula cenários com taxas de 30%, 40% e 50%, mantendo-se constantes os demais parâmetros da simulação. Especificamente, o intervalo de chegada de veículos em cada rua é de 3s, o alcance de comunicação é de 200 m e a velocidade máxima dos veículos é selecionada aleatoriamente no intervalo  $[15, 20]m/s$ .

Um detalhe importante sobre este experimento é que, se um veículo  $V$  é equipado em um cenário  $i$  (cuja taxa de penetração é  $p_i$ ), ele também será equipado em qualquer outro cenário  $j$  (cuja taxa de penetração é  $p_j$ ) em que  $p_j \geq p_i$ . Dessa forma, todos os veículos equipados do cenário  $p = 30\%$  também serão equipados nos cenários  $p = 40\%$  e  $p = 50\%$ . Conseqüentemente, o aumento da taxa de penetração necessariamente aumenta o número de veículos equipados em qualquer segmento do cenário de simulação, salvo naqueles em que todos os veículos já são equipados. Sem essa garantia, seria possível que, mesmo em se aumentando a taxa de penetração, as amostras resultantes

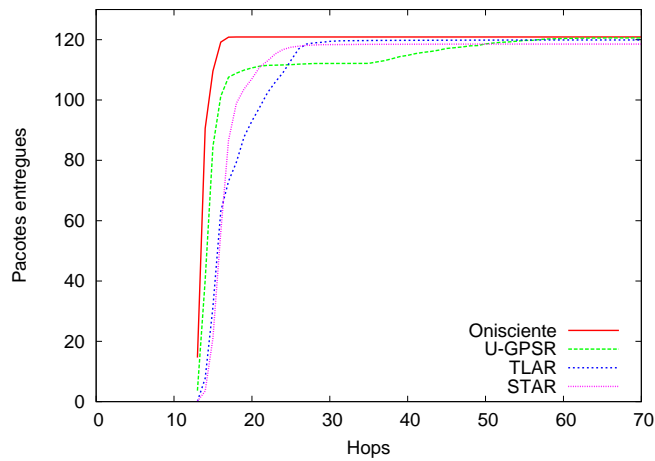




(a) Taxa de penetração: 30%



(b) Taxa de penetração: 40%



(c) Taxa de penetração: 50%

Figura 5.20: Quantidade de pacotes entregues em função de seus tempos de vida para algumas *taxas de penetração*. O intervalo de chegada de veículos é de 3s e o alcance de comunicação é de 200 m.

Algoritmo	Onisciente	U-GPSR	STAR	TLAR
Pacotes entregues				
média	120,35	119,00	112,90	117,55
desvio padrão	1,15	2,10	6,13	2,78
intervalo de confiança	[119, 91; 120, 79]	[118, 19; 119, 81]	[110, 54; 115, 26]	[116, 48; 118, 62]
Retardo total				
média [ms]	61,21	85,21	78,23	81,20
desvio padrão [ms]	7,06	48,05	16,30	21,56
intervalo de confiança [ms]	[60, 97; 61, 45]	[83, 59; 86, 83]	[77, 67; 78, 79]	[80, 47; 81, 93]
Número de hops				
média	14,58	19,67	17,91	18,75
desvio padrão	0,89	10,98	3,45	4,78
intervalo de confiança	[14, 55; 14, 61]	[19, 30; 20, 04]	[17, 79; 18, 03]	[18, 59; 18, 91]

Tabela 5.3: Desempenho dos algoritmos para *taxa de penetração* de 40%, alcance de transmissão de 200 m e intervalo de chegada de veículos de 3s.

Algoritmo	Onisciente	U-GPSR	STAR	TLAR
Pacotes entregues				
média	120,90	120,60	118,55	119,90
desvio padrão	0,30	0,58	3,85	1,76
intervalo de confiança	[120, 78; 121, 02]	[120, 38; 120, 82]	[117, 06; 120, 04]	[119, 22; 120, 58]
Retardo total				
média [ms]	59,98	74,61	75,04	78,68
desvio padrão [ms]	7,64	35,69	12,47	18,71
intervalo de confiança [ms]	[59, 72; 60, 24]	[73, 41; 75, 81]	[74, 62; 75, 46]	[78, 05; 79, 31]
Número de hops				
média	14,24	17,21	17,17	18,04
desvio padrão	0,82	8,09	2,43	3,79
intervalo de confiança	[14, 21; 14, 27]	[16, 94; 17, 48]	[17, 09; 17, 25]	[17, 91; 18, 17]

Tabela 5.4: Desempenho dos algoritmos para *taxa de penetração* de 50%, alcance de transmissão de 200 m e intervalo de chegada de veículos de 3s.

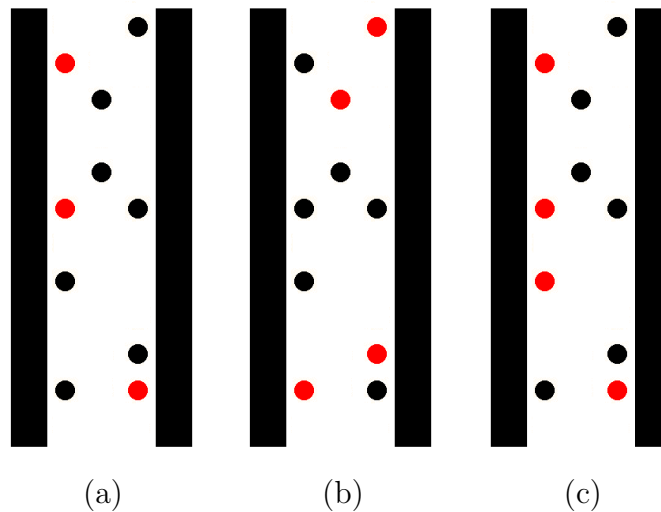


Figura 5.21: Amostras de *veículos equipados* para algumas taxas de penetração ( $p$ ). *Veículos equipados* são mostrados em vermelho. (a)  $p = 30\%$ . (b)  $p = 40\%$ . (c)  $p = 40\%$  (superconjunto da amostra de (a)).

possam comprometer a propagação de um pacote ao longo de algum *segmento*, como mostra a Figura 5.21.

A Figura 5.21(a) ilustra uma amostra em que a taxa de penetração ( $p$ ) é de 30% e a propagação de pacotes ao longo da rua é possível (os *veículos equipados* estão destacados em vermelho). Para  $p = 40\%$ , duas amostras (Figuras 5.21(b) e 5.21(c)) apresentam o mesmo número de *veículos equipados* mas com diferentes distribuições ao longo da rua. Na Figura 5.21(b), apesar de haver um *veículo equipado* a mais do que na Figura 5.21(a), a propagação de um pacote deixa de ser possível ao longo dessa rua. Por outro lado, a amostra da Figura 5.21(c) é um superconjunto da exibida na Figura 5.21(a), e por esse motivo, a propagação de pacotes também é possível ao longo dessa rua.

Neste experimento foi simulado um total de 4186 veículos (incluindo os não-equipados). O tráfego de veículos superou a etapa transiente no instante  $t = 554$ , momento a partir do qual a média de veículos na área simulada foi de 1710. A média de veículos equipados (após a etapa transiente) foi de 516, 689 e 859, para as taxas de penetração de 30%, 40% e 50%, respectivamente.

Os resultados do experimento se encontram na Figura 5.20 e nas Tabelas 5.2, 5.3 e 5.4. Os gráficos e valores obtidos confirmam uma melhoria no desempenho de todos os algoritmos de roteamento à medida que a taxa de penetração aumenta de 30% para 40% e, a seguir, para 50%. Esse efeito se deve ao aumento do número de *veículos equipados* em virtude da adoção de taxas de penetração cada vez maiores, e pode ser observado tanto pelo aumento do número de pacotes entregues e quanto pela diminuição do comprimento das rotas (em *hops*) e dos retardos dos pacotes.

Também é possível observar que a média de pacotes entregues por todos algoritmos de roteamento converge para o número de pacotes enviados (121) para uma taxa de penetração de 40%. Mesmo a uma taxa de apenas 30%, os algoritmos TLAR e U-GPSR atingem uma boa taxa média de entrega (cerca de 110 pacotes), enquanto o STAR exibe um desempenho um pouco inferior (96 pacotes). Para todas as taxas de penetração, a média de pacotes entregues pelo U-GPSR supera a obtida pelo TLAR para rotas curtas, ao passo que essa tendência se inverte para as rotas mais longas. De acordo com a Figura 5.20, essa inversão ocorre para rotas com comprimentos entre 20 e 25 *hops*. A superioridade do U-GPSR sobre o TLAR para rotas curtas aumenta com o crescimento da taxa de penetração, devido à maior densidade de *veículos equipados* e, conseqüentemente, menor probabilidade de a heurística construtiva gulosa conduzir um pacote a ótimos locais.

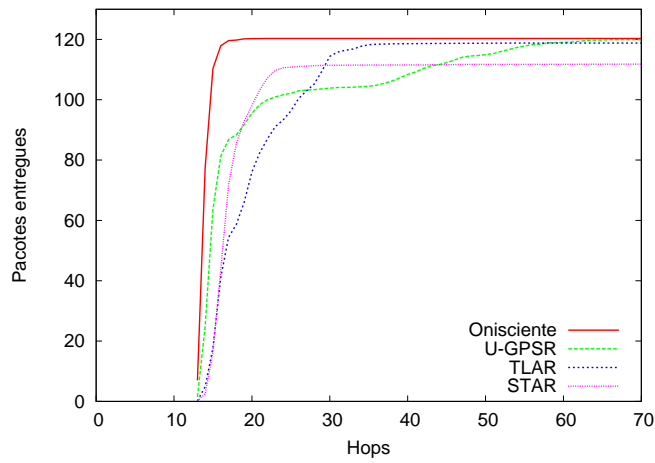
Por outro lado, o desempenho dos algoritmos de roteamento para rotas mais longas depende principalmente de suas respectivas estratégias de reparo. Os resultados de simulação mostram que, apesar de o TLAR e o U-GPSR terem obtido valores similares para as médias de pacotes entregues, o desvio padrão do comprimento das rotas (em *hops*) do U-GPSR é significativamente maior do que o valor correspondente apresentado pelo TLAR. Esse fato também se reflete na Figura 5.20, onde as curvas referentes ao comprimento das rotas do U-GPSR requerem um maior número de *hops* para atingirem o mesmo número de pacotes entregues pelo TLAR. Dessas observações é possível concluir que, nas condições simuladas, a estratégia de reparo do TLAR apresentou um desempenho superior a do U-GPSR.

O mesmo raciocínio pode ser aplicado aos resultados obtidos pelo STAR, que apresenta a menor média de entrega de pacotes em todos os cenários deste experimento. Por outro lado, de acordo com as Tabelas 5.2, 5.3 e 5.4, o comprimento das rotas (em *hops*) possui um desvio padrão menor que os obtidos pelo TLAR e U-GPSR. Isso significa que os pacotes entregues com sucesso pelo STAR percorrem rotas tipicamente curtas, sendo os demais descartados, o que é decorrente de uma estratégia de reparo pouco eficiente.

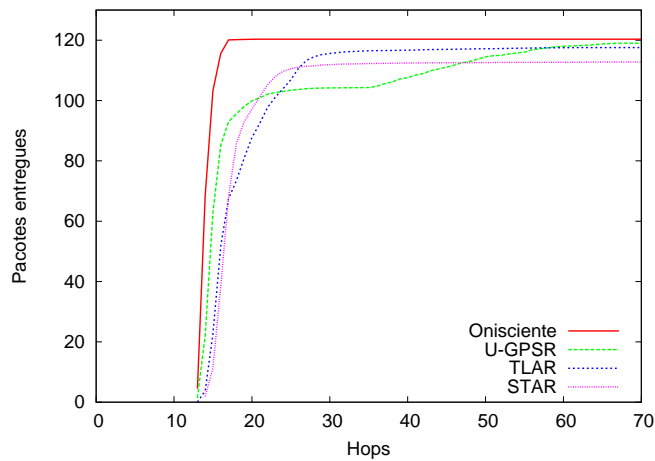
#### 5.2.4

##### Volume de Tráfego

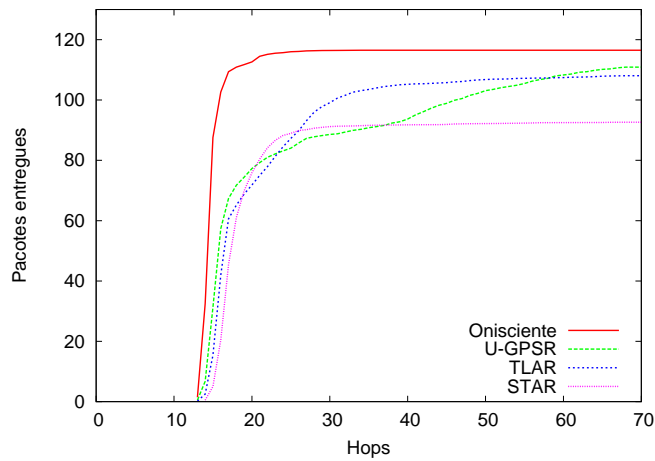
O objetivo deste experimento é analisar o desempenho dos algoritmos de roteamento em face a diferentes volumes de tráfego, mantendo-se os demais parâmetros de simulação inalterados. Os cenários considerados envolvem intervalos de chegada de veículos de 2s, 3s e 4s, a uma taxa de penetração de 40% e com um alcance de comunicação sem fio de 200 m. A velocidade



(a) Intervalo de chegada: 2s.



(b) Intervalo de chegada: 3s.



(c) Intervalo de chegada: 4s.

Figura 5.22: Quantidade de pacotes entregues em função de seus tempos de vida para alguns intervalos de chegada de veículos. A taxa de penetração é de 40% e o alcance de comunicação é de 200 m.

Algoritmo	Onisciente	U-GPSR	STAR	TLAR
Pacotes entregues				
média	120,30	119,90	111,80	118,80
desvio padrão	1,19	1,48	8,59	2,46
intervalo de confiança	[119, 84; 120, 76]	[119, 33; 120, 47]	[108, 49; 115, 11]	[117, 85; 119, 75]
Retardo total				
média [ms]	60,67	86,78	76,70	87,37
desvio padrão [ms]	7,62	47,55	14,44	23,98
intervalo de confiança [ms]	[60, 41; 60, 93]	[85, 18; 88, 38]	[76, 20; 77, 20]	[86, 56; 88, 18]
Número de hops				
média	14,41	20,07	17,56	20,15
desvio padrão	0,83	10,92	2,92	5,30
intervalo de confiança	[14, 38; 14, 44]	[19, 70; 20, 44]	[17, 46; 17, 66]	[19, 97; 20, 33]

Tabela 5.5: Desempenho dos algoritmos para intervalo de chegada de veículos de 2s, taxa de penetração de 40% e alcance de transmissão de 200 m.

Algoritmo	Onisciente	U-GPSR	STAR	TLAR
Pacotes entregues				
média	120,35	119,00	112,90	117,55
desvio padrão	1,15	2,10	6,13	2,78
intervalo de confiança	[119, 91; 120, 79]	[118, 19; 119, 81]	[110, 54; 115, 26]	[116, 48; 118, 62]
Retardo total				
média [ms]	61,21	85,21	78,23	81,20
desvio padrão [ms]	7,06	48,05	16,30	21,56
intervalo de confiança [ms]	[60, 97; 61, 45]	[83, 59; 86, 83]	[77, 67; 78, 79]	[80, 47; 81, 93]
Número de hops				
média	14,58	19,67	17,91	18,75
desvio padrão	0,89	10,98	3,45	4,78
intervalo de confiança	[14, 55; 14, 61]	[19, 30; 20, 04]	[17, 79; 18, 03]	[18, 59; 18, 91]

Tabela 5.6: Desempenho dos algoritmos para intervalo de chegada de veículos de 3s, taxa de penetração de 40% e alcance de transmissão de 200 m.

Algoritmo	Onisciente	U-GPSR	STAR	TLAR
Pacotes entregues				
média	116,50	110,90	92,65	108,05
desvio padrão	3,99	5,08	10,37	8,22
intervalo de confiança	[114, 96; 118, 04]	[108, 94; 112, 86]	[88, 65; 96, 65]	[104, 88; 111, 22]
Retardo total				
média [ms]	64,18	100,63	81,65	88,89
desvio padrão [ms]	9,18	58,79	20,42	33,37
intervalo de confiança [ms]	[63, 87; 64, 49]	[98, 58; 102, 68]	[80, 87; 82, 43]	[87, 71; 90, 07]
Número de hops				
média	15,32	23,23	18,72	20,53
desvio padrão	1,79	13,29	4,48	7,57
intervalo de confiança	[15, 26; 15, 38]	[22, 77; 23, 69]	[18, 55; 18, 89]	[20, 26; 20, 80]

Tabela 5.7: Desempenho dos algoritmos para intervalo de chegada de veículos de 4s, taxa de penetração de 40% e alcance de transmissão de 200 m.

máxima dos veículos foi selecionada aleatoriamente do intervalo [15, 20], e a etapa transiente do tráfego veicular terminou, respectivamente, nos instantes  $t = 573s$ ,  $t = 554s$  e  $t = 554s$ . As quantidades médias de veículos foram de 1770, 1710 e 1457, dos quais 711, 689 e 581, respectivamente foram *equipados*.

Os resultados deste experimento se encontram na Figura 5.22 e nas Tabelas 5.5, 5.6 e 5.7. Os valores obtidos confirmam a piora do desempenho dos algoritmos à medida em que o intervalo de chegada de veículos aumenta, uma vez que isso proporciona uma redução no número de *veículos equipados* na área simulada. O algoritmo cujo desempenho sofreu maior degradação foi o STAR, com uma nítida redução da média de pacotes entregues quando o intervalo de chegada de veículos é de 4 s. As curvas da Figura 5.22 mostram que a maioria dos pacotes entregues com sucesso pelo STAR seguiram por rotas com até 25 hops de extensão, o que explica o menor desvio padrão dessa métrica em relação aos valores obtidos pelo TLAR e pelo U-GPSR para intervalos de chegada de veículos de 4 s (4,48 hops contra 7,57 e 13,29 hops, respectivamente).

A Figura 5.22 também mostra que o desempenho do U-GPSR, em termos da média de pacotes entregues com sucesso, supera o do TLAR para rotas mais curtas (até 25 hops) quando o intervalo de chegada de veículos é de 2 s (Figura 5.22(a)) ou 3 s (Figura 5.22(b)). Nesses mesmos cenários a situação se inverte a favor do TLAR para rotas mais longas, sendo que ambos os algoritmos convergem para a taxa de entrega atingida pelo onisciente à medida em que o tempo de vida dos pacotes (em hops) aumenta.

Por outro lado, quando o intervalo de chegada é de 4 s, todos os algoritmos (exceto o onisciente) apresentam o mesmo desempenho em termos

Tabela	$p$	$T$	$p/T$
5.2	0,3	3	0,1
5.3	0,4	3	0,133
5.4	0,5	3	0,167
5.5	0,4	2	0,2
5.6	0,4	3	0,133
5.7	0,4	4	0,1

Tabela 5.8: Taxas de chegada de *veículos equipados* nos experimentos.

de entrega de pacotes para rotas de até 25 *hops*, limiar a partir do qual o desempenho do TLAR se mostra melhor. Essa tendência só se modifica para rotas maiores do que 60 *hops*, ocasião em que o U-GPSR passa a ter resultados ligeiramente melhores. Uma observação importante que pode ser feita na Figura 5.22(c) é que a maioria dos pacotes entregues pelo TLAR percorreu rotas de até 30 *hops*, e que a taxa de entrega de pacotes não converge para o valor obtido pelo algoritmo onisciente. Esses fatos apontam para a necessidade de se melhorar o desempenho da estratégia de reparo do TLAR para cenários onde haja um menor volume de tráfego, a fim de que a curva que resume seu desempenho convirja em menos *hops* para a obtida pelo algoritmo onisciente.

De posse dos resultados deste experimento e dos obtidos na Seção 5.2.3, a próxima discussão se refere ao desempenho dos algoritmos em relação à quantidade de *veículos equipados* na rede. Se um veículo chega à área simulada a cada  $T$  segundos e a taxa de penetração é  $p$ , pode-se concluir que  $p/T$  *veículos equipados* chegam a cada segundo. Como a tabela 5.8 mostra, chegam 0,2 *veículos equipados* por segundo (ve/s) no experimento a que a Tabela 5.5 se refere, contra apenas 0,167 ve/s do experimento resumido na Tabela 5.4.

A Tabela 5.8 também mostra que dois experimentos tiveram a mesma taxa de chegada de *veículos equipados*: 0,1 ve/s. Contudo, ao contrário do que era esperado anteriormente (Seção 5.1.2, na discussão sobre o gerador de eventos), esses experimentos (Tabelas 5.2 e 5.7) apresentaram resultados bastante semelhantes.

Em princípio, a similaridade dos resultados desses dois experimentos poderia ser atribuída à heterogeneidade das características de velocidade e aceleração máximas dos veículos. Assim, se dois veículos inicialmente juntos apresentarem diferentes acelerações e velocidades máximas, então eles tenderão a se separarem. Tal observação, em princípio, iria de encontro à suposição feita na Seção 5.1.2 (ilustrada na Figura 5.9), segundo a qual veículos emparelhados, ao aguardarem a abertura de um semáforo, tenderiam a permanecer próximos um do outro a partir de então. Contudo, uma análise mais aprofundada do impacto da simulação de veículos *não-equipados* no desempenho dos algoritmos



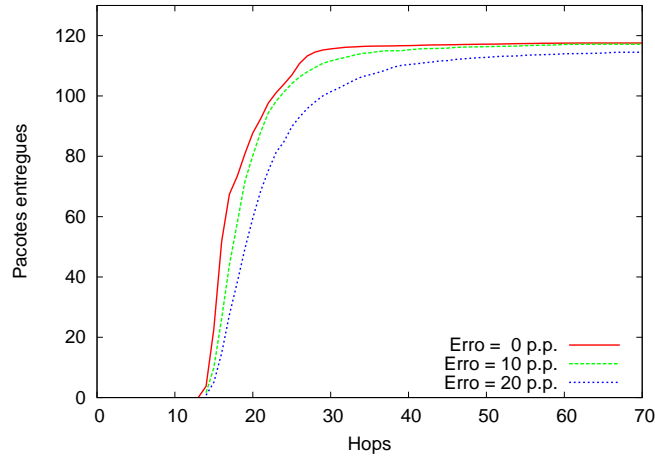


Figura 5.23: Quantidade de pacotes entregues em função de seus tempos de vida para algumas margens de erro  $\epsilon$ . O intervalo de chegada de veículos é de 3s, a taxa de penetração é de 40% e o alcance de comunicação é de 200 m.

de roteamento exigiria experimentos adicionais, que terão de ser conduzidos em trabalhos futuros devido a limitações de tempo.

### 5.2.5 Impacto do Erro nas Conectividades

Conforme explicado na Seção 4.2.1, o TLAR assume que os veículos disponham de uma base de dados com os valores das *conectividades* ao longo do tempo. Os experimentos deste capítulo, além disso, também pressupõem que esses valores possuam um erro desprezível, e que *todos* os veículos disponham de bases de dados idênticas.

Infelizmente, essas suposições fazem parte de um modelo *teórico* cuja execução é inviável na prática. As *conectividades* de uma rua ou cruzamento quaisquer podem não apenas estar sujeitas a uma margem significativa de erro, como também assumir valores diferentes em veículos distintos. Este experimento avalia a robustez do TLAR a possíveis disparidades dos valores das *conectividades* armazenadas na memória de cada veículo. Para esse propósito, cada veículo aplica um erro aleatório, uniformemente distribuído na faixa de  $[-\epsilon, +\epsilon]$ , às *conectividades* de sua base de dados.

Os valores adotados para  $\epsilon$  foram 0, 10 e 20 pontos percentuais, com uma taxa de penetração de 40%, intervalo de chegada de veículos de 3s e alcance de comunicação de 200m. A velocidade máxima dos veículos foi selecionada do intervalo  $[15, 20]m/s$ , sendo que ao todo foram simulados 4186 veículos (incluindo os *não-equipados*) e que o volume de tráfego veicular atingiu o equilíbrio no instante de tempo  $t = 554$ . A partir de então trafegaram pela área de simulação, em média, 1710 veículos, dos quais 689 eram *equipados*. Os

Erro	0 p.p.	10 p.p.	20 p.p.
Pacotes entregues			
média	117,55	117,15	114,50
desvio padrão	2,78	3,32	3,87
intervalo de confiança	[116, 48; 118, 62]	[115, 87; 118, 43]	[113, 01; 115, 99]
Retardo Total			
média [ms]	81,20	87,21	97,51
desvio padrão [ms]	21,56	25,71	33,68
intervalo de confiança [ms]	[80, 47; 81, 93]	[86, 34; 88, 08]	[96, 35; 98, 67]
Número de hops			
média	18,75	20,09	22,52
desvio padrão	4,78	5,75	7,66
intervalo de confiança	[18, 59; 18, 91]	[19, 89; 20, 29]	[22, 26; 22, 78]

Tabela 5.9: Desempenho do TLAR para várias margens de erro nas *conectividades*. Intervalo de chegada de veículos: 3s, *taxa de penetração*: 40%, alcance de transmissão: 200 m.

resultados dessa simulação estão na Figura 5.23 e na Tabela 5.9.

A Figura 5.23 mostra que o desempenho do TLAR é pouco afetado por erros de até dez pontos percentuais nas *conectividades*. Erros de vinte pontos percentuais proporcionam uma degradação maior no desempenho do algoritmo, mas ainda assim a média de pacotes entregues converge para o valor obtido para o caso em que a margem de erro era nula (cerca de 117 pacotes). O comprimento das rotas, por sua vez, aumenta em apenas 2,43 hops quando o erro passa de dez para vinte pontos percentuais e, mesmo nessas condições, o tamanho do intervalo de confiança dessa métrica é de apenas 0,52 hop, um pequeno aumento se comparado ao comprimento do mesmo intervalo de confiança para uma margem de erro nula nas *conectividades* (0,32).

Parte desse comportamento se deve ao fato de a *conectividade* de uma rota ser o produto das *conectividades* das ruas e cruzamentos que a compõem. Tendo em vista que o erro aplicado às *conectividades* tem distribuição uniforme e simétrica em relação ao zero, acréscimos e decréscimos aos valores reais das *conectividades* tendem a se atenuar mutuamente.

O outro motivo da robustez do TLAR à imprecisão das *conectividades* está em sua estratégia de reparo, que penaliza as ruas pelas quais um pacote já passou. Isso inibe dois veículos de encaminharem o mesmo pacote de um para o outro até que o tempo de vida desse pacote se esgote. Tal situação poderia ocorrer se a diferença entre as *conectividades* armazenadas na memória de um veículo produzisse uma rota cujo próximo hop fosse o outro, e vice-versa.

Algoritmo	Onisciente	U-GPSR	STAR	TLAR
Pacotes entregues				
média	120,35	119,00	112,90	117,55
desvio padrão	1,15	2,10	6,13	2,78
intervalo de confiança	[119, 91; 120, 79]	[118, 19; 119, 81]	[110, 54; 115, 26]	[116, 48; 118, 62]
Retardo total				
média [ms]	61,21	85,21	78,23	81,20
desvio padrão [ms]	7,06	48,05	16,30	21,56
intervalo de confiança [ms]	[60, 97; 61, 45]	[83, 59; 86, 83]	[77, 67; 78, 79]	[80, 47; 81, 93]
Número de hops				
média	14,58	19,67	17,91	18,75
desvio padrão	0,89	10,98	3,45	4,78
intervalo de confiança	[14, 55; 14, 61]	[19, 30; 20, 04]	[17, 79; 18, 03]	[18, 59; 18, 91]

Tabela 5.10: Desempenho dos algoritmos para *taxa de penetração* de 40%, alcance de transmissão de 200 m e intervalo de chegada de veículos de 3s. A velocidade máxima dos veículos pertence ao intervalo  $[15, 0; 20, 0]m/s$ , enquanto a aceleração e desaceleração se encontram, respectivamente, nos intervalos  $[0, 7; 0, 9]m/s^2$  e  $[4, 0; 5, 0]m/s^2$ .

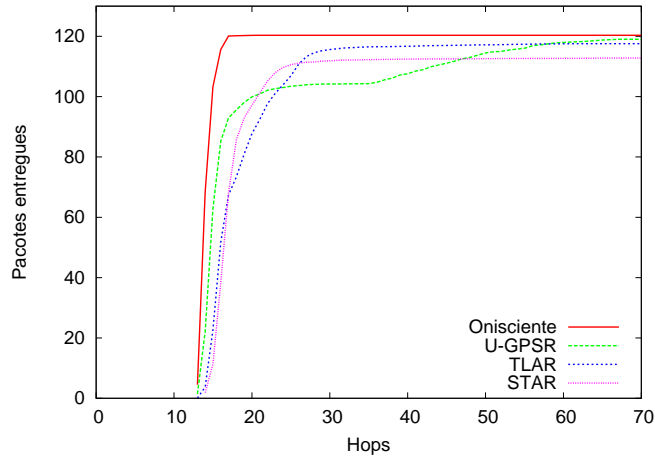
## 5.2.6

### Velocidade dos Veículos

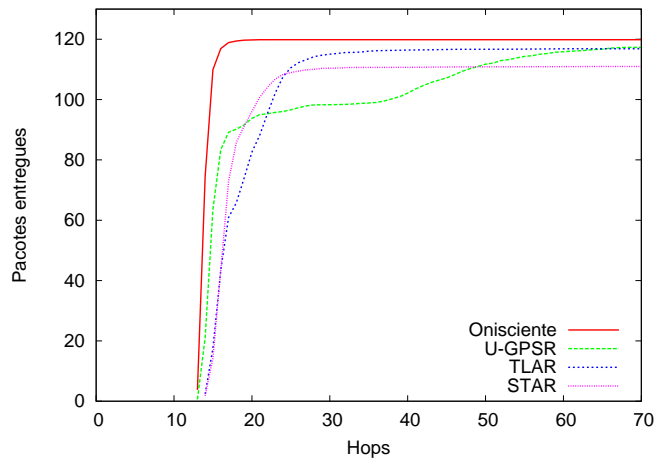
Os experimentos conduzidos nas seções anteriores envolveram veículos cujas velocidades máximas foram selecionadas aleatoriamente do intervalo  $[15, 20]m/s$ , e suas acelerações e desacelerações máximas pertenciam, respectivamente, aos intervalos  $[0, 7; 0, 9]m/s^2$  e  $[4, 5]m/s^2$ . Nesta seção, o objetivo do experimento é verificar qual a influência da velocidade dos veículos no desempenho dos algoritmos de roteamento. Para tanto, um segundo cenário foi simulado utilizando-se intervalos maiores para a velocidade, aceleração e desaceleração máximas, o que tende a produzir um tráfego veicular mais heterogêneo. Especificamente, esses parâmetros foram obtidos, respectivamente, dos intervalos  $[10, 20]m/s$ ,  $[0, 5; 1, 0]m/s^2$  e  $[3, 5; 5, 5]m/s^2$ . Os demais parâmetros de simulação foram mantidos constantes, a saber: taxa de penetração de 40%, intervalo de chegada de veículos de 3 s e alcance de comunicação de 200 m.

Foram simulados 4186 veículos (incluindo os *não-equipados*) em ambos os cenários, mas no primeiro a etapa transiente de simulação terminou no instante de tempo  $t = 554$ , ao passo em que, no segundo, isso se deu no instante  $t = 638$ . A partir de então, passaram a trafegar pela área simulada, em média, 1710 e 1772 veículos, respectivamente. Dentre eles, havia uma média de 689 veículos *equipados* no primeiro cenário e 706 no segundo.

A diferença entre esses valores se deve ao fato de que no segundo cenário a



(a) Velocidade máxima: 15–20  $m/s$   
 aceleração máxima: 0.7–0.9  $m/s^2$   
 desaceleração máxima: 4.0–5.0  $m/s^2$



(b) Velocidade máxima: 10–20  $m/s$   
 aceleração máxima: 0.5–1.0  $m/s^2$   
 desaceleração máxima: 3.5–5.5  $m/s^2$

Figura 5.24: Quantidade de pacotes entregues em função de seus tempos de vida para diferentes velocidades, acelerações e desacelerações. O intervalo de chegada de veículos é de 3 s, a taxa de penetração é de 40% e o alcance de comunicação é de 200 m.

Algoritmo	Onisciente	U-GPSR	STAR	TLAR
Pacotes entregues				
média	119,85	117,30	110,95	116,85
desvio padrão	1,46	2,74	5,38	3,64
intervalo de confiança	[119, 29; 120, 41]	[116, 24; 118, 36]	[108, 87; 113, 03]	[115, 45; 118, 25]
Retardo total				
média [ms]	60,69	89,96	77,25	81,88
desvio padrão [ms]	7,04	53,15	15,29	19,03
intervalo de confiança [ms]	[60, 45; 60, 93]	[88, 15; 91, 77]	[76, 72; 77, 78]	[81, 23; 82, 53]
Número de hops				
média	14,47	20,82	17,68	18,96
desvio padrão	0,83	12,21	3,10	4,29
intervalo de confiança	[14, 44; 14, 50]	[20, 41; 21, 23]	[17, 57; 17, 79]	[18, 81; 19, 11]

Tabela 5.11: Desempenho dos algoritmos para taxa de penetração de 40%, alcance de transmissão de 200 m e intervalo de chegada de veículos de 3s. A velocidade máxima dos veículos pertence ao intervalo  $[15, 0; 20, 0]m/s$ , enquanto a aceleração e desaceleração se encontram, respectivamente, nos intervalos  $[0, 5; 1, 0]m/s^2$  e  $[3, 5; 5, 5]m/s^2$ .

velocidade máxima dos veículos pertence a um intervalo cuja média é menor (15 m/s em vez de 17,5 m/s). Conseqüentemente, o primeiro veículo de cada fluxo demora mais para percorrer a área simulada, proporcionando uma etapa de tráfego transiente maior. O aumento da quantidade de veículos simultâneos no segundo cenário pode ser justificado pela lei de Little (GRO85), segundo a qual o número de entidades em uma fila é o produto da taxa de chegada pelo tempo médio de espera na fila. Tendo em vista que a taxa de chegada é constante (um veículo a cada 3 s por rua) e que o tempo dentro da área simulada aumenta (em virtude da diminuição da velocidade média dos veículos), o número de veículos na área simulada também aumenta.

Contudo, esse aumento em termos percentuais corresponde a apenas 3,63%, o que se deve principalmente ao fato de os veículos raramente conseguirem, em uma área urbana, desenvolver velocidade e aceleração plenas, tanto por causa dos semáforos quanto pela presença de outros veículos em suas proximidades. De fato, a velocidade média desenvolvida pelos veículos no primeiro cenário (velocidades máximas entre 15 m/s e 20 m/s) foi de 5,40 m/s, enquanto no outro cenário (velocidades máximas entre 10 m/s e 20 m/s) ela ficou em 4,92 m/s. Por esse motivo, os dois cenários simulados apresentaram resultados bastante parecidos (Figura 5.24 e Tabelas 5.10 e 5.11).

Algoritmo	Onisciente	U-GPSR	STAR	TLAR
Pacotes entregues				
média	120,35	119,00	112,90	117,55
desvio padrão	1,15	2,10	6,13	2,78
intervalo de confiança	[119, 91; 120, 79]	[118, 19; 119, 81]	[110, 54; 115, 26]	[116, 48; 118, 62]
Retardo total				
média [ms]	61,21	85,21	78,23	81,20
desvio padrão [ms]	7,06	48,05	16,30	21,56
intervalo de confiança [ms]	[60, 97; 61, 45]	[83, 59; 86, 83]	[77, 67; 78, 79]	[80, 47; 81, 93]
Número de hops				
média	14,58	19,67	17,91	18,75
desvio padrão	0,89	10,98	3,45	4,78
intervalo de confiança	[14, 55; 14, 61]	[19, 30; 20, 04]	[17, 79; 18, 03]	[18, 59; 18, 91]

Tabela 5.12: Desempenho dos algoritmos para semáforos sincronizados, a uma taxa de penetração de 40%, alcance de transmissão de 200 m e intervalo de chegada de veículos de 3s.

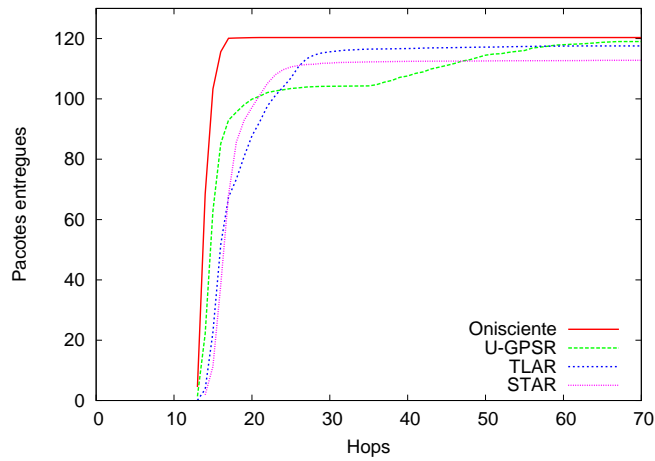
### 5.2.7

#### Assincronismo dos Semáforos

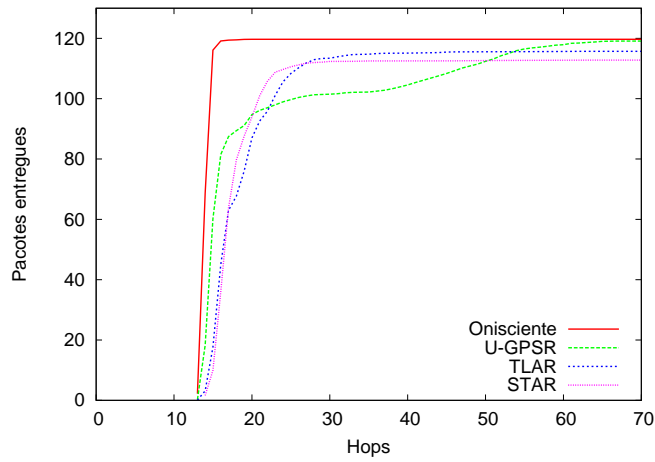
Nos experimentos anteriores, todos os semáforos funcionaram de forma sincronizada. Como consequência, os veículos tiveram a tendência de se distribuírem pelo cenário de simulação de forma regular, menos caótica do que ocorre em cenários reais. Tal cenário tende a ser desfavorável ao TLAR, uma vez que a principal característica deste algoritmo é justamente explorar flutuações momentâneas da concentração de veículos devido à mudança das fases dos semáforos. O objetivo deste experimento é, portanto, averiguar se o TLAR consegue aproveitar essa informação para criar rotas mais eficientes.

A melhor maneira de se verificar essa hipótese seria a simulação de cenários equivalentes a áreas urbanas reais, onde cada semáforo pode funcionar com um período diferente do dos demais e as defasagens são programadas de acordo com critérios de engenharia de tráfego. Devido a limitações de tempo, contudo, não foi possível conduzir o experimento dessa forma. Em vez disso o período dos semáforos foi mantido constante (120 s), mas o ciclo de funcionamento de cada um foi adiantado de um intervalo de tempo aleatório.

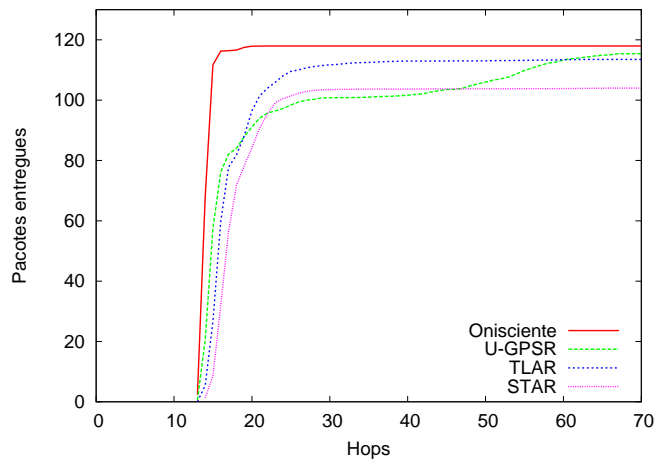
Além do cenário com semáforos sincronizados, dois outros também foram simulados: em um deles cada semáforo é adiantado aleatoriamente em até 30 s e no outro esse valor pode chegar a até 60 s. Cada cenário, por sua vez, é composto de cinco sub-cenários com defasagens independentes. Cada sub-cenário é simulado com quatro repetições, totalizando, dessa forma 20 repetições com defasagens de até 30 s e outras 20 com defasagens de até 60 s.



(a) Semáforos sincronizados



(b) Semáforos defasados de no máximo 30 s



(c) Semáforos defasados de no máximo 60 s

Figura 5.25: Quantidade de pacotes entregues em função de seus tempos de vida para semáforos sincronizados (a) e dessincronizados (b e c). A defasagem dos semáforos é de no máximo 30 s (b) e 60 s (c). O intervalo de chegada de veículos é de 3 s, a taxa de penetração é de 40% e o alcance de comunicação é de 200 m.

Algoritmo	Onisciente	U-GPSR	STAR	TLAR
Pacotes entregues				
média	119,70	119,10	112,80	115,70
desvio padrão	2,08	2,28	5,40	6,33
intervalo de confiança	[118, 90; 120, 50]	[118, 22; 119, 98]	[110, 72; 114, 88]	[113, 26; 118, 14]
Retardo total				
média [ms]	60,76	90,01	78,70	81,38
desvio padrão [ms]	7,72	52,18	14,92	19,64
intervalo de confiança [ms]	[60, 50; 61, 02]	[88, 25; 91, 77]	[78, 18; 79, 28]	[80, 71; 82, 05]
Número de hops				
média	14,45	20,76	18,05	18,80
desvio padrão	0,63	10,93	3,21	4,36
intervalo de confiança	[14, 43; 14, 47]	[20, 36; 21, 16]	[17, 94; 18, 16]	[18, 65; 18, 95]

Tabela 5.13: Desempenho dos algoritmos para semáforos dessincronizados, a uma taxa de penetração de 40%, alcance de transmissão de 200 m e intervalo de chegada de veículos de 3s. A defasagem máxima é de 30 s.

Algoritmo	Onisciente	U-GPSR	STAR	TLAR
Pacotes entregues				
média	117,95	115,40	104,00	113,50
desvio padrão	4,75	6,47	9,59	7,29
intervalo de confiança	[116, 12; 119, 78]	[112, 90; 117, 90]	[103, 30; 107, 70]	[110, 69; 116, 31]
Retardo total				
média [ms]	60,87	90,34	79,47	76,87
desvio padrão [ms]	6,96	55,56	16,01	19,56
intervalo de confiança [ms]	[60, 63; 61, 11]	[88, 44; 92, 24]	[78, 89; 80, 05]	[76, 19; 77, 55]
Número de hops				
média	14,50	20,91	18,25	17,77
desvio padrão	0,82	12,72	3,55	4,31
intervalo de confiança	[14, 47; 14, 53]	[20, 47; 21, 35]	[18, 12; 18, 38]	[17, 62; 17, 92]

Tabela 5.14: Desempenho dos algoritmos para semáforos dessincronizados, a uma taxa de penetração de 40%, alcance de transmissão de 200 m e intervalo de chegada de veículos de 3s. A defasagem máxima é de 60 s.



Os demais parâmetros de simulação foram mantidos constantes como se segue: o intervalo de chegada de veículos foi de 3 s, a taxa de penetração foi de 40%, o alcance de comunicação foi de 200 m e a velocidade máxima dos veículos foi selecionada do intervalo [15, 20] m/s. O tráfego de veículos superou a etapa transiente no instante  $t = 554$  para o cenário com semáforos sincronizados. Para os demais cenários, na média dos cinco sub-cenários, os instantes de tempo foram  $t = 561$  para defasagens de até 30 s e 565 para defasagens de até 60 s. As médias da quantidade de veículos em cada cenário, a cada instante de tempo após a fase transiente, foram 1710, 1699 e 1746, dos quais 689, 683 e 702 foram *equipados*.

Os resultados das simulações, resumidos na Figura 5.25 e nas Tabelas 5.12, 5.13 e 5.14, indicam que uma defasagem aleatória de até 30 s causa pouco efeito sobre o desempenho dos algoritmos, se comparados ao cenário com semáforos sincronizados. Para uma defasagem de até 60 s, STAR e U-GPSR experimentaram as maiores reduções em suas médias de pacotes entregues (8,80 e 3,70, respectivamente). O comprimento das rotas em *hops* de todos os algoritmos simulados sofreu alterações desprezíveis, salvo pela redução de um *hop* em ambos os limites do intervalo de confiança do tamanho das rotas do TLAR.

### 5.3

#### Análise e Discussão

Os experimentos discutidos na Seção 5.2 avaliaram o desempenho dos algoritmos sob diversas condições de volume de tráfego, taxa de penetração e outros parâmetros de simulação. Os resultados obtidos permitem determinar as condições em que os algoritmos TLAR e U-GPSR apresentam melhor desempenho, bem como suas limitações.

Quanto à taxa de penetração, os experimentos mostram que o TLAR é o algoritmo de melhor desempenho para a taxa de 30%, ao passo em que o U-GPSR se torna a melhor opção quando a taxa é de 50%. Nesse último caso, a alta disponibilidade de *veículos equipados* favorece a heurística construtiva gulosa do U-GPSR, mas o mecanismo de monitoramento proativo do STAR proporcionou a este algoritmo uma ligeira vantagem em relação ao U-GPSR.

Porém, conforme explicado na Seção 5.1.3, a implementação simulada nos experimentos utiliza uma tabela de tráfego global compartilhada por todos os veículos. Acreditamos que a simulação do mecanismo de disseminação original do STAR degradaria o desempenho deste algoritmo de roteamento sensivelmente, uma vez que impõe um retardo entre a ocorrência de uma anomalia de tráfego (excesso ou escassez de *veículos equipados* em alguma

rua) e sua disseminação. Além disso, o alcance de disseminação é limitado e esse procedimento consome recursos da rede, principalmente largura de banda.

Uma análise análoga à do efeito da taxa de penetração pode ser conduzida para o experimento em que se varia a taxa de chegada dos veículos, uma vez que ambos os experimentos proporcionam uma quantidade de *veículos equipados* diferentes em cada cenário. Quando essa taxa é de 4s, o TLAR tem um desempenho claramente melhor que o dos demais algoritmos (exceto pelo onisciente), ao passo em que o U-GPSR possui maior taxa de entrega para rotas mais curtas quando o intervalo de chegada é de 2 s. Esse resultado confirma que uma densidade de *veículos equipados* elevada permite que o roteamento seja conduzido por abordagens mais simples como a do U-GPSR. Nesse caso, o custo computacional de alternativas mais sofisticadas como o TLAR e o STAR podem não justificar uma eventual melhoria de desempenho. Um exemplo disso é que ocorre no gráfico da Figura 5.22(a), onde o desempenho do STAR supera o do U-GPSR para rotas curtas a um intervalo de chegada de 2 s. Mesmo assim, nota-se que sua estratégia de reparo não faz com que sua média de pacotes entregues convirja ao patamar alcançado pelos outros algoritmos.

Os demais experimentos mostram ainda que o desempenho dos algoritmos de roteamento praticamente não é afetado por padrões de tráfego caracterizados por uma diversidade maior na velocidade dos veículos. O mesmo já não pode ser afirmado para o caso em que os semáforos não estão sincronizados. Apesar de uma defasagem de até 30 s proporcionar poucas alterações no desempenho dos algoritmos em relação ao cenário sincronizado, nota-se que o TLAR apresenta resultados claramente melhores quando a defasagem é de até 60 s. Nesse caso, o assincronismo dos semáforos produz uma distribuição de tráfego veicular mais heterogênea, para a qual algoritmos mais simples, a exemplo do U-GPSR, não foram projetados. Mesmo a abordagem proativa do STAR não atingiu bons resultados, o que talvez possa ser melhorado com uma calibragem de seus parâmetros (discutida na Seção 5.2.1) específica para este cenário.

Um dos principais limites do TLAR se refere à densidade de *veículos equipados*. De acordo com os resultados obtidos nas Seções 5.2.3 e 5.2.4, o desempenho do TLAR se degrada à medida que a densidade de *veículos equipados* diminui. Porém, mesmo a um intervalo de chegada de veículos de 4s ou a uma taxa de penetração de 30%, tanto o TLAR quanto o U-GPSR conseguem entregar com sucesso uma média de cerca de 90% do total de pacotes enviados, com um retardo da ordem de 100 ms. Nessas condições, o desempenho dos algoritmos atende aos requisitos de aplicações que não sejam de tempo real (a exemplo de navegação pela Internet). Para intervalos de chegada mai-

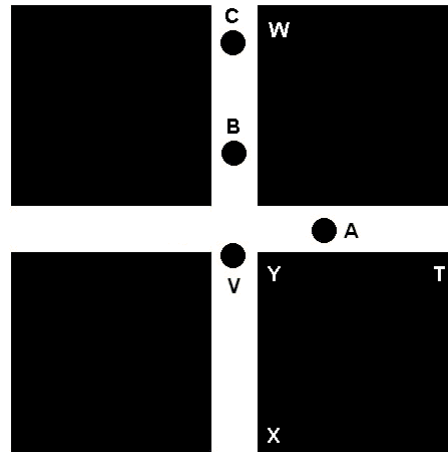


Figura 5.26: Um cenário para a demonstração de uma possível melhoria para a estratégia de reparo do TLAR.

ores ou taxas de penetração menores, as VANETs tendem a se particionarem, tornando necessário o uso da técnica de armazenamento temporário de pacotes (*store-and-forward*). Contudo, a versão atual dos algoritmos simulados não implementa essa funcionalidade, o que deverá ser abordado em trabalhos futuros.

Outra limitação do TLAR diz respeito ao congestionamento no meio sem fio. Conforme explicado na Seção 4.2.3, a heurística construtiva do TLAR seleciona rotas de maior *conectividade*. Porém, a versão atual do TLAR não prevê nenhum mecanismo de controle de congestionamento no meio sem fio. Assim, se vários veículos situados nas proximidades de um mesmo ponto *A* enviarem pacotes destinados a veículos localizados nos arredores de um ponto *B*, a heurística construtiva do TLAR tenderá a produzir as mesmas rotas para todos esses pacotes, o que pode gerar congestionamento no meio sem fio ao longo dessa rota. Esse problema deverá ser investigado em trabalhos futuros.

Adicionalmente, a estratégia de reparo do TLAR confere a esse algoritmo uma boa tolerância ao erro no cálculo das *conectividades* (de 10 a até 20 pontos percentuais). Contudo, o número de pacotes entregues com sucesso não converge ao patamar atingido pelo algoritmo onisciente quando a VANET é mais esparsa, a exemplo dos experimentos onde a taxa de penetração foi de apenas 30% ou em que o intervalo de chegada de veículos foi de 4 s. Mesmo nos experimentos em que o TLAR consegue entregar aproximadamente o mesmo número de pacotes que o algoritmo onisciente, melhorias na estratégia de reparo poderiam reduzir o tamanho médio das rotas criadas pelo TLAR.

Uma possibilidade seria incluir no cabeçalho dos pacotes um campo em que o último veículo que o roteou escrevesse a *conectividade* da segunda

melhor rota disponível para o destinatário (chamado de  $CA$  — *conectividade alternativa*). Dessa forma, se um veículo  $A$  encaminhasse um pacote a um veículo  $B$ , este teria uma estimativa melhor de qual seria a probabilidade de entrega do pacote se ele fosse encaminhado de volta a  $A$ .

Esse raciocínio pode ser exemplificado na Figura 5.26. Seja  $V$  um veículo que envia um pacote destinado a um ponto  $Z$ , localizado várias quadras à direita de  $V$  (que não é mostrado na figura). Seja  $A$  o próximo *hop* da rota a  $Z$ . Ao receber o pacote,  $A$  é obrigado a reenviá-lo a  $V$ , já que não possui nenhum vizinho em  $YT$  e  $V$  é seu único vizinho. Por esse motivo, todas as rotas de  $A$  para  $Z$  passam necessariamente por  $V$ , logo a *conectividade alternativa* ( $CA$ ) é ajustada para zero.

Ao receber o pacote de volta,  $V$  determina que é impossível enviar o pacote a  $Z$  através do *segmento*  $YT$  (pois  $CA = 0$ ), e o encaminha a  $B$ . Suponhamos este veículo determine que as duas melhores rotas de  $B$  para  $Z$  sejam: uma com *conectividade*  $c_1$  que passe novamente por  $V$  e outra, cuja *conectividade* é  $c_2$ , que passe por  $C$ . Seja  $c_1 > c_2$ , de modo que  $B$  envia o pacote de volta a  $V$  (por pertencer à rota de maior *conectividade*) com  $CA = c_2$  (a *conectividade* da segunda melhor rota para  $Z$ ).

A seguir,  $V$  precisa determinar o próximo *hop* na direção de  $Z$ , sendo que o pacote já voltou uma vez de  $YT$  e uma vez de  $YW$ , proporcionando a aplicação da mesma penalidade  $q$  a cada um deles. Tendo em vista que a rota de maior *conectividade* de  $V$  a  $Z$  inicialmente passava pelo *segmento*  $YT$ , a aplicação de penalidades iguais a  $YT$  e  $YW$  faria com que  $A$  fosse selecionado novamente como próximo *hop* pela estratégia de reparo original, conduzindo o pacote novamente a um *ótimo local*. O uso do campo de conectividade alternativa impede que isso aconteça, fazendo o pacote ser reenviado a  $B$  com  $CA = 0$ , uma vez que não existiriam outras rotas de  $V$  a  $Z$  que não passassem por  $B$ .