

2 Redes Ad-hoc – Projeto terminode

Redes móveis *ad hoc* são formadas sem a existência de uma administração central e são compostas por um conjunto de dispositivos móveis (chamados de nós) com capacidade de comunicação entre si, através de um enlace sem fio de banda relativamente restrita. As redes *ad hoc* idealmente funcionam de forma independente de qualquer infra-estrutura. Como os nós da rede são móveis, a topologia pode mudar rapidamente com o tempo e de forma imprevisível. A rede é descentralizada, onde toda a atividade deve ser executada pelos próprios nós, incluindo a descoberta de rotas e a entrega das mensagens.

2.1. Roteamento em redes ad hoc

Em redes móveis *ad hoc* a funcionalidade de roteamento é incorporada aos nós móveis, que atuam como roteadores e terminais. Essa característica permite que dois nós se comuniquem, mesmo estando fora do raio de transmissão radioelétrica um do outro. Essa comunicação é realizada utilizando os outros nós da rede como roteadores, ou seja, a comunicação entre fonte e destino do pacote pode acontecer através de múltiplos saltos. Devido à ausência de infra-estrutura de roteamento, os nós têm que cooperar para se comunicar. Isso significa que a cooperação entre nós precisa ocorrer no momento em que se deseja encontrar o caminho para um pacote chegar ao destino, e também para encaminhar o pacote através da rede [7].

Roteamento é uma função central em qualquer rede, e em redes *ad hoc* existem dois problemas a ela específicos. O primeiro problema consiste no fato de que as soluções tradicionais de roteamento para Internet ou redes celulares, se baseiam em uma rápida propagação da informação de mudanças na topologia, e por isso tais procedimentos são projetados assumindo que a rede é relativamente estável. Porém, em redes móveis *ad hoc* a topologia da rede está em constante

mudança. O segundo problema está ligado ao fato de que as soluções de roteamento tradicionais dependem de alguma forma de um banco de dados de roteamento distribuído, mantido pelos operadores das redes e instalados em alguns nós de administração especializados. Em redes *ad hoc*, não pode ser assumido que os nós possuem tais dispositivos de armazenamento de dados. Além disso, as informações armazenadas nos nós nem sempre são confiáveis [5].

O grupo MANET do IETF [8] estuda principalmente soluções que visam à mitigação dos efeitos decorrentes do problema da mudança constante de topologia. Tais soluções são baseadas na otimização dos algoritmos tradicionais como, por exemplo, o *distance vector* [9], *link state* [9], e *source routing* [10]. Uma abordagem diferente da realizada nesses algoritmos tradicionais pode ser feita baseando a decisão do algoritmo de roteamento no uso de informação da posição geográfica dos nós. Métodos de roteamento que utilizam essa abordagem são conhecidos como de roteamento geográfico ou de roteamento baseado em localização.

Nestes métodos, os nós possuem informação da sua própria posição geográfica, e das posições dos outros nós e o roteamento é baseado nessa informação. Métodos geográficos reduzem a função de roteamento dos sistemas intermediários ao mínimo, e não possuem a necessidade de uma base de dados distribuída. Além disso, eles podem ser usados como complementos dos métodos de roteamento tradicionais [5]. Esses algoritmos permitem explorar os seguintes fatos:

- Conforme a densidade de nós na rede sem fio aumenta, o caminho mais curto entre fonte e destino se aproxima de uma linha reta entre eles.
- Normalmente, em redes sem fio a proximidade geográfica entre nós determina se o enlace entre eles existe.

Estes métodos de roteamento geográficos permitem diminuir o volume de mensagens de controle, reduz os estados que precisam ser armazenadas nos nós, e reduz o risco que uma grande quantidade de informação armazenada sobre a rede

se torne inútil. Isso acontece, porque este tipo de algoritmo de roteamento realiza a descoberta de rotas utilizando somente o conhecimento de quais são os seus vizinhos imediatos na topologia, além da informação de localização. A localização é a posição física do nó e pode ser definida por sua latitude, longitude e altitude. Para que um nó conheça a sua própria localização é necessário que ele possua um sistema de posicionamento (o GPS, por exemplo).

A principal vantagem dos métodos de roteamento geográfico está relacionada com a sua escalabilidade em termos de número de nós e de cobertura geográfica da rede. Por isso, esse tipo de roteamento se mostra uma escolha interessante para redes *ad hoc* distribuídas sobre uma área geográfica extensa, como é o caso do canal de retorno para TV digital. Porém, estes métodos de roteamento dependem de um serviço de gerência de mobilidade que permita a aquisição e distribuição da informação de localização de cada nó através da rede. Cada nó utilizará este serviço para a descoberta da localização de outro nó da rede.

Portanto, para que o roteamento dos pacotes aconteça de maneira eficiente, deve existir um sistema de posicionamento e um serviço de gerência de mobilidade funcionando em conjunto com o protocolo de roteamento. Cada um desses serviços depende de uma distribuição eficiente de informações pela rede, e para isso é necessário que ocorra uma intensa cooperação entre os nós, que é uma das características principais de redes *ad hoc*. Esta cooperação ainda tem como vantagem aumentar a capacidade de transmissão da rede. Por isso, as redes móveis *ad hoc* precisam de algoritmos eficientes para determinar a organização da rede e para realizar o roteamento dos pacotes. Nesse ambiente de controle descentralizado surge a necessidade de auto-organização da rede, que deve ser baseada em um conjunto de regras e políticas, que simplifiquem a operação da rede e assegurar que o sistema permaneça sob controle.

A necessidade de autoconfiguração e flexibilidade nos vários níveis da rede, como o roteamento dinâmico e o controle de acesso ao meio distribuído, cria novos desafios para as redes *ad hoc* [7]. O Projeto Terminode surge com o objetivo de abordar as redes móveis *ad hoc* auto-organizáveis, utilizando um novo

protocolo de roteamento geográfico, o que faz deste projeto o objeto de estudo desse trabalho. A seguir é apresentada uma breve explicação sobre o Projeto Terminode, com foco no protocolo de roteamento.

2.2. Projeto Terminode

O projeto Terminodes é um projeto de pesquisa de longo prazo (2000 - 2010) objetivando estudar e criar protótipos de redes *ad hoc* auto-organizáveis de larga escala. O nome terminode é resultado da combinação de *terminal* e *node*, pois os nós atuam com terminais e nós ao mesmo tempo na rede. O projeto aborda todas as camadas, incluindo desde a camada física até a camada de aplicação e explora a interação entre elas [11].

Redes auto-organizadas se distinguem de outras redes móveis ad hoc, devido as suas seguintes características [2]:

- São redes sem autoridade central, ou seja, elas podem atuar de forma independente de qualquer provedor ou denominador comum, como a Internet. Por causa dessas características é uma forma de regulamentar a maneira que os nós devem cooperar entre si.
- São potencialmente grandes e são irregularmente distribuídas tanto em termos de densidade dos nós como em termo de topologia. A densidade de nós é possivelmente alta em pequenas áreas (cidades, por exemplo) e baixa em áreas extensas.
- São altamente cooperativas. As tarefas em qualquer camada são distribuídas pelos nós, e qualquer operação é o resultado da cooperação de um grupo deles.

Uma das partes mais importante o projeto Terminodes é o desenvolvimento do protocolo de roteamento, que possibilitará a escalabilidade da rede em termos

de número de nós e de área de cobertura. Essa é uma característica fundamental para redes auto-organizadas.

2.2.1. Os terminodes

Os terminodes são projetados para serem auto-organizáveis, ou seja, qualquer número de terminodes pode constituir uma rede e qualquer terminode pode se conectar automaticamente nessa rede. Assim, todos os terminodes devem desempenhar um conjunto mínimo e comum de funções que são necessárias e suficientes para a auto-operação da rede. Comparada as redes atuais, os mecanismos que incluem processamento ou armazenamento centralizado precisam ser substituídos com soluções completamente distribuídas. Contudo, isto não significa que todos os terminodes são idênticos. Um terminode pode possuir uma capacidade maior de processamento, armazenamento ou capacidade de transmissão do que os outros terminodes, o que pode beneficiar toda a comunidade de terminodes [11]. Nesse trabalho serão considerados apenas terminodes idênticos.

Cada terminode possui um identificador permanente único (EUI - *End-system Unique Identifier*) e um endereço temporário dependente da sua localização (LDA - *location-dependent address*). O LDA é simplesmente uma trinca de coordenadas geográficas (longitude, latitude, altitude), obtidas por algum método de posicionamento [5]. O terminode deve possuir um sistema de posicionamento que permita a ele descobrir sua localização geográfica. Esse sistema pode ser o GPS ou um outro método qualquer de posicionamento. Além, disso deve ser capaz de obter o LDA de um nó através do endereço EUI por meio do sistema de gerência de mobilidade.

Na rede terminode, dois importantes fatores afetam o roteamento de pacotes. O primeiro fator é a exigência de escalabilidade em termos de número de nós e de cobertura de área geográfica. O segundo está ligado ao fato de que os nós da rede são equipamentos de usuários, e podem estar disponíveis apenas esporadicamente.

Por isso, um terminode se baseia somente nele e em um pequeno número de outros terminodes para o roteamento. Além disso, este segundo fato cria a necessidade de algum incentivo para os usuários cooperarem e encaminharem pacotes originados por outros usuários [2].

Para encaminhar o pacote pela rede, o protocolo de roteamento dos terminodes utiliza o endereço (EUI) e a localização geográfica (LDA) do terminode de destino. Isso permite atender as exigências de escalabilidade, redução da função dos sistemas intermediários e robustez. Para atingir estes objetivos o roteamento na rede terminode é feito pela combinação de dois protocolos: o roteamento remoto do terminode (do Inglês, *terminode remote routing*, denotado por TRR) e o roteamento local do terminode (do Inglês, *terminode local routing*, denotado por TLR).

O TLR é um protocolo de roteamento do tipo *link state* que utiliza apenas o endereço EUI para a realização do roteamento. É o mecanismo que permite alcançar o terminode destino localizado em sua vizinhança [6].

O TRR baseia suas decisões na localização (LDA) do destino e é usado quando o destino está geograficamente longe da origem. Este protocolo é o elemento chave para atingir a escalabilidade. Se somente a informação de localização for usada, podem ocorrer erros ou *loops* de roteamento devido à imprecisão na localização do nó de destino (LDA) que a fonte possui. Para evitar esse problema, quando o pacote se aproxima da localização conhecida do destino, o roteamento passa a ser feito através do TLR. Esse procedimento aumenta a robustez do protocolo de roteamento [6].

Para o funcionamento correto do TRR, assume-se a existência de um sistema de localização composto de duas funções. A primeira consiste de um algoritmo de rastreamento da localização entre os nós quando eles estabelecem uma comunicação com sucesso. Isto permite que os terminodes que estejam se comunicando informem continuamente a atualização de seus LDAs ao outro terminode. A segunda função consiste em um serviço de descoberta de

localização, que é usado pela fonte para obter uma localização provável do destino D (LDA_D), que ainda não esteja sendo rastreado pelo método anterior [4].

A figura 2 ilustra o procedimento empregado no encaminhamento do pacote. O terminode S é a fonte dos pacotes. Os terminodes D1 e D2 são dois destino distintos para os pacotes. O destino D1 se localiza na vizinhança de S, e assim o protocolo utilizado para rotear o pacote é o TLR. O destino D2 está localizado geograficamente distante da fonte S, e então é usada a combinação do TLR e do TRR. O TRR é usado para rotear o pacote até um ponto próximo da localização conhecida do destino (no exemplo da figura 2, o terminode I). A partir desse ponto, o protocolo utilizado até o pacote chegar o destino passa a ser o TLR.

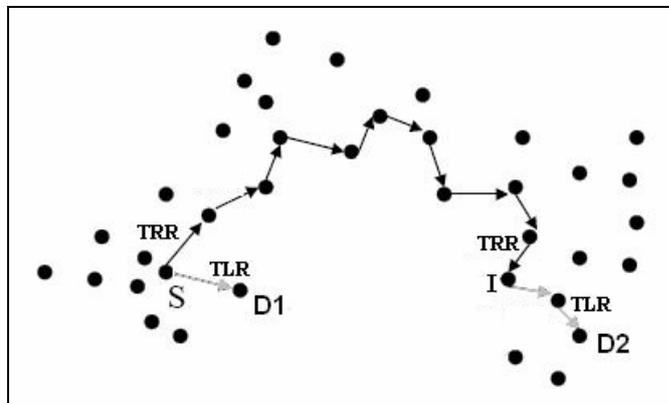


Figura 2 - exemplo do roteamento terminode

A seguir, são mostradas algumas características do projeto Terminodes necessárias como suporte para o protocolo de roteamento, que são: gerência de mobilidade, sistema de posicionamento e cooperação entre nós. O protocolo de roteamento é descrito em maiores detalhes, pois é um dos itens mais importantes de estudo dessa dissertação.

2.2.2. Gerência de Mobilidade

Algoritmos de roteamento baseados em localização visam enviar os pacotes na direção geográfica da posição do destino. Isto exige que cada nó possa obter a localização do nó destino do pacote através de um serviço de gerência de mobilidade (ou também chamada de serviço de localização). Como uma das

principais características das redes ad hoc é ser independente de infra-estrutura, esse serviço deve utilizar um algoritmo distribuído. Isso evita que uma falha em poucos nós da rede interrompam o serviço.

O principal objetivo da gerência de mobilidade é distribuir a informação de localização (LDA) ao longo da rede de maneira dinâmica, escalável, segura e sem privilegiar nenhum nó ou região. Através desse processo um terminode consegue obter a localização de qualquer terminode a partir de sua EUI. Não é necessário que a informação de localização mantida seja exata, mas é exigido que seja precisa o suficiente para que o pacote eventualmente chegue em algum terminode que encontre o destino dentro de sua área atingível pelo TLR [2].

Existem algumas arquiteturas diferentes para implementar o serviço de gerência de mobilidade podem ser encontradas na literatura [12, 13, 14], que possuem diferentes mecanismo para atualizar e distribuir a informação de localização. O projeto Terminode utiliza o conceito de região virtual doméstica (do Inglês, *virtual home region*, denotada por VHR) para realizar esse sistema de gerência de mobilidade [15], também utilizado em [14].

A região virtual doméstica de cada nó é definida pelo seu centro geográfico e por um raio. O centro geográfico é calculado através de alguma função de *hash* aplicada ao identificador único (EUI) do nó. A região virtual doméstica é formada por todos os nós que se localizam dentro da área delimitada pelo raio da região, conforme mostrado na figura 3. Todos os nós dentro da VHR de um certo nó devem armazenar a informação de localização atualizada deste nó.

Cada nó deve constantemente enviar mensagens para os nós localizados em sua VHR, atualizando sua informação de localização. Para obter a informação de localização deve ser aplicada a mesma função de *hash* ao EUI do nó desejado e enviar a solicitação para a posição obtida. Qualquer nó dentro da VHR pode responder a solicitação.

O raio da região é escolhido de forma a manter um número adequado de nós dentro da VHR. Ou seja, a região deve possuir um número mínimo de nós para

armazenar a informação e tornar o sistema robusto a falha de alguns nós. Porém, o raio não deve ser escolhido muito grande, pois nessa caso a região pode conter muitos nós, gerando uma elevada quantidade de mensagens necessárias para a gerência e acesso da informação de localização.

Na rede Terminode em complemento a essas funções, o sistema de mobilidade assume que um terminode tem a habilidade de localizar os terminodes na sua vizinhança. Além disso, existe um protocolo de rastreamento que mantém atualizadas as posições dos nós que estão se comunicando.

A figura 3 ilustra essa arquitetura, um terminode S deseja acessar a localização de D, conhecendo previamente o seu EUI, ele envia uma requisição para a VHR de D. Uma vez recebida a requisição, a localização de D é enviada pelos nós da VHR_D . Esta informação é usada para iniciar a comunicação com S. Durante a comunicação, D e S dependem de um sistema direto de rastreamento das posições, para que a localização atual de um seja sempre conhecida pela outro. A figura seguinte ilustra esse exemplo.

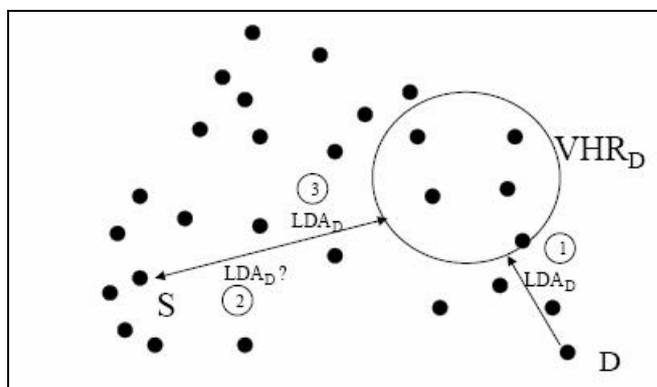


Figura 3 - exemplo da atualização e requisição da posição de um terminode.

Existem vários esquemas possíveis para a atualização da posição que podem ser aplicados na rede Terminode. Na implementação atual, utiliza-se um esquema de atualização baseado em distâncias. Neste esquema, cada terminode rastreia a distância que percorreu desde que enviou a última atualização de seu LDA para sua VHR, e envia novamente sua atualização se essa distância ultrapassa um certo limiar [2].

Para obter um gerenciamento com robustez e auto-operação, os terminodes precisam cooperar para armazenar a informação de localização e servir temporariamente como uma memória distribuída para outros terminodes. A vantagem dessa abordagem é que ela requer uma quantidade razoável de troca de mensagens para distribuição da informação de localização, mas entretando nenhum servidor central ou estático é necessário.

2.2.3. Sistema de Posicionamento GPS-Free

O sistema de posicionamento é usado quando um terminode precisa obter sua posição geográfica. O GPS pode ser usado para isso, porém em algumas situações ele pode não estar disponível. Isto acontece em lugares onde o sinal GPS é muito fraco, quando o sinal recebe interferência, ou quando um receptor GPS não está integrado ao nó devido ao custo.

O *Self Position Algorithm* (SPA) é um método de posicionamento que não utiliza GPS, e pode ser usado na rede terminode, fornecendo a informação de posição ao terminode. O SPA utiliza medidas de distância entre terminodes para construir o sistema de coordenadas da rede (*network coordinate system* - NCS). O tempo de chegada (TOA – *time of arrival*) é o método utilizado para obter a distância entre dois terminodes. Apesar dos erros de medida de distância e da mobilidade dos terminodes, o SPA fornece informação de localização com precisão suficiente para sustentar as funções de rede básicas. Por simplicidade, o SPA será aqui apresentado apenas para duas dimensões, mas pode ser estendido para três dimensões. O SPA é realizado em cada terminode e consiste das seguintes etapas [16]:

- O terminode mede sua distância até seus vizinhos e envia essa informação para todos eles.
- Cada terminode constrói seu sistema de coordenadas locais (*local coordinate system* - LCS) e calcula a posição de seus vizinhos neste sistema.

- O terminode calcula o fator de densidade em uma vizinhança de n saltos, ou seja, o número de terminodes dentro da vizinhança de n saltos.
- O terminode com o maior fator de densidade forma um grupo de referencia de localização, e computa o centro e a direção do NCS.
- Todos os terminodes computam sua posição no NCS.

Todo terminode i define seu LCS_i escolhendo dois terminodes vizinhos p e q , de maneira que a distância d_{pq} entre p e q seja conhecida, e que os terminodes i , p e q não estejam alinhados. O terminode i se torna o centro do LCS_i tendo coordenadas $(0,0)$. A direção do sistema de coordenadas associado ao LCS_i é definida posicionando o terminode p sobre a parte positiva do eixo x e o terminode q no primeiro quadrante [16]. Esse procedimento é ilustrado na figura 4.

Todo terminode i calcula a posição de seus vizinhos no seu LCS_i . Se um terminode j é vizinho de i , p e q , ou seja, as distâncias d_{ij} , d_{qj} , d_{pj} são conhecidas, sua posição é computada por triangulação. Se k não é vizinho de p e q , i calcula a posição de k em LCS_i , escolhendo dois vizinhos de k que já tenham suas posições conhecidas em LCS_i [16].

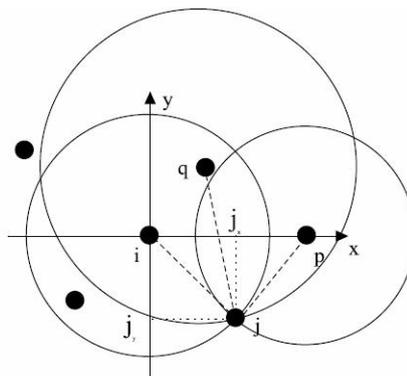


Figura 4 – Sistema de coordenadas local do terminode i

Após cada terminode construir o seu LCS, eles então calculam o fator de densidade. A vizinhança do terminode com o maior fator de densidade, será o grupo de referencia de localização (*location reference group* - LRG) da rede. O centro do NCS será computado como sendo o centro geométrico dos terminodes

contidos no LRG. A direção do NCS será obtida pela média da direção dos LCS do terminodes pertencentes a LRG. Essa informação é propagada para os terminodes na rede, e eles computam sua posição no NCS. O uso do grupo de referência permite obter um sistema de coordenadas mais estável, pois o seu centro e a direção de seus eixos mudarão com menos frequência, já que são calculados a partir da média de medidas efetuadas em vários nós [16]. A figura 5 ilustra um exemplo de grupo de referência de localização.

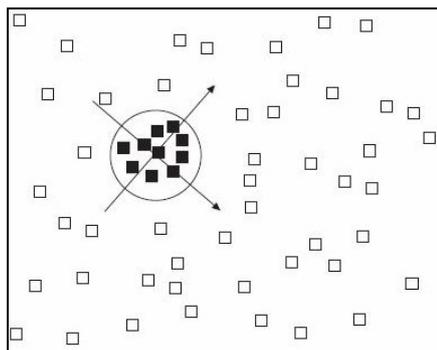


Figura 5 - grupo de referência de localização

A precisão das posições obtidas pelo SPA depende fortemente das medidas de distância e da velocidade de movimento dos terminodes. Velocidades altas para os terminodes introduzem erros adicionais e as posições dos terminodes precisam ser constantemente recalculadas. Posicionamento impreciso tem impacto negativo no algoritmo de encaminhamento de pacotes na rede terminode.

2.2.4. Sistema de incentivo para cooperação

Na rede terminode todos os serviços de rede são fornecidos pelos próprios terminodes. Contudo, fornecer esse serviço não é de interesse do terminode, ou de seu usuário, porque consumirá energia, reduzindo o tempo de vida de sua bateria, sem obter nenhuma vantagem. Por isso, um terminode tende a ser egoísta, ou seja, ele utiliza serviços fornecidos por outros terminodes, mas não deseja fornecer serviços para o restante do terminodes pertencentes a rede. Esse comportamento pode comprometer o funcionamento da rede inteira. Logo, é necessário um mecanismo para encorajar os terminodes a oferecerem serviços para os outros terminodes [2].

No projeto terminode, o incentivo para cooperação é baseado na criação de uma moeda virtual chamada de *nuglets*, e de mecanismos para trocar os *nuglets* por serviços. Assume-se que o hardware contém um estoque inicial de *nuglets*. A idéia natural é que os terminodes precisam pagar aos terminodes que lhe fornecem serviços. Isto faz o *nuglet* indispensável para utilizar a rede. Assim, cada usuário estará interessado em aumentar seu estoque de *nuglet*. Sendo que a única forma de fazê-lo é fornecer serviços para a comunidade de terminodes [17].

O pagamento do serviço de roteamento dos pacotes pode ser feito pela fonte ou pelo destino do pacote. Quando o nó que origina o pacote paga pelo roteamento, o serviço de cobrança é distribuído entre os terminodes que estão encaminhando o pacote da seguinte maneira. Quando for enviar um pacote, a fonte carrega o pacote com um número suficiente de *nuglets* para atingir o destino. Cada terminode que participa do encaminhamento do pacote pela rede adquire alguns *nuglets* do pacote e aumenta seu estoque. O número exato de *nuglets* retirados do pacote pode ser variável. Se um terminode não tem *nuglets* suficientes para encaminhar o pacote, ele é descartado. A principal vantagem desse modelo além do estímulo para cooperação entre os terminodes, é que previne os usuários de enviarem dados inúteis e sobrecarregar a rede. A principal desvantagem está na dificuldade de estimar o número de *nuglets* necessário para chegar até o destino [17].

Este último problema é resolvido no modelo onde o terminode de destino do pacote paga pela função de roteamento. Neste modelo o pacote não carrega *nuglets*, mas é trocado por *nuglets* pelo terminode que está encaminhando o pacote. Cada terminode compra o pacote do salto anterior por alguns *nuglets* e vende o pacote para o próximo salto (ou para o destino) por mais *nuglets* que ele pagou. Dessa maneira, cada terminode que encaminhou o pacote aumenta seu número de *nuglets*, e o custo total do roteamento é coberto pelo destino do pacote. Além de liberar a origem da necessidade de estimar o número de *nuglets* necessário para entregar o pacote, outra vantagem desse modelo é que ele é mais adequado para comunicação *multicast*. Uma desvantagem entretanto é que esse modelo não impede um terminode de inundar a rede de pacotes [17].

Na figura 6, apresenta-se um exemplo onde a fonte paga pelo serviço de roteamento. Na figura a, os números próximos aos terminodes representam o estoque inicial de *nuglets* de cada terminode. O terminode A está enviando um pacote para o terminode G. Para isso, o pacote leva 30 *nuglets* retirados de A. O terminode C retira 5 e o terminode E retira 12 *nuglets* para rotear o pacote.

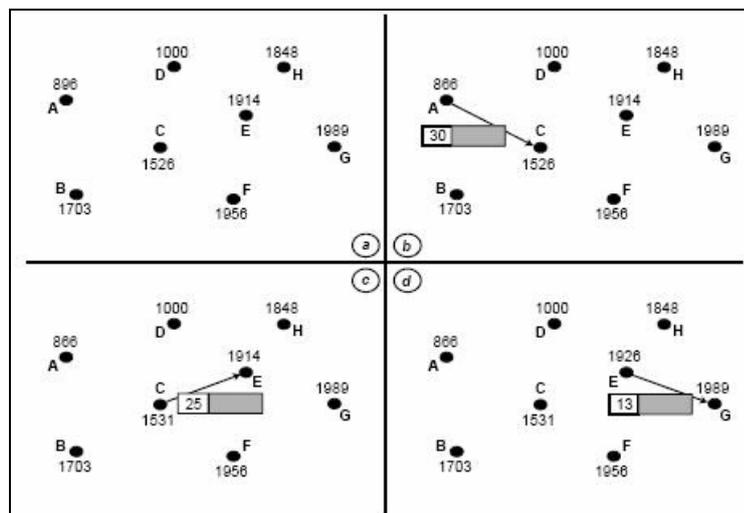


Figura 6 – exemplo de troca de *nuglets* por serviço

Obviamente os modelos aqui descritos precisam ser seguros e protegidos contra vários tipos de ataques. O problema básico a ser resolvido relaciona-se com a falsificação de *nuglets* e a justiça na troca de pacotes por *nuglets*. O desafio é encontrar o equilíbrio entre a robustez da solução e sua eficiência, ou seja, encaminhar um pacote não pode exigir protocolos complexos de criptografia e grande esforço computacional, porque esse custo pode exceder o valor do serviço [2].

2.2.5. Protocolo de roteamento terminode

O protocolo de roteamento Terminode procura manter os benefícios da escalabilidade dos protocolos de roteamento baseados em informação de posição, visando as questões da topologia irregular e mobilidade dos nós.

Conforme mencionado anteriormente, o roteamento dos pacotes é feito por uma combinação de dois protocolos de roteamento. O primeiro é baseado em

localização (denominado de *terminode remote routing*, denotado por TRR), usado quando o destino está distante da fonte do pacote, e o segundo é um protocolo de roteamento do tipo *link state*, o *terminode local routing* (TLR), usado quando o destino está próximo.

2.2.5.1. Terminode local routing (TLR)

Os terminodes utilizam o TLR para construir suas tabelas de roteamento local, e para encaminhar pacotes para destinos na sua vizinhança. Assim, o TLR é usado para rotear pacotes para terminodes que estiverem presentes na tabela de roteamento local da fonte ou de um terminode intermediário. O TLR contém uma componente de roteamento do tipo *link-state*, limitada no escopo dos nós que estão a dois saltos de distância (este número pode variar), utilizando somente o endereço independente da localização EUI para realizar o roteamento [6].

A área de um terminode alcançável pelo TLR inclui todos os terminodes que possuem distância mínima em saltos menor ou igual que o raio local. O raio local é uma medida, em número de saltos, da área que é coberta pelo TLR. Esse valor é um parâmetro da implementação do protocolo de roteamento e aqui será usado o valor de dois saltos [6].

A tabela de roteamento local do TLR é construída de maneira pró-ativa, contudo o endereço EUI e a localização (LDA) dos seus vizinhos de um salto e o EUI dos seus vizinhos de dois saltos. Cada terminode envia periodicamente para seus vizinhos de um salto mensagens do tipo HELLO, contendo seu EUI e seu LDA, além da lista dos EUIs de seus vizinhos de um salto. O endereço EUI é usado para encaminhar o pacote por TLR, e a posição LDA é usada para encaminhar por TRR [6]. Cada entrada na tabela de roteamento possui um tempo de expiração associado. Se o nó não escuta seu vizinho imediato por algum tempo, ele o remove da sua tabela de roteamento, assim como os vizinhos dois saltos que são por ele alcançáveis [4].

2.2.5.2. Terminode remote routing (TRR)

O TRR permite enviar pacotes para destinos que não podem ser alcançados diretamente pelo TLR, ou seja, destinos que estão a uma distância maior que dois saltos do terminode que está enviando o pacote. O TRR usa a informação da localização geográfica do terminode de destino para rotear o pacote e é o ponto principal para a escalabilidade do protocolo. TRR utiliza uma combinação de caminho direto, modo perímetro e âncoras, para calcular o melhor caminho até o terminode de destino [6].

Em complemento ao roteamento utilizando a informação de posição do terminode de destino, quando um pacote chega a dois saltos de distância do destino, o pacote passa a ser roteado utilizando o TLR. Isso ocorre porque a essa distância o terminode de destino deve estar presente na tabela de roteamento local de algum terminode presente nessa área. Então, é feita uma abordagem do tipo *link state*, que não usa a informação de localização e se baseia apenas no endereço permanente. Isso evita erros de roteamento devido à imprecisão da informação de localização. O TRR usa uma combinação de caminhos diretos, modo perímetro e âncoras, conforme será descrito mais adiante.

O caminho direto consiste em aproximar o caminho até o destino por uma linha reta. Assumindo que a fonte S conhece a localização aproximada do destino D, S envia o pacote para o terminode vizinho que o leve para mais perto da suposta localização de D. Esse procedimento é repetido pelo nós intermediários, enquanto ele funcionar, de modo a aproximar cada vez mais o pacote da localização do destino [4].

Normalmente, o uso do caminho direto seria a melhor opção para o envio do pacote, porém existem casos onde o caminho direto é inadequado. O pacote pode ficar retido em algum nó que não possui um vizinho mais próximo do destino do que ele mesmo. Nesse caso o TRR utiliza o modo perímetro para contornar este “buraco” na topologia, ou seja, contornar obstáculos. Para realizar o roteamento em modo perímetro, a rede é representada por um grafo planar, permitindo

encontrar caminhos ao redor desses obstáculos, conforme será descrito mais adiante. O TRR continua nesse modo até encontrar um nó que reduz a distância até o destino, a partir desse ponto o modo em caminho direto volta a ser usado. O uso do modo perímetro deve ser evitado, pois pode resultar em caminhos longos sub-ótimos. Além disso, pode causar *loops* de roteamento freqüentes em redes ad hoc móveis. Esse método de envio de pacotes que utiliza uma combinação de caminho direto e modo perímetro é chamado de *Geodesic Packet Forwarding* (GPF), e será explicado em seqüência [4]. A figura 7 mostra um exemplo do GPF, onde as linhas pontilhadas indicam o uso do modo perímetro. As regiões sombreadas mostram as áreas onde o caminho direto não pode ser utilizado.

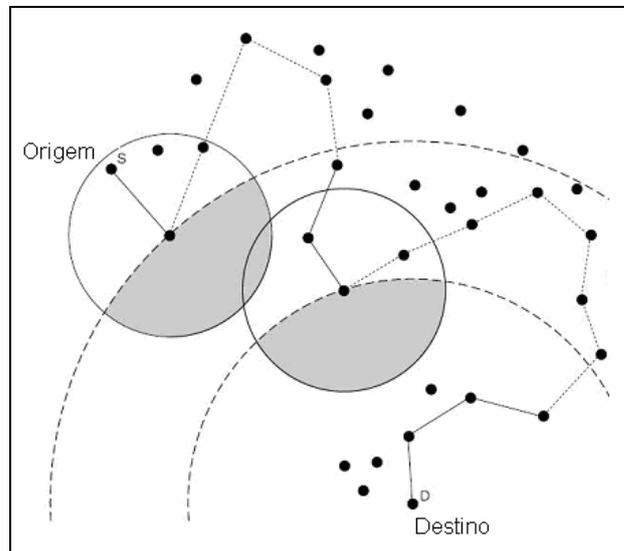


Figura 7 – exemplo do roteamento utilizando GPF

Com o objetivo de evitar o modo perímetro, foi introduzido o conceito de âncoras, que são localizações imaginárias usadas para auxiliar no roteamento. Âncoras são pontos geográficos e não estão associadas a nenhum terminode específico [4]. As âncoras indicam um caminho aproximado que o pacote deve percorrer até chegar ao destino. Esse método de enviar pacotes por um caminho com âncoras é chamado *Anchored Geodesic Packet Forwarding* (AGPF) [6].

O uso de âncoras bem escolhidas reduz enormemente o número de saltos executados pelo pacote comparado ao modo perímetro, por duas razões. Primeiro, pode levar a uma melhor estratégia de roteamento. Segundo, mesmo quando o modo perímetro e os caminhos com âncoras seguem direções similares, o caminho

utilizando modo perímetro tende a ser menos retilíneo e usar mais saltos, porque ele está sempre tentando desviar de áreas de problema. Porém, quando um caminho com âncoras não está disponível, o *Geodesic Packet Forwarding* é utilizado [4].

2.2.5.2.1. Geodesic Packet Forwarding (GPF)

O GPF é um método simples para enviar dados na direção de um ponto geográfico. Este ponto pode ser uma âncora ou a localização do terminode destino. A fonte S usa o GPF para enviar dados para um destino remoto D da seguinte maneira: S adquire a LDA de D de alguma maneira, e coloca a nos pacotes. Com isso, um nó encaminhando o pacote pode fazer uma escolha ótima do melhor caminho de maneira local. Se um nó conhece a posição de seus vizinhos, a escolha local ótima do próximo salto é o vizinho que está geograficamente mais próximo do nó destino. Este método se repete em saltos sucessivamente mais próximos, até alcançar o destino [18]. Esse método de repasse de dados é dito guloso porque procura escolher sempre o nó mais próximo do destino, sem considerar os próximos saltos da comunicação [19].

Cada terminode intermediário X quer receber o pacote, verifica se D pode ser alcançado pelo TLR, e em caso afirmativo, o pacote passa a ser roteado utilizando TLR. Caso contrário, X envia o pacote para seu vizinho mais próximo do destino D [6]. Essa é a forma de envio por caminho direto citada anteriormente.

Normalmente nessa forma simplificada, o GPF tende a não funcionar. O problema ocorre em topologias onde a rota para o destino exige que o pacote se mova temporariamente para uma distância geográfica mais longe destino, do que o nó atual [18]. O pacote pode ficar retido em terminode que não possui vizinho mais próximo do destino que ele mesmo. Uma solução para isso é utilizar o método de passagem por grafos planares, onde o pacote é roteado ao redor do perímetro da região onde não existem terminodes mais próximos da origem [6].

A figura 8 apresenta um exemplo onde a fonte S não possui nenhum vizinho mais próximo da origem que ela mesma. A área sombreada da figura ilustra a região que se encontra dentro do raio de cobertura de S, e que se localiza mais próximo do destino D do que o nó S. Como pode ser visto, S não possui nenhum vizinho dentro dessa área. Utilizando somente o método do caminho direto, o pacote ficaria retido em S apesar de existirem rotas para alcançar D. Essa situação é inaceitável, e o modo perímetro é então utilizado para contorná-la .

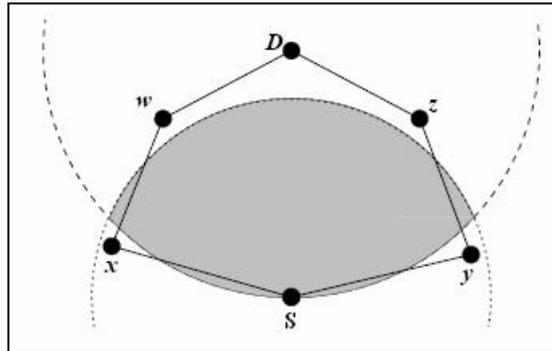


Figura 8 - Exemplo de topologia onde o método simplificado GPF não funciona

O modo perímetro é realizado utilizando um grafo planar da rede e a regra da mão direita para percorrer o grafo. A regra da mão direita define que ao chegar em um nó x vindo do nó y , a próxima borda a ser percorrida será a borda subsequente de (x,y) , no sentido anti-horário em relação a x . A figura 9 mostra um exemplo do emprego da regra da mão direita, onde o caminho percorrido no grafo é formado por $y \rightarrow x \rightarrow z \rightarrow y$. Esta propriedade pode ser usada para contornar regiões problemáticas da topologia da rede. Porém, para este método funcionar, o grafo não pode possuir bordas que se cruzem e por isso deve ser usado um grafo planar da rede [18].

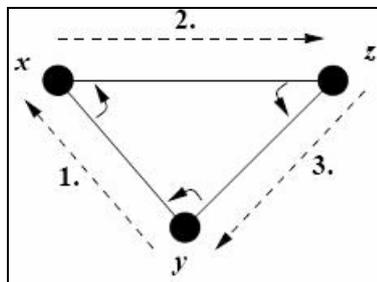


Figura 9 – regra da mão direita

Um grafo é planar se não existe duas bordas que se cruzem. Um conjunto de nós (com raios de transmissão circular idênticos de valor r) pode ser visto como

um grafo. Cada nó é um vértice e a borda (n,m) existe entre os nós m e n se a distância entre eles for menor que r . Grafos em que a existência das bordas é definida pela distância são chamados de grafos unitários [18].

O *Relative Neighborhood Graph (RNG)* e o *Gabriel Graph (GG)* são dois grafos planares conhecidos, sendo que o RNG é um subgrupo dos grafos GG. O RNG é abordado aqui como exemplo. Um grafo unitário pode ser reduzido para um grafo RNG, removendo as bordas que não fazem parte do RNG, obtendo-se dessa forma um grafo planar da rede. A figura 10 ilustra a construção do grafo RNG. Uma borda (u,v) existirá entre u e v , se a região sombreada não possuir nenhum outro nó, ou seja, se a distância $d(u,v) \leq \max [d(u,w), d(v,w)]$, para qualquer w diferente de u e v . Este algoritmo pode ser localmente executado por cada nó, conhecendo apenas seus nós vizinhos [18].

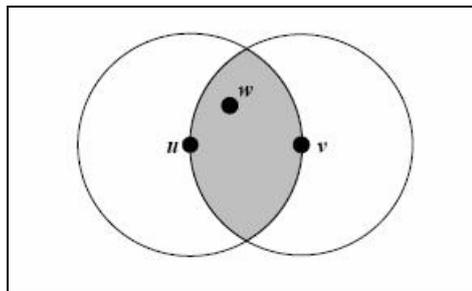


Figura 10 – construção do grafo RNG

No método GPF, quando um nó intermediário receber um pacote, ele procura um vizinho mais próximo do destino para enviar o pacote. Caso esse vizinho não exista, o pacote é enviado utilizando o modo perímetro. A figura 11 mostra um exemplo do roteamento utilizando este modo. Quando este modo é usado de S para D , o algoritmo envia o pacote para faces que são cortadas pela linha SD , e sejam cada vez mais próximas de D . No exemplo, isso ocorre nas faces 1, 2, 3 e 4 respectivamente. Em cada face, a regra da mão direita é utilizada para atingir uma borda que cruzem a linha SD .

A localização LDA do nó onde o pacote passou a ser encaminhado pelo modo perímetro é armazenada. Quando o pacote chegar em um nó em que a distância até o destino seja menor que o LDA armazenado, o caminho direto volta a ser utilizado.

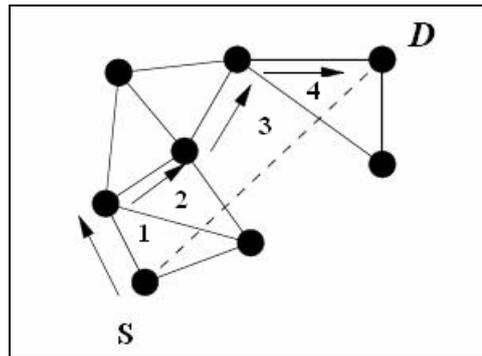


Figura 11 – encaminhando um pacote pelo modo perímetro

2.2.5.2.2. Anchored Geodesic Packet Forwarding (AGPF)

O principal elemento do AGPF é o caminho com âncoras. As âncoras são computadas pelos nós fontes utilizando um dos dois seguintes métodos: *Friend Assisted Path Discovery* (FADP) ou *Geographical Map-Based Path Discovery* (GMPD).

Uma lista de âncoras é escrita pela fonte no cabeçalho do pacote. Essa lista forma o caminho com âncoras, que servirá como uma aproximação do caminho a ser seguido pelo pacote. Na fonte o pacote é enviado na direção da primeira âncora (AP1) contida na lista, utilizando o método GPF, ou seja, a fonte envia o pacote para o vizinho que possui menor distância até AP1. Quando o pacote é recebido por algum nó intermediário, ele verifica se AP1 está geograficamente dentro de seu raio de transmissão. Se estiver, o nó intermediário remove AP1 do caminho com âncoras e envia o pacote para a posição da próxima âncora (AP2). Esse procedimento é repetido até o destino, sendo que localização final do destino funciona como se fosse última âncora [6].

A figura 12 mostra um exemplo do uso do AGPF. No exemplo, S não possui um caminho direto para o destino D, então o roteamento é efetuado utilizando-se o AGPF. S envia o pacote em direção a primeira âncora AP1. Ao chegar ao nó A, este verifica que AP1 está dentro de seu raio de transmissão. O nó A atualiza o próximo destino do pacote, que passa a ser então enviado na direção de AP2. O nó B verifica que AP2 está dentro do seu raio de transmissão e

envia o pacote para o destino final D. Para rotear o pacote entre dois pontos geográficos do caminho com âncora é usado o GPF.

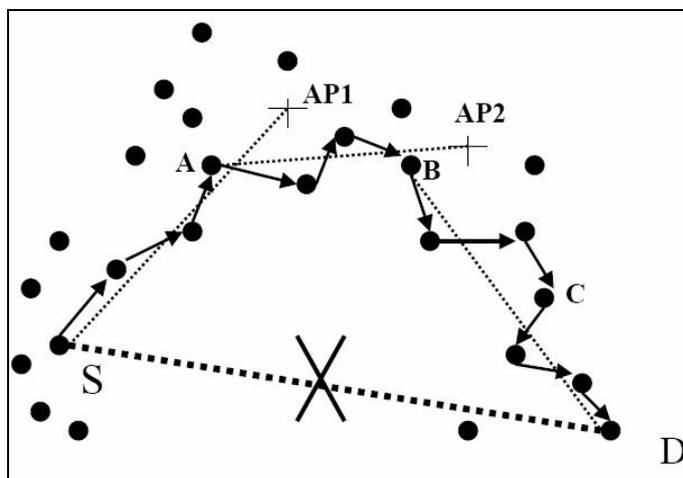


Figura 12 – Roteamento utilizando AGPF

2.2.5.2.3. Métodos para cálculo de âncoras

O roteamento adequado depende da qualidade da geração de caminhos com âncoras. Dois métodos são propostos para o cálculo de âncoras, sempre implementados pela fonte.

- *Friend Assisted Path Discovery (FADP)*. Um terminode mantém uma lista de terminodes, que ele chama de amigos, para os quais ele mantém um ou vários bons caminhos. Um terminode contata seus amigos para encontrar um caminho com âncoras para um destino [6]. Este método assume que alguns nós (*FADP responders*) estão aptos para fornecer assistência para outros, normalmente porque eles possuem uma visão estável da densidade de terminodes da rede. Os *FADP responders* ajudam a encontrar âncoras, mas não são usados no caminho dos dados [4].
- *Geographical Map-Based Path Discovery (GMPD)*. Assume que os mapas de densidade da rede estão disponíveis para todos os nós fontes. Isto é para uma rede ad hoc onde todos os nós são

individualmente móveis, mas a densidade de nós ainda pode ser prevista. O GMPD tem o *overhead* da distribuição do mapa [4].

Além disso, através de sua configuração local um nó determina se:

- Ele possui o mapa de densidade, estando apto a calcular âncoras com GMPD.
- Ele é um FAPD responder, capaz de fornecer assistência para outros nós.

2.2.5.3. Restricted local flooding

Existem situações onde a precisão do sistema de localização é baixa e o TLR sozinho não é suficiente para lidar com essa situação. Quando a informação de localização do destino é imprecisa, pode ser usado um método chamado *Restricted Local Flooding* (RLF). Esse método envia de quatro a seis cópias do pacote para a região em que provavelmente o destino está localizado, aumentando assim a possibilidade de encontrá-lo. O RLF recupera-se das imprecisões da localização quando o destino está dentro de algumas faixas de transmissão de onde o RLF foi iniciado, ou seja, próximo a localização de destino contida no pacote. Em grandes redes, enviar cópias sempre causa um overhead consideravelmente menor que o *flooding*. O RLF é usado para dois tipos de descobertas: 1) procurar em uma área limitada por algum nó ou por um tipo de nó (FAPD). 2) estabelecer relações de longa distância [4].

O RLF é uma forma de *flooding* local e restrito, não usa broadcasting como o *flooding* comum e os pacotes são excluídos depois de um certo número de saltos se não encontrarem o destino. A figura 13 ilustra o funcionamento do RLF para um terminode X. O terminode X envia cada cópia na direção de uma dos seis pontos geográficos ao seu redor. Sejam $\{X_i\}$ essas localizações geográficas, para i igual a 1 até 6. No i -ésimo pacote o LDA de destino do pacote é selecionado como X_i . Contudo, os campos de destino EUI não são alterados. Os pontos de X_1 a X_6

estão a mesma distância de X, que é igual a duas vezes o raio de transmissão de X. Então a região com raio igual a duas vezes o raio de transmissão de X será coberta, e é provável que se o destino estiver em seu interior ele receba uma cópia do pacote [4].

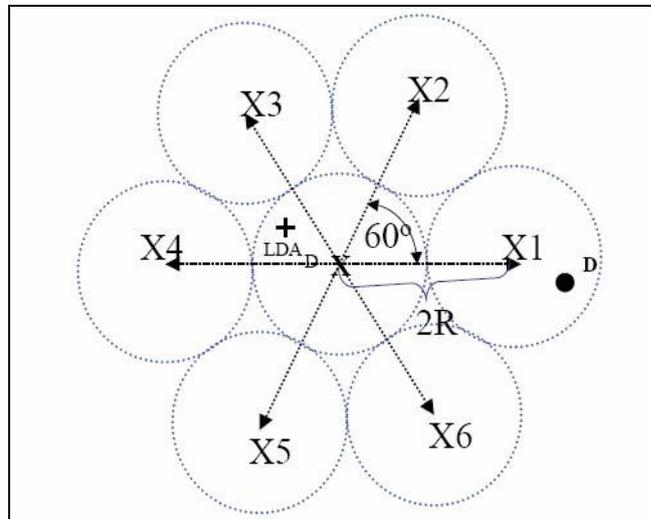


Figura 13 – cálculo de posição para o uso do RLF

O campo TTL em cada cópia do pacote é selecionado para *term_rlf* (uma constante de projeto, escolhida com o valor 4). Desta maneira, o tempo de vida do pacote é restringido a *term_rlf* saltos. Pacotes com o bit RLF igual a 1 são encaminhados para as localizações geográficas X_i . Há três possíveis situações para cada cópia do pacote [4]:

- O pacote é entregue ao destino por algum nó intermediário que encontra D na sua tabela de roteamento TLR.
- O pacote é excluído devido à expiração do TTL
- Algum nó intermediário N encontra X_i na sua área de transmissão, mas o destino não está na sua tabela TLR. Então N exclui o pacote.

2.2.5.4.

Detalhes do funcionamento protocolo de roteamento

Quando um nó é ativado, ele inicia o método de roteamento local TLR enviando mensagens de HELLO com seu endereço (EUI) e sua localização (LDA). Ele também inicia a escuta de mensagens HELLO enviada por outro nós,

que é utilizada para a construção das tabelas de roteamento local. O nó periodicamente transmite suas mensagens HELLO, que além do seu endereço e localização, contém o endereço (não a localização) de seus nós vizinhos [4]. Dessa forma, cada terminode monta sua tabela TLR. A tabela abaixo mostra os principais campos de controle do protocolo de roteamento terminodes.

Tabela 1 – alguns campos do pacote usados pelo protocolo de roteamento terminode

Campo do cabeçalho	Descrição
EUI do destino	Endereço permanente do nó de destino, usado pelo TLR
LDA do destino	Informação de localização do destino, usado pelo TRR
Use TLR	Se igual a 1, indica o uso do TLR. Caso, contrário o protocolo usado é o TRR.
Use RLF	Indica o uso do RLF, quando o pacote se aproximar de uma localização.
RLF	Se igual a 1, indica que o RLF já está sendo usado para rotar o pacote.
Next location	Indica o próximo ponto geográfico para o qual o pacote é enviado.
Lista de âncoras	Lista de todos os pontos geográficos que formam o caminho com âncoras.

2.2.5.4.1. Remessa dos pacotes

Antes que um terminode fonte S envie um pacote para um destino D, ele primeiro deve verificar se o destino D está na tabela TLR. Em caso afirmativo, o bit “*use TLR*” no cabeçalho do pacote recebe o valor 1. Esse bit serve para indicar aos nós intermediários qual protocolo de roteamento deverá ser usado: TLR (*use TLR=1*) ou TRR (*use TLR=0*). A partir do momento que o TLR for selecionado para encaminhar o pacote, o TRR não poderá voltar a ser utilizado. Nesse caso, o próximo salto é determinado pela tabela de roteamento TLR. Se D está a dois saltos de distância, é escolhido o vizinho com a entrada mais recente na tabela. Se o bit “*use TLR*” do pacote está selecionado em 1, o pacote deve ser entregue diretamente para o destino, que deve ser um de seus vizinhos de um salto. Caso

contrário o pacote é excluído, assegurando que o TLR é livre de *loops* de roteamento [4].

No caso da fonte S não possuir o destino D em sua tabela de roteamento local, o protocolo usado é o TRR. O nó S deve determinar se uma localização válida de D é conhecida, o que provavelmente não acontece imediatamente após ele ser ligado. S então usa um serviço de descoberta de localização e o pacote é enfileirado até S conseguir esta informação. Após obter a informação de localização do terminode D (LDA_D), S coloca essa informação no cabeçalho do pacote e o envia usando TRR [4].

A localização de D é armazenada e atualizada pelo protocolo de rastreamento. A informação de localização mantida pelo nó possui dois *timers* que controlam a sua validade. Quando o primeiro expira, a entrada é válida, mas antiga e portanto imprecisa. Então, S ativa o bit “*use RLF*” do pacote, informando aos nós intermediários que o método RLF deve ser utilizado quando este estiver próximo da localização suposta de D. Isso aumenta a probabilidade do pacote chegar ao destino, mesmo com informação de localização imprecisa. Quando o segundo timer expira, a entrada é considerada ultrapassada e é removida [4].

Para utilizar caminhos com âncoras, a fonte deve calculá-las para cada destino de interesse. Se o nó fonte possuir o mapa de densidade da rede, ele calcula o GMP e caso contrário utiliza o FAPD, armazenando assim o caminho com âncoras calculado. Quando essa informação se torna antiga, um novo caminho é procurado. Quando considerado ultrapassado, esse caminho é removido. Uma fonte pode remover prematuramente uma âncora se avaliar que o desempenho do caminho é ruim, por exemplo, quando o destino reporta baixa taxa de entrega. Para enviar pacotes com âncoras, o nó fonte adiciona ao cabeçalho do pacote as âncoras utilizadas para formar o caminho. O ponteiro “*Next location*” no cabeçalho do pacote recebe o valor da primeira âncora, e o pacote é enviado na direção dessa primeira âncora do caminho até o destino [4].

Quando um nó intermediário recebe um pacote, ele verifica se o endereço de destino (EUI) é o dele e se for, o pacote é entregue internamente, senão ele será

encaminhado para o próximo destino. Primeiro o nó verifica se o destino está na sua tabela TLR ou se o bit “use TLR” do pacote está em 1. Em caso afirmativo, a operação é realizada como se esse nó fosse a fonte do pacote. Se o nó determina que o TLR pode ser usado, ele coloca em 1 o bit “use TLR”. Com isso o TRR não pode voltar a ser usado, evitando *loops* devido à mobilidade do nó [4].

Se o nó intermediário não possuir o destino em sua tabela TLR, ele pode decidir quando o TRR deve ser abortado. Este procedimento pode ser necessário se a precisão da localização não for suficiente ou se o pacote for atrasado devido a congestionamento ou caminho ruim. Nesses casos o bit “*use TLR*” pode nunca ser selecionado em 1, e o pacote pode ficar circulando ao redor de LDA_D . Neste caso, o pacote é encaminhado por nós que estão próximos a LDA_D , mas nunca alcançando o destino porque D se moveu consideravelmente e nenhum nó na vizinhança o contém em sua tabela TLR. Depois de alguns saltos sendo roteado em círculo, o pacote é excluído por expirar seu tempo de vida (TTL). Para evitar essa situação o TRR deve ser terminado e o pacote excluído nas seguintes condições [4]:

- A distância entre a localização do terminode atual e a informação de localização de destino contida no pacote tem que ser menor que raio de transmissão. Isso porque nessa condição, se o destino não está na tabela TLR do terminode significa que LDA_D não é precisa.
- O bit *RLF* não pode estar em 1. O bit “*RLF*” no cabeçalho do pacote previne o uso do RLF em cascata.

Para que o terminode intermediário termine o TRR, ele depende do valor do bit “*use RLF*” no cabeçalho do pacote. Este valor, que normalmente está em zero, se estiver em 1, significa que a fonte suspeita que a localização do destino não é precisa. Duas ações podem ser consideradas pelo nó intermediário [4]:

- Se “*use RLF*” = 0, o nó faz o valor TTL igual ao mínimo valor de ($term_trr, TTL$). Isto limita o efeito do erro de posição, fazendo com que o número máximo de saltos que o pacote irá circular seja igual ao valor de $term_trr$ ($term_trr$ é uma constante de projeto, e foi escolhida com o valor 3).

- Se “*use RLF*” = 1, o *restricted local flooding* (RLF) é usado. Ele consiste em enviar seis cópias do pacote em direções ao redor do nó intermediário (X) que está enviando o pacote. Dessa forma o pacote é enviado numa área ao redor de X, onde é esperado encontrar o nó de destino. Todos os pacotes têm o mesmo endereço de destino, mas com diferentes informações de localização como destino.

Para continuar encaminhando o pacote usando TRR, o nó verifica se o ponteiro “*next location*” contido no pacote, que indica a âncora para onde o pacote está sendo roteado, está dentro de sua área de transmissão. Se estiver, a próxima âncora contida na lista é selecionada como “*next location*”. Caso seja a última, ele passa a apontar para o destino final. Então o nó encaminha o pacote para a próxima localização atualizada.

2.2.5.4.2.

Descoberta de caminhos com âncoras utilizando o FAPD

O *Friend Assisted Path Discovery* (FAPD) utiliza alguns terminodes, chamados *FAPD responders*, que fornecem assistência para que outros terminodes descubram caminhos com âncoras. Alguns dos terminodes da rede são configurados para atuar como *FAPD responders*. Estes terminodes mantêm uma conexão para alguns de outros *FAPD responders* da rede, que serão seus terminodes amigos. Um terminode inicia a descoberta de um caminho com âncoras enviando uma requisição para algum *FAPD responder*. Se o *FAPD responder* que receber a requisição não possuir um caminho conhecido para o destino, ele contata seus *FAPD responders* amigos para encontrá-lo. Vários *FAPD responders* podem participar em uma descoberta de um caminho com âncoras [4].

Cada *FAPD responder* deve manter uma lista de terminodes amigos, sendo essa operação iniciada quando o terminode é ligado, e a seguir de forma periódica. Para o sucesso dessa operação, um *FAPD responder* usa o RLF, mas com apenas quatro cópias do pacote. Nesse caso, o RLF é usado com o objetivo de estabelecer algumas relações de amizade de longa distância. Cada um dos quatro pacotes

contém uma mensagem *get_friend_request*, tem o bit RLF igual a 1, o EUI de destino igual ao valor de broadcast (pois o terminode não sabe a identidade dos FAPD *responders*), o campo TTL igual a 6, e o LDA de destino igual a um dos quatro campos geográficos selecionados. Os quatro pontos são selecionados em direções ortogonais, a uma distância de 4 raios de transmissão do terminode que está enviando o pacote. Nenhuma âncora é usada para enviar esses pacotes. Quando um FAPD *responder* na direção FP_i recebe a mensagem de *friends_request*, ele não repassa a mensagem. Ele envia como resposta uma mensagem *friends_reply*, contendo sua uma lista de amigos e seu endereço. Ao receber a mensagem de *friends_reply*, o FAPD *responder* combina a informação recebida com sua lista atual de amigos [4].

Um nó precisa saber como selecionar um número específico de amigos a partir de uma lista de amigos em potencial. A chave para gerar o fenômeno de *small-world* [20] é a presença de uma pequena fração de conexões de longa distância, conectando a partes distantes do gráfico, enquanto a maioria das conexões permanece local, contribuindo assim para a propriedade do grafo da rede ser altamente aglomerado. A estratégia é considerar a posição geográfica dos nós quando construir a conexão com os amigos.

Após o FAPD *responder* selecionar seus amigos, cada amigo é associado com as seguintes informações: identidade do amigo, localização e caminho até o amigo. As localizações dos amigos são rastreadas e o caminho até eles avaliado. Um amigo é declarado desatualizado se o nó não escuta aquele amigo por algum tempo. Se o número de amigos é considerado pequeno, um FAPD *responder* pode novamente iniciar o procedimento de descoberta de amigos.

2.2.5.4.3.

Emprego de caminhos com âncoras

A fonte S inicia uma descoberta de um caminho com âncoras para o destino D, obtendo a sua localização (LDA_D). Os FAPD *responders* são responsáveis por encontrar as âncoras e o terminode de destino finaliza essa operação.

Se o terminode S for um FAPD *responder* e tiver uma lista de amigos, ele requisita a assistência de algum amigo para fornecer um caminho com âncoras até o destino D. S então envia um pacote de controle chamado *anchored_path_request* para um amigo, que leve o pacote para mais perto D, se existir algum. S usa um caminho conhecido para enviar o pacote para o terminode amigo. Esse pacote de controle contém um campo *fapd_anchored_path*, que acumulará o caminho até D. Se S possui um caminho com âncoras para o terminode amigo, o pacote é enviado por esse caminho e o campo *fapd_anchored_path* é inicializado com as âncoras desse caminho. S marca a requisição de caminhos com âncoras com um número de seqüência, e faz o campo *tabu_index* do pacote de controle igual a 0. O *tabu_index* é um campo do cabeçalho do pacote e sua função é mostrada na tabela 2. Se o terminode S não possuir amigos que sejam mais próximos de D do que ele mesmo, S inicializa uma busca FAPD em modo tabu [4]. A tabela 2 mostra os campos de controle do pacote para o uso em modo tabu.

Tabela 2 – campos utilizadas na busca em modo tabu

Campo	Descrição
<i>tabu_index</i>	Contador que marca o número de amigos percorridos pelo pacote em modo tabu, na execução atual.
<i>min_dist</i>	Distância geográfica entre o ponto onde o modo tabu foi iniciado e o destino.

Se S não é um FAPD *responder*, ele envia vários pacotes *anchored path request* para a região geográfica ao seu redor. Para isso, S usa o método RLF. Cada uma das quatro cópias do pacote é enviada em modo *anycast* (*anycast* é um valor EUI predefinido, significando qualquer FAPD *responder*), em uma região de até quatro vezes o raio de transmissão ao redor de S, e tem *tabu_index* = 0. Qualquer terminode FAPD *responder* que receber o pacote de requisição de S, e que mantém uma lista de amigos, poderá responder ao pacote. Se vários FAPD *responders* recebem o pacote de S, significa que S pode aprender vários caminhos

com âncoras para D. Alternativamente, se nenhum pacote alcança um FAPD responder, S não obterá nenhum caminho com âncoras para D [4].

Quando um FAPD *responder*, digamos F1, recebe um pacote de solicitação para um caminho com âncoras, ele adiciona sua localização geográfica ao campo *fapd_anchored_path*, e pode realizar uma das seguintes ações [4]:

- Se F1 tem um caminho com âncoras para D, ou para alguma localidade em que a distância para LDA_D seja duas vezes menor que o raio de transmissão, F1 adiciona esse caminho em *fapd_anchored_path* e envia o pacote para D usando TRR com âncoras. Se F1 ainda não possui uma lista de amigos, F1 envia o pacote diretamente para o destino, utilizando TRR sem âncoras.
- Se o caso anterior não for atendido, F1 testa se o campo *tabu_index* é igual a 0, agindo como se ele fosse a fonte da solicitação do caminho. Se o campo *tabu_index* for diferente de zero, a seguinte condição tem que ser atendida: o pacote deve ser enviado para algum amigo (chamado aqui de F2), cuja distância entre F2 e D seja menor que o valor contido no campo *min_dist* do pacote de solicitação. Essa busca em modo tabu é explicada a seguir.

A busca em modo Tabu ocorre quando: 1) a fonte ou um nó FAPD *responder* recebe uma solicitação para descobrir um caminho com âncoras e não possui nenhum amigo que encaminhe essa solicitação para um ponto geográfico mais próximo do destino do que ele mesmo; 2) quando não possui um amigo com distância até o destino com valor menor que o campo *min_dist* contido na solicitação. A idéia é semelhante ao modo perímetro do GPF, onde o pacote precisa se afastar temporariamente do destino. Ao entrar em modo tabu os seguintes passos são realizados [4]:

- Se o campo *tabu_index* da mensagem recebida é 0, o campo *min_dist* recebe o valor da distância do nó até o destino.

- Se o valor do campo *tabu_index* não ultrapassa o valor *max_tabu_index* (constante de projeto do protocolo, com valor igual a 2), T seleciona um amigo disponível F1 a uma distância que não exceda *max_dist* (constante de projeto do protocolo, com valor igual a 5 vezes o raio de transmissão do nó). Nesse caso, o terminode anexa o caminho até F1 ao *fapd_anchored_path* e encaminha a solicitação até F1. Se o terminode não possuir amigos atendendo essas condições o pacote é silenciosamente descartado.
- Se número máximo de saltos em modo tabu já tiver sido atingido (ou seja, *tabu_index* maior que *max_tabu_index*) o nó que recebeu o pacote anexa seu LDA ao *fapd_anchored_path* e envia o pacote diretamente para o destino, usando o GPF.

Quando o pacote atingir um amigo mais próximo do destino que o valor de *min_dist*, o valor de *tabu_index* é zerado. Qualquer fase desse modo é limitada em dois amigos, podendo haver várias fases no modo tabu ao longo do caminho. A figura seguinte mostra o exemplo de uma busca em modo tabu. A fonte S não possui amigos mais próximos do destino D que ela mesma. Então ela envia a solicitação para F1 e em seguida para F2, ambos mais distante de D do que o valor *min_dist*. Porém, F2 possui um amigo F3 que está mais próximo do destino que *min_dist*, e assim o pacote sai do modo tabu [4].

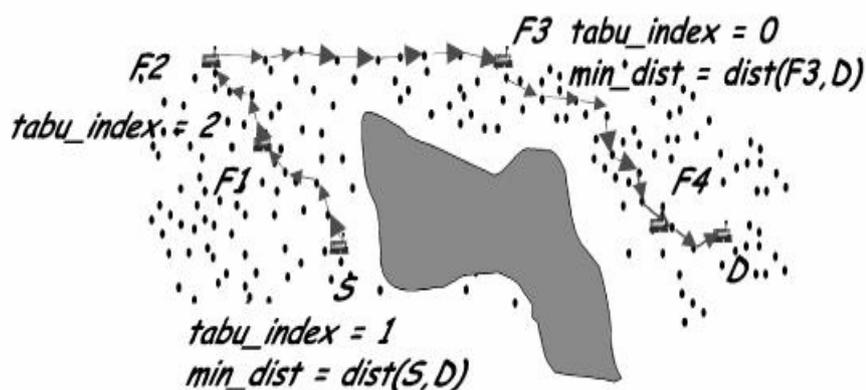


Figura 14 – exemplo da utilização do modo tabu

O nó de destino D será o responsável final por atender a solicitação de um caminho com âncoras até ele. O pacote de solicitação de caminho contém uma

lista acumulada de âncoras de S até D. Este nó executa um método de simplificação de caminho, e retorna para S um pacote de controle *path_reply* que contém o caminho com âncoras de S até D. O pacote de controle de resposta é marcado com o mesmo número de seqüência que o pacote original de solicitação. Se S recebe várias respostas, o número de seqüência é usado para determinar o pacote recebido contendo caminho mais atual. Para enviar o pacote de controle de resposta, D reverte o caminho com âncoras e aplica o TRR nesse caminho. Ao receber de D o pacote de resposta, S armazena o caminho nele contido [4].

A simplificação do caminho consiste em aproximar o caminho com âncoras por um outro que utilize o menor número de âncoras possível. As âncoras (que correspondem à localização dos FAPD *responders* que ajudaram na descoberta do caminho) são acumuladas da fonte até o destino durante o processamento da solicitação de caminho pelos FAPD *responders*. É possível que muitos amigos geograficamente próximos sejam contatados e o caminho resultante possua âncoras muito próximas umas das outras. A meta da simplificação é manter o número de âncoras o menor possível. O destino promove essa simplificação removendo âncoras da lista inicial que estejam localizadas muito próximas de outras âncoras [4].

Um método de manutenção do caminho é usado para decidir quando enviar solicitação para novos caminhos ou quanto tempo o caminho existente deve ser mantido. Depois de enviar uma solicitação de caminho, a fonte aguarda *wait_for_path* segundos para enviar uma nova solicitação. Os caminhos são mantidos por *path_validity* segundos. FAPD *responders* periodicamente atualizam sua lista de amigos com intervalo de tempo de *refresh_friends* segundos.