

## 4 Propostas de Algoritmos de Roteamento

Uma vez discutidas as principais características das redes veiculares, este capítulo apresenta na Seção 4.1 a modelagem de VANETs utilizada no restante desta tese, enquanto apresentamos a proposta de dois algoritmos de roteamento para o modelo de VANETs assumido nas Seções 4.2 e 4.3.

### 4.1 Modelo de VANETs

O desempenho de protocolos e aplicações para redes veiculares depende fortemente de vários fatores, a exemplo da comunicação sem fio e do tráfego veicular, cujas modelagens são discutidas a seguir.

#### 4.1.1 Propagação de Sinal

A comunicação inter-veicular requer interfaces de comunicação sem fio e antenas. Os algoritmos propostos nesta tese não requerem uma tecnologia de comunicação específica, mas assumem o uso de antenas omnidirecionais e enlaces simétricos, ou seja, se um veículo  $A$  for capaz de se comunicar diretamente (i.e. sem roteamento) com outro veículo  $B$ , então  $B$  será capaz de se comunicar com  $A$ . Essa hipótese é adotada por motivos de simplicidade nesta tese, assim como em vários outros trabalhos na literatura (SEE04, GIU05, KAR00, ZHA08).

Conforme discutido na Seção 2.1, a propagação em radiofrequência em cenários urbanos é bastante complexa em virtude de fenômenos como reflexão e difração. Como uma modelagem mais minuciosa da propagação de sinal está fora do escopo desta tese, um modelo mais simplificado foi adotado, discutido a seguir.

#### Modelo de Propagação

Conforme discutido na Seção 2.1, é possível observar que mesmo os modelos não-determinísticos (a exemplo do *shadowing*) prevêm uma baixa degradação da qualidade do sinal quando transmissor e receptor estão separados

por uma distância curta (por exemplo, até 50 m). Nesse caso, a taxa de sucesso da transmissão pelo meio sem fio tende a ser alta, como mostram os experimentos de medição citados na Seção 2.1.

Esses experimentos permitem estimar a distância máxima entre transmissor e receptor para que a probabilidade de sucesso da comunicação esteja acima de um patamar pré-estabelecido. Para fins de modelagem de propagação, esta tese denomina esses parâmetros, respectivamente, como *alcance efetivo de transmissão* (AET) e *nível de confiança de transmissão* (NCT).

Dessa forma, o modelo de comunicação utilizado assume que uma das condições necessárias para que dois veículos possam se comunicar é a de que a distância que os separa deve ser inferior ao AET adotado. Trata-se portanto de uma modelagem simplificada, tendo em vista que a comunicação a distâncias maiores é possível no mundo real, e que, por outro lado, mesmo a distâncias curtas o sucesso da comunicação não é garantido.

A outra condição necessária para o sucesso da comunicação inter-veicular se refere à existência de linha de visão entre transmissor e receptor. Conforme discutido na Seção 2.1, a presença de obstáculos afeta o desempenho da comunicação sem fio. Contudo, os resultados apresentados naquela seção indicam que seria pouco realista assumir que os obstáculos impeçam *completamente* a propagação do sinal. Por esse motivo, o modelo adotado também define o *alcance efetivo de transmissão obstruída* (AETO) como a distância máxima entre dois veículos tal que a comunicação ocorra com uma probabilidade de sucesso maior ou igual ao *nível de confiança de transmissão* (NCT), apesar da não existência de uma linha de visão desobstruída entre eles.

Essa distância é função de diversos fatores, dentre os quais podem ser citados o tamanho e a posição dos obstáculos, bem como o material de que são constituídos. Outros fatores importantes também incluem a frequência do sinal e o ângulo de incidência, como discutido na Seção 2.1. O modelo adotado nesta tese simplifica esses fatores assumindo que os obstáculos são menores do que no mundo real e restringindo a comunicação exclusivamente ao caso em que exista uma linha de visão desobstruída entre transmissor e o receptor no cenário resultante.

A Figura 4.1 ilustra a modelagem adotada, onde  $A$  e  $B$  são dois veículos cuja linha de visão é obstruída por um conjunto de prédios, representados pelos retângulos pretos. Na Figura 4.1(a), o sinal emitido por  $A$  é recebido por  $B$  apesar da obstrução pelos prédios. Supondo que  $\overline{AB}$  seja o *alcance efetivo de transmissão obstruída*, o mesmo resultado seria obtido se a transmissão tivesse ocorrido no cenário da Figura 4.1(b), assumindo-se que a propagação só seja possível na existência de linha de visão.

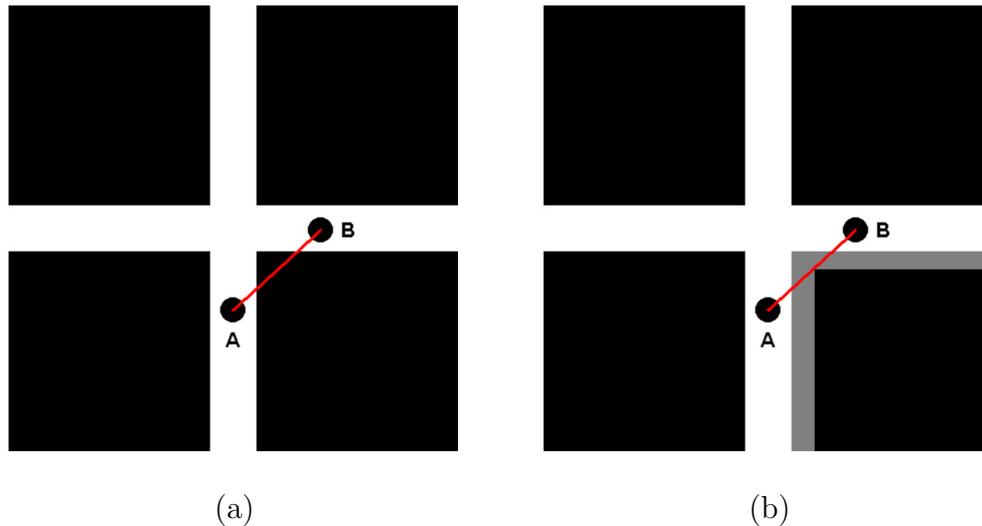


Figura 4.1: Redução do tamanho dos obstáculos para fins de propagação.

A Figura 4.1(b), por sua vez, foi obtida pela redução do tamanho do bloco interceptando a transmissão de  $A$  para  $B$ , indicada em cinza. Essa área também pode ser interpretada como uma calçada virtual, determinada exclusivamente pela capacidade de transpor obstáculos de certa espessura. Em outras palavras, o problema de determinar se dois veículos são capazes de se comunicar diretamente na ausência de linha de visão pode ser aproximado pelo problema equivalente contendo as calçadas virtuais e onde a propagação é completamente bloqueada quando incide em blocos subtraídos dessas calçadas virtuais (área preta na Figura 4.1).

### Modelo da Camada MAC

O modelo adotado para a camada MAC (*Medium Access Control* — Controle de Acesso ao Meio) assume que todas as transmissões sejam feitas em um único canal de frequência, possibilitando que haja interferência entre transmissões simultâneas. O sucesso ou fracasso de uma transmissão é determinado de acordo com a potência com que ela atinge o receptor. Para essa finalidade, dois limiares são definidos: a *sensibilidade* (CS — *carrier sense*) e o *limiar de recepção* (RxT — *receiving threshold*).

A *sensibilidade* é a intensidade mínima de sinal que uma interface de comunicação identifica como uma transmissão. Nenhuma transmissão recebida com uma potência mais tênue do que esse limiar pode ser decodificada, e é tratada como ruído. Por outro lado, se a intensidade do sinal recebido estiver acima do *limiar de recepção* e não houver transmissões simultâneas, considera-se que a transmissão possa ser decodificada corretamente.

Finalmente, se a intensidade do sinal estiver entre esses limiares, ele

não será forte o bastante para permitir sua decodificação, mas poderá causar uma colisão no meio sem fio caso haja alguma outra transmissão em curso. Nesse caso, nenhuma das transmissões simultâneas é decodificada com sucesso. Contudo, se a transmissão em curso tiver intensidade consideravelmente maior (da ordem de 10 dB) (DRI04) do que a que está causando a interferência, a primeira será decodificada com sucesso apesar da interferência causada pela segunda.

#### 4.1.2

##### Modelo de Tráfego

Conforme mencionado no Capítulo 2, os modelos de tráfego veicular podem ser divididos em microscópicos e macroscópicos. Os algoritmos propostos nesta tese, bem como a maioria dos trabalhos relacionados, assumem que cada veículo (nó da VANET eventualmente participante do roteamento) tenha acesso à informação sobre sua própria posição em tempo real, o que exige uma modelagem de tráfego em escala microscópica.

O principal objetivo da modelagem microscópica é determinar a velocidade com que cada veículo se desloca ao longo do tempo (a posição do veículo pode ser calculada por integração). Uma abordagem comum consiste em determinar a velocidade de cada veículo  $v_i$  em função da velocidade desenvolvida pelo veículo à sua frente  $v_{i+1}$ , como proposto em (LOU53) (Equação 4-1).

$$\frac{dv_i(t)}{dt} = \frac{v_{i+1}(t) - v_i(t)}{\tau}, \quad (4-1)$$

onde  $v_i$  é a velocidade do veículo  $i$  e  $\tau$  é um parâmetro de intervalo de tempo.

Um modelo um pouco mais realista (CHA58) acrescenta um tempo de reação finito  $\Delta t$ , ou seja, a velocidade atual de cada veículo seria função da velocidade desenvolvida pelo veículo à frente num instante de tempo anterior,  $t - \Delta t$  (Equação 4-2).

$$\frac{dv_i(t)}{dt} = \frac{v_{i+1}(t - \Delta t) - v_i(t - \Delta t)}{\tau} \quad (4-2)$$

Porém, uma vez que pela Equação 4-2 a velocidade de cada veículo não depende da distância que o separa do veículo imediatamente à frente, esse modelo não garante que colisões entre veículos sejam evitadas. Outra limitação do modelo é a de não ser possível derivar, a partir da Equação 4-2, uma relação entre a velocidade de um veículo e a densidade de tráfego, ou seja que veículos trafeguem mais lentamente em áreas de maior densidade veicular.

Uma solução para esses problemas foi proposta em (GAZ59), onde a velocidade dos veículos segue a Equação 4-3.

$$\frac{dv_i(t)}{dt} = \alpha \frac{v_{i+1}(t - \Delta t) - v_i(t - \Delta t)}{x_{i+1}(t - \Delta t) - x_i(t - \Delta t)}, \quad (4-3)$$

onde  $\alpha$  é uma constante e  $x_i$  é a posição linear do veículo  $i$ , ou seja,  $x_{i+1}(t) - x_i(t)$  é a distância de um veículo àquele imediatamente à sua frente no instante de tempo  $t$ .

A Equação 4-3 foi generalizada em (GAZ61), resultando na Equação 4-4.

$$\frac{dv_i(t)}{dt} = \alpha v_i(t)^m \frac{v_{i+1}(t - \Delta t) - v_i(t - \Delta t)}{(x_{i+1}(t - \Delta t) - x_i(t - \Delta t))^l}, \quad (4-4)$$

onde  $l$  e  $m$  são parâmetros livres. Contudo, a determinação desses valores a partir de dados experimentais é uma tarefa difícil (GAZ61).

Ao contrário dos modelos apresentados até este ponto, outras abordagens determinam a velocidade de cada veículo de forma iterativa, em passos discretos (YUK95, LUB98, GIP81, CRE86, NAG94, NAG92, KAI94, SCH93, SCH95). Um dos principais trabalhos nessa linha é o modelo de Nagel-Schreckenberg (NAG94, NAG92, KAI94, SCH93, SCH95), em que as ruas são divididas em células de tamanho constante, que podem ou estar vazias ou ocupadas por um veículo. A velocidade de cada veículo corresponde ao número de células que ele consegue percorrer por unidade de tempo. Porém, trata-se de um modelo estocástico que acrescenta uma componente de ruído (erro) aleatório no cálculo das velocidades dos veículos, de modo a simular imperfeições dos motoristas ao dirigirem. Este modelo reproduz bem as relações empíricas de velocidade e densidade de veículos (NAG94, NAG92, KAI94, SCH93, SCH95).

Nesta tese, o modelo adotado (KRA98) segue uma abordagem similar, porém é discreto no tempo, mas contínuo no espaço. Apesar de aplicar a cada veículo valores de aceleração e desaceleração comparáveis aos exibidos por veículos reais, ele não consegue reproduzir situações de congestionamento com precisão (KRA98). Mesmo assim, ele é adequado aos experimentos conduzidos nesta tese, que, em sua maioria, não adotam volumes de tráfego muito altos. Além disso, tendo em vista que o foco da tese são algoritmos de roteamento para VANETs, erros da ordem de alguns metros no posicionamento dos veículos são aceitáveis.

## 4.2

### TLAR

Nesta seção é apresentado o TLAR (*Traffic Light Aided Routing* — roteamento auxiliado por semáforos), um algoritmo para comunicação inter-veicular destinado a áreas urbanas. Primeiramente, as hipóteses em que o TLAR se baseia são discutidas, e, a seguir, passa-se à apresentação da heurística construtiva e da estratégia de reparo do algoritmo.

### 4.2.1

#### Hipóteses

Assim como muitos dos algoritmos de roteamento para VANETs, o TLAR emprega alguns recursos disponíveis em *veículos equipados*, a saber, um sistema de posicionamento (como o GPS (KAP05)) e mapas digitais (TIG08). As abordagens que utilizam tais tecnologias obtêm desempenho substancialmente melhor do que os protocolos de roteamento para MANETs sem informação de localização (LOC03), como os bem conhecidos AODV (PER99), DSR (JOH01), etc.

O algoritmo TLAR assume que cada veículo tenha à sua disposição a informação aproximada sobre sua localização, pois a grande maioria dos veículos equipados com uma interface de comunicação sem fio também dispõe de sistemas de navegação GPS. Apesar da margem de erro existente em sistemas como o GPS em ambientes urbanos (conforme discutido no Capítulo 2), muitos trabalhos (inclusive o TLAR) (JER06, SEE04, ZHA08, GIU05) presumem que cada veículo tenha a seu dispor a informação precisa sobre a sua própria localização.

Outra hipótese, adotada em nosso trabalho e por outros protocolos do tipo *geocast*, é a disponibilidade de um *serviço de localização* (GRO03, LI00, KIE04), cujo objetivo é inferir a posição de um veículo, dado o seu identificador. Tais serviços normalmente empregam uma combinação de protocolos do tipo inundação controlada e *caching*, e não fazem parte do escopo dessa tese. A versão atual dos algoritmos propostos nesta tese assume que a posição do destinatário de um pacote pode ser obtida com erro e retardo desprezíveis.

Um aspecto relevante, apesar de pouco discutido na literatura, é o papel da *taxa de penetração* da tecnologia de comunicação inter-veicular, definida como o percentual de veículos equipados com uma interface de comunicação sem fio e que executam o algoritmo TLAR (aos quais esta tese se refere como *veículos equipados* a partir deste ponto). Este é um dos fatores determinantes da densidade de *veículos equipados* em uma VANET e, conseqüentemente, do desempenho dos protocolos de roteamento, conforme discutido no Capítulo 5. O algoritmo TLAR assume que cada veículo possua a informação sobre esse valor e que seja necessário muito tempo (da ordem de meses ou anos) para que o valor da taxa de penetração se altere de forma significativa permitindo que seja atualizado durante inspeções veiculares esporádicas (como reparos em redes autorizadas e vistoria por órgãos governamentais), ou ainda obtido diretamente de outros veículos, previamente atualizados, por difusões regulares de mensagens de atualização.

Outro parâmetro importante para o roteamento em VANETs é a densi-

dade de tráfego nas ruas da cidade, o que ajuda a evitar o envio de pacotes através de vias menos movimentadas. A densidade de tráfego varia de acordo com data, hora e local, estando sujeita, ainda, a variações decorrentes de acidentes, de condições climáticas adversas (tais como chuva e neve), da realização de eventos públicos (a exemplo de partidas de futebol e espetáculos), etc. Abordagens proativas, tais como o STAR (GIU05), estimam e disseminam valores de densidade de tráfego em tempo real. Isso pode ser feito, por exemplo, pela contagem de vizinhos que cada veículo possui em determinada rua ou pela velocidade média desenvolvida pelos veículos, que tende a ser menor em ruas congestionadas. Desse forma, o algoritmo de roteamento pode se adaptar a mudanças inesperadas das condições de tráfego.

Por outro lado, algoritmos como o VADD (ZHA08) e o TLAR constroem suas rotas em função de estatísticas de tráfego obtidas de bases de dados como (MAP09) e (SMA09). Essas bases de dados são alimentadas por informações que podem ser obtidas ao longo do dia por meio de câmeras (SAN00, TAK94), sensores instalados na pista (SIN05), ou até mesmo catalogadas pelos próprios veículos à medida em que trafegam no dia-a-dia (CAT01).

Seja qual for o método de obtenção, as informações de tráfego precisam ser transferidas a uma base de dados central, de modo que ela reflita eventuais alterações decorrentes, por exemplo, do aumento da frota de veículos de uma cidade. A instância da base de dados armazenada na memória dos veículos, por sua vez, também precisa ser atualizada pelo mesmo motivo. Ambas as operações podem ser efetuadas durante procedimentos de vistoria, abastecimento ou manutenção veiculares.

As informações contidas em uma base de dados como esta precisa englobar as várias combinações possíveis de parâmetros (dia da semana, hora, local), o que pode requerer uma quantidade considerável de espaço de armazenamento. Contudo, em vista do atual barateamento desse recurso, tal exigência passou a ter menor importância.

Por fim, uma outra hipótese adotada pelo TLAR é a de que a mudança dos estados dos semáforos, de aberto para fechado e vice-versa, ocorrem com período fixo, e em instantes de tempo conhecidos. Essa informação é essencial para a construção de rotas no TLAR, uma vez que permite prever flutuações do fluxo de veículos (o que é discutido no Capítulo 5).

No entanto, o ciclo “aberto-fechado” de semáforos reais se modifica ao longo do dia, e isso demandaria que no TLAR cada veículo se mantivesse atualizado em relação aos instantes de tempo em que cada semáforo muda de estado, o que atualmente não ocorre. No entanto, cada veículo poderia, em princípio, detectar se os semáforos estão fechados ou abertos à medida em que,

respectivamente, param próximos aos cruzamentos ou trafegam por eles. A disseminação da informação do estado (aberto ou fechado) de cada semáforo ao longo do tempo permitiria estimar os instantes futuros de suas transições aberto-fechado e vice-versa, uma vez que o período desse ciclo não se altera num horizonte de vários minutos. Contudo, a investigação desse procedimento se encontra além do escopo desta tese e é objeto de trabalhos futuros.

#### 4.2.2 Definições

Conforme explicado na seção anterior, prédios e outros obstáculos atenuam sinais de rádio, podendo bloqueá-los completamente. Define-se, assim, que dois veículos são mutuamente *visíveis* se, e somente se, levando em conta calçadas virtuais (discutido na Seção 4.1.1), houver uma linha de visão desobstruída entre eles. Se, além disso, a distância que os separa for inferior ao alcance de transmissão, diz-se que eles são mutuamente *alcançáveis*.

Adicionalmente, um veículo  $A$  é dito *comunicável diretamente* com  $B$  se, e somente se, ambos forem *equipados*, e mutuamente *alcançáveis*. Por motivos de simplicidade, será assumido que o alcance do sinal de rádio de todos os nós é o mesmo, implicando assim que toda comunicação é bidirecional. Define-se, ainda, que dois veículos  $A$  e  $B$  são *comunicáveis indiretamente* entre si se forem *comunicáveis diretamente* um com o outro, ou se existir um terceiro veículo que seja *comunicável indiretamente* com ambos.

Nas seções seguintes, os conceitos fundamentais do TLAR, a *conectividade cruzada* (CC) e *conectividade reta* (CR), são apresentados.

#### Conectividade Cruzada

Conforme explicado na Seção 2.1, obstáculos como prédios e árvores afetam a propagação de sinal de forma significativa, ou seja, a transmissão entre veículos exhibe desempenho pior na ausência de linha de visão. Tendo em vista que essa situação ocorre frequentemente nos cruzamentos de áreas urbanas, esta tese define o conceito de *conectividade cruzada* (CC) como sendo a probabilidade de existirem pelo menos dois veículos *comunicáveis diretamente* entre si localizados em ruas diferentes e concorrentes em um cruzamento.

A *conectividade cruzada* depende de vários fatores, tais como a densidade de veículos, o percentual dos que são *equipados*, a largura das ruas e calçadas e até mesmo a existência de outras transmissões simultâneas no cruzamento em questão. O cálculo do valor *exato* da *conectividade cruzada* é, portanto,

inviável. Contudo, um valor aproximado pode ser calculado conforme explicado na Seção 5.2.2, para fins de avaliação do TLAR.

### Conectividade Reta

Conforme explicado na Seção 4.2, uma rota no TLAR consiste em uma seqüência ordenada de cruzamentos pelos quais o pacote deve ser enviado a fim de chegar a seu destino. Dois cruzamentos são ditos *consecutivos* se, e somente se, estiverem conectados entre si por um *segmento*. Um *segmento*, por sua vez, consiste em um trecho de rua que possua cruzamentos *exclusivamente* em suas extremidades.

Assim, define-se como a *conectividade reta* (CR) de um segmento  $AB$  a probabilidade de dois veículos *equipados*, localizados em suas extremidades, estarem *comunicáveis indiretamente* entre si *exclusivamente* através de veículos situados em  $AB$ . Analogamente ao que ocorre com a *conectividade cruzada*, a *conectividade reta* também é afetada pelo número de veículos *equipados*, por suas posições ao longo das ruas e pelo alcance de comunicação sem fio. O algoritmo para se calcular o valor aproximado da *conectividade reta* é apresentado na Seção 5.2.2.

Combinadas, as *conectividades reta* e *cruzada* permitem estimar a probabilidade de sucesso da entrega de um pacote em função da rota escolhida. Uma rota, por sua vez, é definida como a seqüência de cruzamentos que, ao ser percorrida por um pacote, o conduzirá a seu destinatário. A *conectividade* de uma *rota*, por sua vez, é o produto das *conectividades retas* de todos os seus *segmentos*, bem como das *conectividades* de cada cruzamento onde dois segmentos consecutivos formam um ângulo diferente de  $180^0$ .

### Grafos Multiplanares

Muitos trabalhos sobre roteamento em cenários urbanos mapeiam a topologia das ruas e avenidas em grafos, cujos vértices são cruzamentos e arestas são *segmentos*. A Figura 4.2(b) é a representação do cruzamento da Figura 4.2(a) segundo essa abordagem. Contudo, ela não seria adequada ao TLAR, uma vez que a *conectividade* de um cruzamento só deve ser levada em consideração se o ângulo entre os segmentos anterior e posterior a ele na *rota* for diferente de  $180^0$ , conforme definido na Seção 4.2.2.

Assim, esta proposta associa à topologia de uma cidade um grafo composto de vários planos paralelos, denominado *grafo multiplanar* (MIN08). Estruturas de dados semelhantes já foram usadas com sucesso em outras aplicações, a exemplo de (FRA02) e (KAR97). Especificamente, (FRA02) propõe o uso de grafos multiplanares para otimizar o processamento de gran-

des quantidades de consultas (*queries*) de caminhos mais curtos aplicadas a um mesmo grafo. Situações como essa podem ocorrer em aplicações como planejamento de tráfego veicular (CAR94, ISH91, JAC99, JUN96, SHE93) ou consultas a bases de dados e à Internet (BAR00, SHE97).

Outro problema em que os grafos multiplanares podem ser utilizados é o particionamento de hipergrafos. Um hipergrafo é uma generalização dos grafos convencionais em que as arestas podem conectar um número arbitrário (mas não nulo) de vértices. O particionamento de grafos (ou hipergrafos) consiste em dividir seus vértices em  $k$  conjuntos de cardinalidade aproximadamente igual, minimizando o custo das arestas que conectam vértices situados em partições diferentes. Esse problema tem aplicações em várias áreas, como projeto VLSI (ALP95) e mineração de dados (*data mining*) (EUI98).

O algoritmo TLAR, por sua vez, emprega esse grafo para determinar qual a é rota de maior *conectividade total* entre o remetente e o destinatário de um pacote. A idéia principal dessa modelagem é separar as arestas paralelas em um mesmo plano, de modo que qualquer caminho nesse plano envolva somente o custo referente à *conectividade reta*. Por outro lado, se uma rota requerer o envio de pacotes para uma rua transversal (ou incidente), ela envolverá arestas que conectam vértices em planos diferentes no grafo.

Formalmente, cada cruzamento  $X$  é representado por um conjunto de pontos  $R_x = \{X_1, X_2, \dots, X_n\}$ , um único ponto em cada plano, onde  $n$  é o número de planos do grafo,  $X_p$  é o *representante de X* no plano  $p$  e  $R_x$  é chamado *conjunto de representantes de X*.

Além disso, cada *segmento* entre os cruzamentos  $X$  e  $Y$  corresponde a *uma única* aresta bidirecional no grafo,  $X_p Y_p$ , chamada *aresta intraplana*, conectando os vértices  $X_p$  e  $Y_p$ , ambos pertencentes ao mesmo plano  $p$ . Duas arestas são coplanares se, e somente se, o ângulo que seus respectivos *segmentos* formam com um eixo horizontal imaginário for o mesmo. Desse modo, cada plano ou possui uma única aresta ou todas as que contém são paralelas duas a duas.

Adicionalmente, há uma aresta  $X_p X_q$  conectando, dois a dois, todos os representantes de um mesmo vértice  $X$  que estiverem conectados a pelo menos um representante de outro vértice. Tais arestas são ditas *interplanares*, e representam o custo de enviar um pacote por uma rua diferente em um cruzamento, ou seja, a *conectividade cruzada*.

As Figuras 4.2(b) e 4.2(c) mostram os grafos resultantes do cenário da Figura 4.2(a) pelas duas abordagens. Na Figura 4.2(b), a representação tradicional conecta um cruzamento com seus *cruzamentos consecutivos* (não exibidos). Seu *grafo multiplana*r correspondente está na Figura 4.2(c), cujas arestas

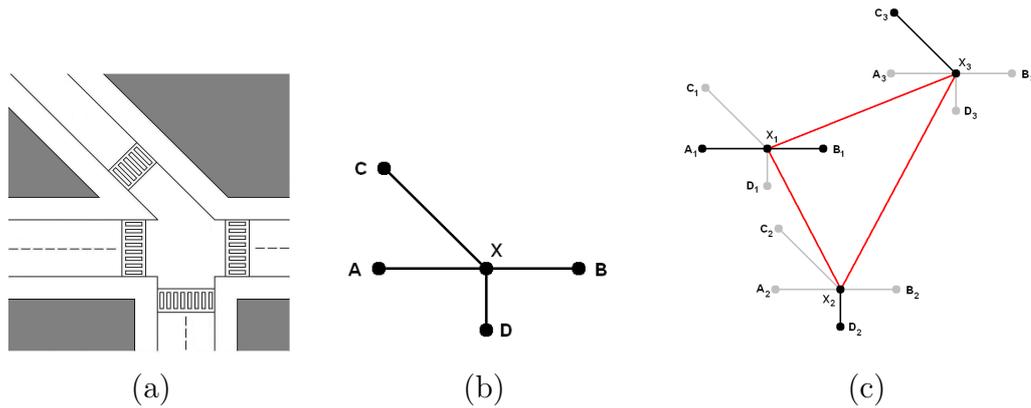


Figura 4.2: (a) Exemplo de cenário para conversão em grafo. (b) Representação tradicional em grafo. (c) *Grafo multiplanar* correspondente

*interplanares* são traçadas em vermelho, enquanto as “arestas” cinzas, na verdade, *não* fazem parte do grafo; apenas indicam que cada plano é composto por um subconjunto das arestas pertencentes à representação tradicional do grafo.

O objetivo de um *grafo multiplanar* é uniformizar e simplificar o cálculo de rotas pelo TLAR, reduzindo-o ao problema tradicional de se encontrar o caminho de menor custo em um grafo. Conforme mencionado nesta seção, no caso da heurística construtiva do TLAR, essa rota corresponde àquela de maior probabilidade de entrega de um pacote entre seu remetente e destinatário. Cada aresta entre *representantes* de cruzamentos diferentes tem peso igual à *conectividade reta* do respectivo *segmento*; as arestas *interplanares*, por sua vez, têm custo igual à *conectividade* do cruzamento em questão. Finalmente, o custo de qualquer caminho em um *grafo multiplanar* consiste no produto dos pesos das arestas que compõem esse caminho.

Desse modo, o custo de um caminho só levará em conta a *conectividade* de um cruzamento se, se somente se, esse caminho envolver arestas *interplanares*, o que reflete a transmissão entre ruas diferentes em um cruzamento. É importante observar que esse efeito não pode ser conseguido com a representação em grafo tradicional. A título de exemplo, na Figura 4.2(b), o custo do caminho  $AD$  seria o produto entre os pesos de  $AX$  e  $XD$ , o que não levaria em conta a *conectividade cruzada* em  $X$ . O mesmo caminho, no grafo da Figura 4.2(c), teria custo igual ao produto entre os pesos de  $A_1X_1$ ,  $X_1X_2$  e  $X_2D_2$ , sendo  $X_1X_2$  a *conectividade cruzada* em  $X$ .

Outra observação interessante é a de um grafo de um único plano, a exemplo do que é mostrado na Figura 4.2(b), consiste em um caso particular de um grafo multiplanar onde a *conectividade* dos cruzamentos é definida como sendo 1 (*conectividade* perfeita ou total).

### 4.2.3

#### Heurística Construtiva

No TLAR, a construção de rotas é feita de forma distribuída e sob demanda, ou seja, somente o próximo *segmento* de uma rota é calculado de cada vez. Quando o pacote chega a um veículo situado nesse *segmento*, a heurística construtiva é acionada para determinar o *segmento* seguinte (Seção 4.2.5). Esse processo se repete até que o pacote chegue ao seu destinatário ou seja descartado em virtude de seu tempo de vida ter se esgotado. A estratégia de reparo e o tratamento de ótimos locais é discutido na Seção 4.2.6.

Uma vez conhecido o próximo *segmento*, torna-se necessário selecionar para qual veículo o pacote deve ser encaminhado a seguir. Por esse motivo, cada veículo mantém uma lista de todos os outros com os quais está *comunicável diretamente*, o que é discutido nesta Seção. Os veículos também usam essa lista para fazer ajustes nas *conectividades* de ruas e de cruzamentos próximos, como explicado na Seção 4.2.4.

#### Lista de Vizinhos

Define-se como *vizinho* de um nó  $A$  qualquer veículo  $V$  tal que  $A$  e  $V$  estejam *comunicáveis diretamente* entre si. Muitos algoritmos de roteamento, inclusive o TLAR, mantêm em cada nó uma lista de vizinhos que é atualizada por meio de *beacons* — pequenas mensagens que um veículo difunde (para veículos próximos) a fim de divulgar alguma informação, como sua posição ou velocidade.

A lista de *vizinhos* pode ser mais ou menos fiel à realidade, dependendo do período com que cada nó envia seus *beacons*, e o percentual dos *beacons* que são perdidos por problemas como colisões no acesso ao meio sem fio, mobilidade dos veículos e obstrução de sinal. Por tais motivos, se um nó  $A$  não recebeu um *beacon* de um veículo  $V$ , isso não significa, necessariamente, que  $V$  deixou de ser *vizinho* de  $A$ . A solução mais comum consiste em concluir a quebra da comunicação direta entre dois veículos somente após um intervalo de tempo  $\Delta t_L$  desde que  $A$  recebeu o último *beacon* de  $V$ . Mesmo assim, protocolos de atualização de vizinhos estão sujeitos a:

- *falsos positivos* — se  $V$  ainda pertence à lista de *vizinhos* de  $A$ , mas ter deixado de estar *comunicável diretamente* com aquele veículo.
- *falsos negativos* — se nenhum *beacon* emitido por  $V$  for recebido por  $A$  no período de  $\Delta t_L$ , apesar de  $V$  ainda estar *comunicável diretamente* com  $A$ . Outro período de ocorrência de um falso positivo é o meio-tempo entre  $V$  se tornar *vizinho* de  $A$ , e  $A$  receber o primeiro *beacon* de  $V$ .

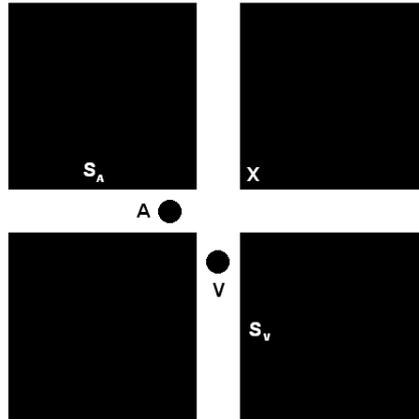


Figura 4.3: Uma situação onde  $CC(X)$  é ajustada para 100%.

Na versão atual do TLAR, cada *beacon* contém apenas a *posição* do veículo que o enviou. A velocidade e a direção de seu movimento poderiam ser derivadas a partir dessa informação, mas tal otimização está além do escopo da tese. Tendo em vista que o funcionamento do TLAR é independente dos procedimentos de manutenção da lista de vizinhos, abordagens mais sofisticadas para esse propósito poderão ser incorporadas a futuras versões do TLAR sem exigir modificações adicionais no protocolo de roteamento propriamente dito.

#### 4.2.4 Ajustes Locais

Como discutido na Seção 4.2.2, as *conectividades reta* e *cruzada* dependem das posições de *todos* os veículos em cada rua. Tendo em vista a impossibilidade de prever as posições *exatas* de cada veículo, e o fato de que nem todos os veículos executam o TLAR (são *equipados*), existe um certo grau de incerteza nesses cálculos, representado numericamente em termos de *probabilidades*. Com o propósito de atenuar essas incertezas, TLAR ajusta as *conectividades* de ruas e cruzamentos próximos de cada veículo de acordo sua lista de *vizinhos*.

Especificamente, seja  $A$  um *veículo equipado* situado em um *segmento*  $S_A$ , e  $V$  um *vizinho*, localizado em um *segmento*  $S_V$ , tais que  $S_A$  e  $S_V$  sejam transversais e tenham o cruzamento  $X$  como interseção, como mostra a Figura 4.3. Pela definição de *conectividade cruzada*, é possível concluir que  $CC(X) = 100\%$ .

A *conectividade reta*, por sua vez, pode ser ajustada como mostra a Figura 4.4. Se  $r$  é o alcance de transmissão sem fio,  $\overline{XV} < r$ ,  $\overline{YV} < r$  e  $V$  for um *vizinho* de  $A$ , a *conectividade reta* do *segmento*  $XY$  pode ser ajustada para 100% pois um par de *veículos equipados* situados nos cruzamentos  $X$  e

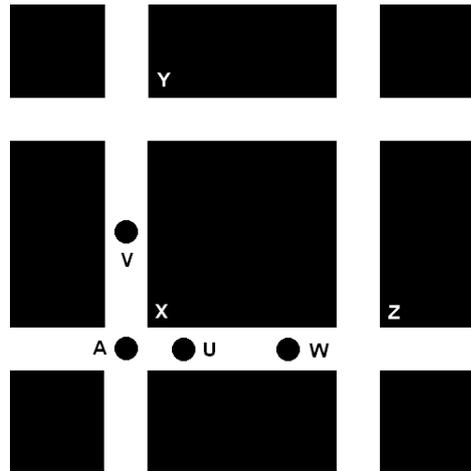


Figura 4.4: Exemplo de ajuste para *conectividades retas*.

Campo	Tipo	Descrição
<i>destAddr</i>	<i>address</i>	Endereço do destinatário
<i>destPos</i>	<i>point</i>	Posição do destinatário
<i>srcAddr</i>	<i>address</i>	Endereço do remetente
<i>srcPos</i>	<i>point</i>	Posição do remetente
<i>prevAddr</i>	<i>address</i>	Endereço do <i>hop</i> anterior
<i>prevPos</i>	<i>point</i>	Posição do <i>hop</i> anterior
<i>nextSeg</i>	<i>point</i>	Próximo <i>segmento</i> da rota
<i>prevSeg</i>	<i>point</i>	<i>Segmento</i> anterior da rota
<i>ttl</i>	<i>int</i>	Tempo de vida do pacote, em <i>hops</i>
<i>seqNum</i>	<i>int</i>	Número de seqüência do pacote

Tabela 4.1: Formato do cabeçalho de um pacote do TLAR.

*Y* seriam *comunicáveis indiretamente* entre si por meio de *V*.

#### 4.2.5 Construção de Rotas

Como mencionado no início da Seção 4.2.3, uma rota no TLAR consiste em uma seqüência de *segmentos* determinada de forma distribuída e sob demanda. Por esse motivo, como mostra a Tabela 4.1, um dos campos do cabeçalho de cada pacote é o próximo *segmento* de sua rota. Os demais campos serão explicados ao longo deste capítulo.

Assim, seja *V* um veículo situado no *segmento XY* que precisa rotear um pacote cujo próximo *segmento* é *YZ*, como mostra a Figura 4.5. Se *V* possuir *vizinhos* em *YZ*, então deverá encaminhar o pacote para aquele que estiver mais distante de *Y*, no caso *B*. Se isso não for possível, o *vizinho* que estiver mais próximo de *Y* deverá ser selecionado como o próximo *hop* da rota (o veículo *D* no exemplo da Figura 4.5). Contudo, se *nenhum* vizinho de *V* estiver mais próximo de *Y* do que o próprio *V*, um *ótimo local* foi atingido.

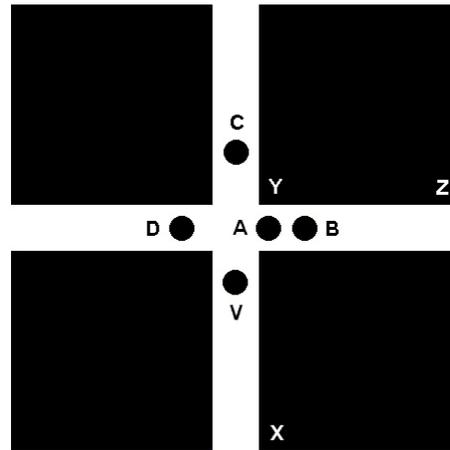


Figura 4.5: Selecionando o próximo *hop* de uma rota.

Essa situação e a estratégia de reparo são discutidas na Seção 4.2.6.

Quando um pacote é enviado para um veículo que já se encontra no próximo *segmento* de sua rota, torna-se necessário determinar o veículo seguinte. Para isso, o TLAR executa uma variante do algoritmo de *Dijkstra* (Algoritmo 1) sobre o *grafo multiplano* construído a partir das *conectividades* armazenadas na memória do veículo que estiver computando a rota.

Nesse algoritmo,  $G(V, E)$  é um *grafo multiplano*,  $weigh(X, Y)$  o peso da aresta  $XY$ ,  $S$  é o vértice de origem e  $D$  o de destino de uma rota. Para todo vértice  $x$  desse grafo,  $x.conn$  é a *conectividade total* da rota de  $S$  até  $x$ , enquanto  $x.prev$  é o vértice que precede  $x$  nessa rota. Finalmente, um vértice  $x$  é dito *aberto* se a rota de maior conectividade de  $S$  a  $x$  ainda não foi encontrada, e *fechado* caso contrário.

O algoritmo mantém uma lista de vértices abertos na variável *open Vertices*. A cada iteração do laço *while* (linhas 8–19), o vértice aberto cuja rota a  $S$  tiver a maior *conectividade total* é selecionado (variável *next*, linha 9) para ser fechado. A seguir, o laço *for* das linhas 11–18 calcula, para cada vértice  $y$  que seja vizinho de *next*, qual seria a *conectividade total* da rota que chegasse a  $y$  tendo *next* como predecessor (ou seja,  $S \rightarrow next \rightarrow y$ ), armazenada na variável *newConn*, linha 12. Se esse valor é maior do que a *conectividade total* da melhor rota encontrada de  $S$  a  $y$  até o momento ( $y.conn$ ), então os campos  $y.conn$  e  $y.prev$  são atualizados de modo a registrar que a melhor rota de  $S$  a  $y$  passou a ser  $S \rightarrow next \rightarrow y$ , o que é feito pelo bloco *if* das linhas 13–17.

Assim, quando a lista de *vértices abertos* estiver vazia, cada vértice do grafo terá em seu campo *prev* o predecessor da melhor rota de  $S$  até si próprio. Contudo, o problema a ser resolvido consiste em descobrir qual é o *próximo* cruzamento de uma rota, e não o *anterior*. Isso pode ser feito invertendo-se

---

**Algoritmo 1** Encontrando a rota de maior conectividade em um *grafo multiplanar*.

---

```

1: for all  $x \in V$  do
2:    $x.conn \leftarrow 0.0$ ;
3:    $x.prev \leftarrow undefined$ ;
4: end for
5:  $S.conn \leftarrow 1.0$ ;
6:  $S.prev \leftarrow undefined$ ;
7:  $openVertices \leftarrow \{S\}$ ;
8: while  $openVertices \neq \{\}$  do
9:    $next \leftarrow x, \forall y (y \in openVertices \rightarrow x.conn > y.conn)$ ;
10:   $openVertices \leftarrow openVertices - \{next\}$ ;
11:  for all  $y, (next, y) \in E$  do
12:     $newConn \leftarrow next.conn \times weigh(next, y)$ ;
13:    if  $newConn > y.conn$  then
14:       $openVertices \leftarrow openVertices \cup \{y\}$ 
15:       $y.conn \leftarrow newConn$ ;
16:       $y.prev \leftarrow next$ ;
17:    end if
18:  end for
19: end while

```

---

os cruzamentos de origem e o destino ao se aplicar o Algoritmo 1. Em outras palavras, encontrar o cruzamento seguinte a  $S$  em uma rota de  $S$  a  $D$  equivale a determinar o cruzamento anterior a  $S$  em uma rota de  $D$  a  $S$ . É importante observar, todavia, que essa inversão só pode ser feita porque as arestas do *grafo multiplanar* são bidirecionais.

Uma limitação conhecida do algoritmo de Dijkstra é que ele pressupõe a inexistência de ciclos (*loops*) de custo negativo. Essa restrição corresponderia, no *grafo multiplanar*, à inexistência de ciclos cujo custo (produto das *conectividades* de suas arestas) fosse maior que 1. Essa restrição nunca é violada pelo TLAR, uma vez que *todas* as arestas do *grafo multiplanar* são valores de *conectividade*, e, portanto, valores no intervalo  $[0, 1]$ .

Informalmente, a correção do Algoritmo 1 pode ser verificada como se segue: se um vértice  $next$  que pertence a  $openVertices$  é fechado (removido desse conjunto) então a rota de maior *conectividade total* de  $S$  a  $next$  é tal que o predecessor de  $next$  é  $next.prev$ . Isso pode ser mostrado por meio de indução: a melhor rota de  $S$  a si mesmo é trivial, o que constitui a base da indução. Se  $ST$  é a aresta de peso máximo partindo de  $S$ , então a melhor rota de  $S$  a  $T$  é justamente seguir por essa aresta, ou seja,  $T.prev = S$ . Tal afirmação pode ser provada por contradição: se  $SU$  é outra aresta que parte de  $S$  tal que  $weigh(S, U) < weigh(S, T)$ , seria necessário que existisse um caminho de  $U$  a  $T$  que tivesse *conectividade total* maior do que 1 para que a *conectividade*

*total* da rota  $S \rightarrow U \rightarrow T$  fosse maior do que o peso da aresta  $ST$ , o que é impossível. O passo de indução, por sua vez, consiste em usar esse mesmo argumento para constatar que nenhum outro *vértice aberto*  $x$  diferente de *next* poderia ser usado como rota alternativa de  $S$  a *next*, pois demandaria uma rota de  $x$  a *next* com *conectividade total* superior a 1. Finalmente, um vértice  $y$  só tem seu predecessor alterado para *next* se a rota  $S \rightarrow next \rightarrow y$  tiver uma *conectividade total* maior do que a da melhor rota encontrada até o momento (ou se nenhuma rota de  $S$  a  $y$  tiver sido encontrada até então).

#### 4.2.6 Estratégia de Reparo

Apesar de a heurística construtiva apresentada na Seção 4.2.3 selecionar as rotas de maior *conectividade*, sempre existe a possibilidade de ela conduzir um pacote a um *ótimo local*, ou seja, a um *segmento* em que não exista um veículo para prosseguir com a transmissão. Em princípio, uma nova rota poderia ser computada eliminando-se temporariamente o *segmento* em questão e reaplicando-se o Algoritmo 1 ao *grafo multiplanar* resultante.

Quando o pacote chegar ao próximo *segmento* da nova rota, a heurística construtiva seria usada para determinar para qual *segmento* encaminhá-lo a seguir, possivelmente reenviando-o novamente ao *segmento* onde o *ótimo local* foi detectado. Desse modo, poderia acontecer que dois veículos encaminhassem o pacote insistentemente de um para o outro, até que seu tempo de vida (TTL — *time to live*) chegasse a zero e ele fosse descartado.

A estratégia de reparo do TLAR explora o fato de que os pacotes ora seguem ao longo da rota de maior *conectividade* na direção do destinatário, ora se afastam dos *ótimos locais* à medida em que estes são detectados. Em ambos os casos, isso sugere que os *segmentos* pelos quais um pacote já passou não devem fazer parte da rota remanescente até o destinatário e, portanto, devem ser evitados.

Entretanto, existem situações em que, após atingir um *ótimo local*, a melhor rota alternativa consiste, justamente, em encaminhar o pacote por *segmentos* por onde este já passou. Por esse motivo, a estratégia de reparo do TLAR não deve *impedir* um pacote de retornar a algum *segmento*, mas sim de *priorizar* os demais *segmentos* quando for necessário selecionar o próximo *segmento* de uma rota.

Esse efeito é conseguido aplicando-se *penalidades* às *conectividades* dos *segmentos* já percorridos por um pacote, como explicado na Seção 4.2.6). Essa técnica é conhecida para evitar laços (*loops*) em roteamento (KAS02, LIA02, WAN03), e é implementada no TLAR como uma pequena modificação em sua

heurística construtiva, discutida na Seção 4.2.6.

### Tabela de Penalidades

Toda vez que um pacote passa por um *segmento* sua *conectividade* é penalizada (i.e. reduzida), o que é feito ao ser multiplicada por uma *constante de penalidade*  $q \in [0, 1]$ . Assim, se  $C_0$  é a *conectividade* inicial de um *segmento*,  $n$  o número de vezes que um pacote o percorreu e  $q$  a *constante de penalidade*, a nova *conectividade* desse *segmento*,  $C$ , é obtida pela Equação 4-5.

$$C = C_0 \times q^n \quad (4-5)$$

A aplicação da Equação 4-5 causa o efeito desejado na heurística construtiva, pois desestimula os veículos a transmitirem um pacote ao longo de ciclos (*loops*). Isso ocorre porque, à medida que um pacote percorra um ciclo, a *conectividade* de cada *segmento* envolvido decresceria em progressão geométrica, fazendo com que a heurística construtiva naturalmente selecionasse *segmentos* alternativos de *conectividade maior*. A Equação 4-5 também mostra que a estratégia de reparo nunca reduzirá a *conectividade* de um *segmento* a zero, de modo que ele sempre poderá ser reutilizado pela heurística construtiva caso os demais *segmentos* também apresentem baixa *conectividade*. Contudo, a mobilidade dos veículos pode tornar as penalidades dos *segmentos* obsoletas, o que requer que elas tenham um tempo de vida após o qual sejam desconsideradas.

Todavia, seria inviável armazenar no cabeçalho de um pacote os *segmentos* pelos quais ele já passou sem onerar excessivamente o desempenho do protocolo. Por isso, o TLAR mantém em cada veículo uma tabela chamada de *tabela de penalidades*, cujas entradas possuem o seguinte formato:

- *packetId* — o identificador de um pacote, que consiste no par ordenado formado pelo identificador do remetente do pacote e de seu número de seqüência;
- (*previousSeg*,  $n$ ) — um par ordenado que indica que o pacote de identificador *packetId* passou  $n$  vezes pelo *segmento previousSeg*;
- *neighborId* — o *vizinho* situado em *sourceSeg* que enviou o pacote para este veículo;
- *removeTime* — o instante de tempo em que essa entrada deve ser removida da tabela.

Toda vez que um veículo recebe um pacote, ele busca em sua *tabela de penalidades* por uma entrada cujos campos *packetId* e *previousSeg* sejam compatíveis com seus correspondentes no pacote. Se tal entrada existir, seu

campo  $n$  é incrementado e  $neighborId$  é atualizado com o endereço do vizinho do qual o veículo acaba de receber o pacote. Caso contrário, uma nova entrada é criada com  $n$  inicializado em 1,  $previousSeg$  é o último *segmento* percorrido pelo pacote e os demais campos são copiados de seu cabeçalho (listado na Tabela 4.1). Finalmente, cada entrada possui um temporizador, para que possa ser removida depois de algum tempo.

Para ilustrarmos o funcionamento da estratégia de reparo do TLAR, reconsideremos a Figura 4.5. Suponhamos que o veículo  $V$ , situado no *segmento*  $XY$ , encaminhou um pacote a  $B$ , localizado em  $YZ$ . Ao recebê-lo,  $B$  buscará em sua *tabela de penalidades* uma entrada cujo campo  $packetId$  coincida com o remetente e o número de seqüência do pacote, que satisfaça a condição  $previousSeg = XY$ . Encontrando-a,  $B$  incrementará seu contador  $n$ ; senão criará uma nova entrada com  $(previousSeg, n) = (XY, 1)$ ,  $neighborId = V$  e  $packetId$  inicializado com o remetente e o número de seqüência do pacote.

Se  $YZ$  era o próximo *segmento* da rota do pacote, caberá a  $B$  determinar o seguinte. Ao montar o *grafo multiplanar*, ele percorrerá sua *tabela de penalidades* e, para cada entrada cujo  $packetId$  seja igual ao do pacote, aplicará a Equação 4-5 ao *segmento* do campo  $previousSeg$  com o valor do respectivo contador  $n$ . A seguir,  $B$  aplica o Algoritmo 1 para determinar o próximo *segmento* da rota, como explicado na Seção 1.

### Super-vizinhos

Uma limitação da estratégia de reparo discutida neste capítulo é que somente o veículo que recebe um pacote atualiza sua *tabela de penalidades*. Assim, se um mesmo pacote atravessar um *segmento* várias vezes por meio de veículos diferentes, o valor real do contador  $n$  relativo a esse *segmento* seria distribuído entre esses vários veículos.

De modo a concentrar em um único veículo a contagem do número de vezes que um mesmo pacote percorre um *segmento* específico, a versão atual do TLAR utiliza o conceito de *super-vizinhos*. Um veículo  $W$  é chamado *super-vizinho* de  $V$  se, e somente se, existir uma entrada na *tabela de penalidades* de  $V$  cujo campo  $neighborId$  se refira a  $W$ .

Assim, a heurística construtiva do TLAR é modificada para que o próximo *hop* da rota seja escolhido exclusivamente entre os *super-vizinhos* do veículo em questão. Se nenhum deles proporcionar progresso na direção do próximo *segmento*, a seleção passa a ser feita levando-se em conta *todos* os vizinhos, como explicado na Seção 4.2.5.

A versão completa do TLAR é resumida no Algoritmo 2. As linhas 5–7 constróem o grafo multiplanar (Seção 4.2.2), aplicam os ajustes locais (Seção

**Algoritmo 2** Versão completa do TLAR.

---

```

1: if neighbors == {} then
2:   descartar pacote;
3: end if
4:
5:  $G \leftarrow \text{buildMultiplanarGraph}()$ ;
6:  $\text{locallyAdjust}(G, \text{neighbors})$ ;
7:  $\text{penalize}(G, \text{penaltyTable})$ ;
8:
9: while true do
10:   $\text{nextSegment} \leftarrow \text{Dijkstra}(G)$ ;
11:
12:   $\text{nextHop} \leftarrow \text{selectHop}(\text{nextSegment}, \text{superNeighbors})$ ;
13:  if  $\text{nextHop} \neq \text{myId}$  then
14:    break;
15:  end if
16:
17:   $\text{nextHop} \leftarrow \text{selectHop}(\text{nextSegment}, \text{neighbors})$ ;
18:  if  $\text{nextHop} \neq \text{myId}$  then
19:    break;
20:  end if
21:
22:   $G.\text{erase}(\text{nextSegment})$ ;
23: end while
24: return nextHop;

```

---

4.2.4) e as penalidades (Seção 4.2.6). A seguir, o laço *while* aplica o algoritmo de Dijkstra modificado (Algoritmo 1) para determinar o próximo *segmento* da rota. O *hop* seguinte é selecionado da lista de *super-vizinhos* (linha 12). Se nenhum deles proporcionar progresso na direção do próximo *segmento*, a seleção é feita dentre todos os *vizinhos* (linha 17). Caso essa segunda tentativa falhe, o algoritmo elimina a aresta em questão do grafo (linha 22) e prossegue no laço *while* para nova tentativa de roteamento.

O principal problema dessa abordagem é que, devido à mobilidade nas VANETs, um veículo pode não estar mais *comunicável diretamente* a algum de seus *super-vizinhos*. Nesse caso, a informação de estado relativa a tal *super-vizinho* na *tabela de penalidades* seria perdida. Tendo em vista que o tempo de envio de um pacote entre dois veículos consecutivos de uma rota é da ordem de milissegundos, a perda de um *super-vizinho* tenderá a acontecer principalmente em condições com alto tráfego de pacotes, proporcionando filas de transmissão maiores em cada veículo e, conseqüentemente, aumentando a latência de transmissão.

Outra situação em que a seqüência de *super-vizinhos* de uma rota tende a se romper ocorre quando uma mensagem, na tentativa de transpor uma região

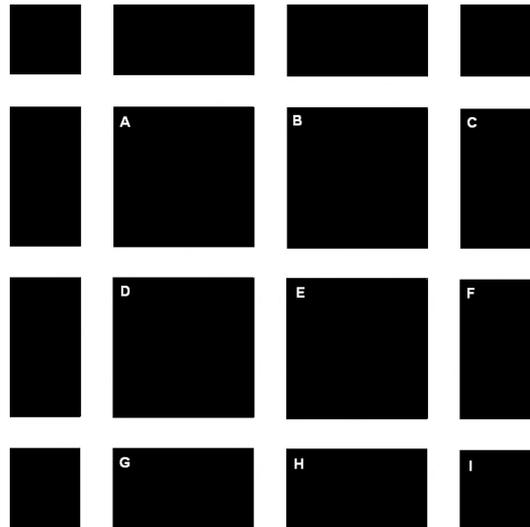


Figura 4.6: Exemplo de situação em que pode ocorrer a quebra da seqüência de super-vizinhos.

de ótimo local, é encaminhada de volta a um veículo após percorrer vários *segmentos*. Quanto maior o número de *segmentos* que uma mensagem percorre antes de voltar a um veículo específico, maior o tempo gasto neste percurso e, devido a isso, maior a probabilidade de a comunicação direta entre esse veículo e seus *super-vizinhos* não ser mais possível. Porém, conforme discutido a seguir, a probabilidade de um pacote ser encaminhado de volta a um veículo qualquer decresce à medida em que a distância desse veículo à região do ótimo local aumenta. Logo, a quebra da seqüência de super-vizinhos tende a ser um evento raro.

A Figura 4.6 mostra um exemplo em que isso pode acontecer. Um veículo situado próximo ao cruzamento *D* envia um pacote para um destinatário localizado em *F*. O pacote é encaminhado pelo *segmento DE* e, a seguir, por *EF*, onde um ótimo local é detectado. Nesse momento, o pacote é roteado de volta ao cruzamento *E* para que siga em direção ao seu destinatário através de *EH*. Supondo que este *segmento* também não permita o encaminhamento do pacote adiante (por não possuir *veículos equipados* suficientes), ele será retransmitido de volta a *E* para que seja encaminhado por *EB*. Se o roteamento por este *segmento* também não for possível, será necessário encaminhar o pacote mais uma vez para *E* e, então, novamente para *D*. A partir desse cruzamento, uma nova rota para *F* deveria ser computada, mas esse procedimento é desnecessário para exemplificar a quebra da seqüência de *super-vizinhos*, e, portanto, omitida.

Sabendo-se que a probabilidade de um pacote se propagar por um *segmento XY* é  $CR(XY)$ , e que a probabilidade de ele não poder se propagar por ele é  $1 - CR(XY)$ , a probabilidade  $P$  de a situação proposta no parágrafo

anterior se concretizar pode ser calculada pela Equação 4-6.

$$P = CR(DE) \times (1 - CR(EF)) \times (1 - CR(EH)) \times (1 - CR(EB)) \times CC(E) \quad (4-6)$$

Tendo em vista que cada termo do produto da Equação 4-6 se encontra no intervalo  $[0, 1]$ , é possível concluir que  $P$  tende a diminuir com o aumento do número de termos, e que  $P$  será menor que o menor deles.

#### 4.2.7

##### Discussão

Em áreas urbanas, a variação do estado dos semáforos causa flutuações temporárias, mas significativas, da densidade de tráfego nas ruas. Quando os semáforos estão sincronizados, essas flutuações tendem a ocorrer de forma aproximadamente simétrica, resultando em ruas com conectividades semelhantes. Por outro lado, semáforos dessincronizados proporcionam distribuições assimétricas de tráfego, em que, a cada instante de tempo, a conectividade de algumas ruas se sobressai, tornando-as mais atraentes à heurística construtiva do TLAR.

Uma desvantagem dessa heurística é que, dependendo das posições dos remetentes e destinatários de transmissões simultâneas, a concentração de rotas por ruas com maior densidade de tráfego veicular pode causar congestionamento no meio sem fio. Algoritmos mais sofisticados precisam considerar, portanto, a distribuição do tráfego de pacotes pela rede. Essa abordagem, naturalmente, requer a existência de múltiplos caminhos entre remetente e destinatário de um pacote. Essa condição, por outro lado, normalmente não pode ser satisfeita quando a densidade de *veículos equipados* é baixa. Nesse caso, tornam-se necessárias estratégias como armazenamento temporário (*store-and-forward*) e o envio simultâneo de duplicatas de cada pacote por rotas diferentes, a fim de aumentar a probabilidade de entrega. Em virtude de limitações de tempo, o estudo dessas possibilidades foi deixado para trabalhos futuros.

As limitações do algoritmo TLAR expostas nesta seção restringem as aplicações a que ele se destina, principalmente, às da classe de conveniência. A falta de um controle de congestionamento de pacotes o torna inadequado a transmissões simultâneas de grandes volumes de dados, como ocorre em aplicações de transferência de arquivos ou de multimídia, sendo que estas últimas possuem requisitos de qualidade de serviço adicionais. Por esse motivo, o algoritmo TLAR seria mais apropriado a aplicações com requisitos menos rígidos de largura de banda e latência, tais como envio de mensagens de texto curtas, serviços de informações textuais, etc.

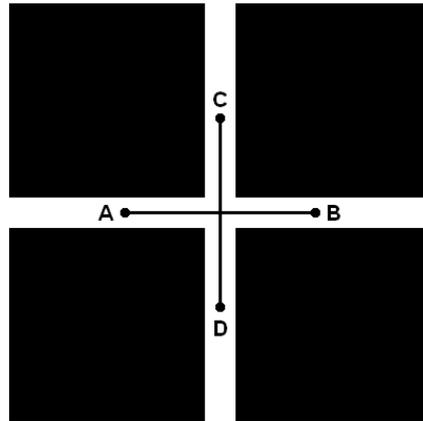


Figura 4.7: Prédios podem impedir a detecção de arestas que se cruzam.

### 4.3 U-GPSR

U-GPSR (*Urban GPSR*) é uma versão adaptada do GPSR (*Greedy Perimeter Stateless Routing*) (KAR00) para áreas urbanas, proposto com o intuito de comparação de desempenho com o TLAR. Conforme discutido na Seção 3.5, o algoritmo GPSR consiste, resumidamente, em selecionar como próximo veículo de uma rota aquele que se encontra mais próximo do destinatário do pacote sendo roteado. Se um ótimo local for atingido, a regra da mão direita é utilizada a fim de encontrar um caminho que contorne a região onde o ótimo local foi detectado.

Uma limitação importante de algoritmos de planarização como o apresentado na Seção 3.5 é que obstáculos (como prédios, morros, etc.) podem impedir a detecção de algumas arestas que se cruzam. A Figura 4.7 mostra uma situação onde isso pode acontecer: dois pares de veículos que estão *comunicáveis diretamente* entre si ( $A, B$ ) e ( $C, D$ ) formam arestas que se cruzam e não podem ser detectadas.

Contudo, em cenários urbanos, a própria topologia das ruas já constitui um grafo planar. Em vista disso, esta tese propõe o algoritmo de roteamento U-GPSR, que adota a mesma heurística construtiva do GPSR, mas combinada com uma versão modificada da estratégia de reparo daquele protocolo. Ao contrário de sua versão original, que determina a rota alternativa em função da topologia da VANET (grafo formado por veículos), a abordagem do U-GPSR o faz a partir do grafo urbano.

Naturalmente, a estratégia de reparo do U-GPSR requer que cada veículo disponha, além de sua posição atual, um sistema de mapas digital. Quando o roteamento guloso atinge um ótimo local, a estratégia de reparo usa a regra da mão direita para escolher por qual rua o pacote deve ser encaminhado a

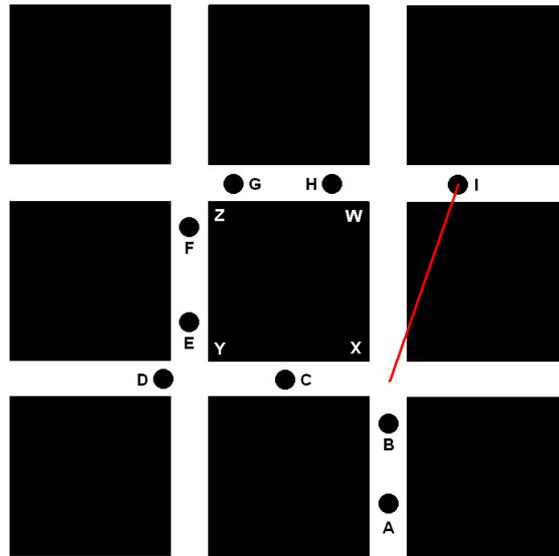


Figura 4.8: Exemplo de aplicação da estratégia de reparo do U-GPSR.

seguir.

Dada a posição atual de um veículo e o *segmento* obtido pela aplicação da regra da mão direita, o próximo veículo da rota é selecionado da mesma forma como no TLAR (vide Seção 4.2.5), e exemplificado na Figura 4.5.

### 4.3.1 Exemplo

O funcionamento da estratégia de reparo do U-GPSR é ilustrado na Figura 4.8, que mostra um cenário contendo quatro cruzamentos ( $X$ ,  $Y$ ,  $Z$  e  $W$ ) e nove veículos ( $A - I$ ). Supondo que o veículo  $A$  envie uma mensagem destinada a  $I$ , que é encaminhada primeiramente a  $B$  devido à heurística gulosa. Porém, esse veículo não possui nenhum vizinho mais próximo de  $I$  do que si mesmo, ou seja, um ótimo local foi alcançado. Nesse momento, o modo de roteamento do pacote é ajustado para *REPAIR* e a distância deste ponto ao destinatário ( $\overline{BI}$ ) é armazenada no cabeçalho do pacote.

Traça-se então uma linha imaginária de  $X$  até  $I$ , indicada em vermelho, que sofre uma rotação em sentido anti-horário em torno de  $X$ . O primeiro *segmento* atingido pela rotação dessa linha é  $XW$ . Contudo,  $B$  não possui nem vizinhos nesse *segmento* e nem algum outro que esteja mais próximo de  $X$  do que si mesmo. Assim, uma nova rotação da linha imaginária é conduzida, deslocando-a para o *segmento*  $XY$ .

O único vizinho de  $B$  nesse *segmento* é  $C$ , veículo para o qual, então, o pacote é encaminhado. Este veículo, por sua vez, detecta que o pacote sendo roteado está em modo de reparo (informação obtida do cabeçalho). Tendo em

vista que o próximo cruzamento da rota é  $Y$  e o anterior é  $X$ , o algoritmo executa uma rotação de  $XY$  em sentido anti-horário em torno de  $Y$ , que atinge o *segmento*  $YZ$ . Apesar de  $C$  não possuir vizinhos em  $YZ$ , o pacote é encaminhado a  $D$  porque este veículo está mais próximo de  $Y$  do que si mesmo.

O pacote será encaminhado desse modo ao longo do *segmento*  $YZ$  até chegar a  $G$ . Pela rotação de  $YZ$  em sentido anti-horário em torno de  $Z$  obtém-se  $ZW$  como próximo *segmento*. O veículo  $G$  encaminha o pacote a  $H$ , que, por sua vez, retoma o roteamento guloso, tendo em vista que sua distância a  $J$  é inferior à de  $I$  ao ponto em que o pacote atingiu o ótimo local (obtida do cabeçalho do pacote), ou seja,  $\overline{HI} < \overline{BI}$ .

### 4.3.2

#### Discussão

Uma vantagem do U-GPSR é que os veículos não precisam mais manter uma versão planarizada da topologia da VANET em sua vizinhança, diminuindo a sobrecarga (*overhead*) de processamento. Além disso, como o mapa das ruas é conhecido e estático, o U-GPSR não é afetado pela incerteza no posicionamento dos veículos provocada pela mobilidade. Em contrapartida, sua estratégia de reparo não leva em consideração o volume de tráfego nas ruas e nem a distância a ser percorrida ao longo delas até o destinatário. Conseqüentemente, o uso da regra da mão direita pode conduzir pacotes por ruas pouco movimentadas, sem saída ou até mesmo que se afastem da localidade do destinatário.

Por esses motivos, o U-GPSR apresenta melhores resultados em áreas de alta densidade de tráfego e com topologia de ruas que permitam vários possíveis caminhos entre o remetente e o destinatário. A gama de aplicações tolerantes a essas restrições se limita, principalmente, àquelas que envolvem comunicação a distâncias curtas, a exemplo de anúncios, bate-papo (*chat*) entre passageiros, etc.