

UNIVERSIDADE FEDERAL FLUMINENSE

Clayton Reis da Silva

Seleção de gateways em redes em malha sem fio

NITERÓI

2012

UNIVERSIDADE FEDERAL FLUMINENSE

Clayton Reis da Silva

Seleção de gateways em redes em malha sem fio

Dissertação de **Mestrado** *submetida* ao “Programa de Pós-Graduação em Computação” da Universidade Federal Fluminense como requisito parcial para a obtenção do título de Mestre. Área de concentração: Redes e Sistemas Distribuídos e Paralelos.

Orientador:

Prof. Célio Vinicius Neves de Albuquerque

Co-orientador:

Prof. Igor Monteiro Moraes

NITERÓI

2012

Seleção de *gateways* em redes em malha sem fio

Clayton Reis da Silva

Dissertação de Mestrado submetida ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do título de Mestre. Área de concentração: Redes e Sistemas Distribuídos e Paralelos.

Aprovada por:

Prof. Célio Vinicius Neves de Albuquerque / IC-UFF
(Orientador)

Prof. Igor Monteiro Moraes / IC-UFF (Co-Orientador)

Prof. Pedro Braconnot Velloso / IC-UFF

Prof. Mauro Sergio Pereira Fonseca / PUCPR

Niterói, 06 de Fevereiro de 2012

Procure ser um homem de valor, em vez de ser um homem de sucesso.
(Albert Einstein)

Ao meu avô Janes e minha avó Zélia.

Agradecimentos

Em primeiro lugar a Deus, que me criou e me deu forças para poder chegar a este momento tão importante em minha vida.

Meus pais, Jeremias e Conceição, que sempre estiveram comigo nos diversos momentos, bons e ruins, durante esses longos seis anos e meio de faculdade.

Minha irmã Kelli e meu cunhado Franklin, sempre dispostos a me ajudar nos mais diversos problemas que eu enfrentei ao longo desse tempo.

Minha avó Zélia e meu avô Janes, exemplos de vida. Meus tios e tias, primos e primas, que sempre estão comigo.

Meu orientador, professor Célio. Se não fosse por ele, minha caminhada durante a graduação e o mestrado seriam muito mais difíceis. Muito obrigado, professor, por ter depositado suas confianças em mim desde 2007.

Meu co-orientador, professor Igor. Nos conhecemos durante o mestrado, mas você foi de vital importância para que essa tese pudesse chegar a esse ponto para a defesa. Mesmo com algumas divergências, posso dizer que ganhei um grande amigo ao conviver contigo.

Professor Mauro e professor Pedro, obrigado por aceitarem participar da banca, por lerem esse texto e pelas suas sugestões.

Laboratório Midiacom, minha segunda casa nessa jornada. Professores Carrano, Débora e Schara, obrigado pelo apoio e pelas diversas sugestões. Joacir, Rafael Valle, Ladislau, Joel, Felipe Rolim, e todos os outros que riram e trabalharam comigo nesse tempo. Marister, obrigado por tudo, você é sensacional.

Diego Passos, muito obrigado por suas (muitas) sugestões. Tudo que você me ensinou, assim como sua amizade, é de muito valor para mim. Siga em frente nesse doutorado, continue mostrando toda a sua capacidade.

Eduardo e Carlos (Bill), meus grandes amigos durante esses seis anos e meio. Rimos, estudamos, discutimos. Estou saindo da faculdade, mas a nossa amizade permanece para sempre.

Todos que estudaram e se divertiram comigo nesse tempo de faculdade. Daniel Marques, Daniel Souza, Sean, Manoel, André, Matheus, Slobodian, Breno, Carolina, Camila, Rafael Alves, Léo Azevedo, Léo Aramaki, Marcelo... são tantos nomes, que vou parar de listar, senão a Seção de agradecimentos ficará mais longos do que já está.

Meus amigos de infância, adolescência e juventude que me apoiam desde sempre.

Universidade Federal Fluminense e seus professores e funcionários. Apreendi muito neste lugar.

CAPES e TBE (especificamente, à Empresa Amazonense de Transmissão de Energia S.A. e à Empresa Norte de Transmissão de Energia S.A.) pela infraestrutura utilizada neste trabalho.

Resumo

Este trabalho propõe um mecanismo de seleção de *gateways* que provê de forma eficiente o balanceamento de carga entre os múltiplos *gateways* de uma rede em malha sem fio, além de solucionar o problema do uso da técnica NAT em redes em malha sem fio com múltiplos *gateways*. O QuAGS (*Quality-Aware Gateway Selection*) seleciona probabilisticamente o *gateway* a ser usado por uma conexão de um cliente e realiza a ligação entre o nó e os *gateways* através de túneis lógicos. A probabilidade de seleção de um *gateway* é calculada a partir da qualidade do caminho do cliente para esse *gateway*. Portanto, com o QuAGS, *gateways* que não seriam selecionados levando-se em conta somente a métrica de roteamento podem ser usados por conexões de saída e, assim, aumenta-se o uso dos recursos da rede.

O mecanismo proposto é implementado em roteadores sem fio reais e em simulações. Os experimentos práticos mostram que o QuAGS obtém um balanceamento de carga eficiente entre os *gateways* da rede, sem que a vazão seja comprometida. Nas simulações, o QuAGS apresenta vazão superior quando utilizado em nós que estão mais distantes em número de saltos dos *gateways* da rede. Dentre as topologias avaliadas, destacam-se o cenário da rede linear, que mostra que o balanceamento de carga entre *gateways* com o QuAGS resulta em ganho de vazão, e o da rede em grade 8x8, que evidencia a limitação do mecanismo em alguns cenários, nos quais não há ganho em termos de vazão.

Palavras-chave: Múltiplos *Gateways*, Balanceamento de Carga, Redes em Malha Sem Fio.

Abstract

This work proposes a gateway-selection mechanism that efficiently provides load balancing among multiple gateways of a wireless mesh network. It also solves the joint usage of the NAT technique and multiple gateways in wireless mesh networks. QuAGS (Quality-Aware Gateway Selection) probabilistically selects the gateway to be used by each outgoing client connection, and connects nodes and gateways through logical tunnels. The probability of selecting a given gateway is based on the quality of the client-gateway path. Therefore, with QuAGS, gateways that would not be selected by only taking into account the routing metric can be used for outgoing connections, thus increasing the usage of network resources.

The proposed mechanism is implemented in real wireless routers and in simulations. The practical experiments show that QuAGS obtains an efficient load balancing between the network gateways, without lowering the average throughput. In the simulations, QuAGS has shown better performance when used in nodes that are far from any network gateways. Among the evaluated topologies, we highlight the linear network, which shows that load-balancing with QuAGS results in a throughput gain, and the 8x8 grid, which shows the QuAGS limitation that it should not be used in some scenarios, where QuAGS does not provide throughput gain.

Keywords: Multiple Gateways, Load Balancing, Wireless Mesh Networks.

Palavras-chave

1. Múltiplos *Gateways*.
2. Balanceamento de Carga.
3. Redes em Malha Sem Fio.

Abreviações

BSS	:	Basic Service Set
CSMA/CA	:	Carrier Sense Multiple Access with Collision Avoidance
CSMA/CD	:	Carrier Sense Multiple Access with Collision Detection
DCF	:	Distributed Coordination Function
DIFS	:	Distributed Interframe Space
DynTun	:	Dynamic Tunnels
FTP	:	File Transfer Protocol
GRE	:	Generic Routing Encapsulation
HNA	:	Home Network Announcement
IFS	:	Interframe Space
ML	:	Minimum Loss
NAT	:	Network Address Translation
OLSR	:	Optimized Link State Routing Protocol
OSI	:	Open Systems Interconnection
PCF	:	Point Coordination Function
QuAGS	:	Quality-Aware Gateway Selection
SIFS	:	Short Interframe Space
TCP	:	Transmission Control Protocol
UDP	:	User Datagram Protocol

Sumário

Lista de Figuras	xii
Lista de Tabelas	xiv
1 Introdução	1
1.1 Objetivos do Trabalho	2
1.2 Organização do Texto	3
2 Conceitos Básicos	4
2.1 Redes IEEE 802.11	4
2.1.1 Interferência em Redes IEEE 802.11	5
2.1.2 Controle de Acesso ao Meio em Redes IEEE 802.11	6
2.2 Redes em Malha Sem Fio	7
3 Descrição do Problema	11
3.1 Uso de Múltiplos <i>Gateways</i> em Redes <i>Mesh</i>	11
3.2 A Técnica NAT	13
3.3 Balanceamento de Carga e Utilização de Recursos em Redes em Malha Sem Fio	16
3.3.1 Soluções Existentes na Literatura	16
3.3.1.1 Soluções que necessitam um nó central conectado aos <i>gateways</i> da rede	17
3.3.1.2 Outras Abordagens	18

4	O Mecanismo QuAGS	20
4.1	O Mecanismo DynTun	20
4.2	Motivação para o Desenvolvimento do QuAGS	23
4.2.1	Potencial do DynTun	24
4.2.2	Limitação do DynTun	27
4.3	Proposta: Seleção Probabilística dos <i>Gateways</i>	28
4.4	Algoritmo da Proposta	29
4.5	Implementação	31
4.5.1	Módulos Necessários	31
4.5.2	Métodos	32
5	Avaliação de Desempenho	36
5.1	Experimentos na Rede Real	36
5.1.1	Resultados Experimentais	37
5.2	Resultados de Simulações	39
5.2.1	Topologia da Rede Interna	40
5.2.2	Topologia Linear	44
5.2.3	Topologias em Grade	47
5.2.3.1	Resultados - Topologia em Grade 7x7	50
5.2.3.2	Resultados - Topologia em Grade 8x8	54
6	Conclusão	56
6.1	Trabalhos Futuros	57
	Referências	59

Lista de Figuras

2.1	Acesso ao meio com DCF [31].	7
2.2	Exemplo de topologia de uma rede <i>mesh</i>	8
3.1	Escalabilidade em redes em malha sem fio.	12
3.2	O problema da justiça em redes em malha sem fio com 2 <i>gateways</i>	13
3.3	O funcionamento da técnica NAT.	14
3.4	Mudança de rotas em redes <i>mesh</i> com um <i>gateway</i>	14
3.5	Mudança de rotas em redes <i>mesh</i> com dois <i>gateways</i>	15
4.1	Topologia para o exemplo do funcionamento do DynTun.	21
4.2	Descoberta de <i>gateways</i> com o DynTun.	22
4.3	Criação de tabelas e túneis lógicos com o DynTun.	23
4.4	Seleção de <i>gateway</i> e marcação de pacotes com o DynTun.	24
4.5	Qualidade dos melhores caminhos entre um nó e os <i>gateways</i> da rede de testes.	25
4.6	Topologia para o exemplo do potencial do DynTun.	26
4.7	Queda da qualidade dos enlaces por causa de tráfego na rede.	26
4.8	Troca da escolha do melhor <i>gateway</i> com o DynTun.	27
4.9	Máquina de estados do QuAGS.	30
4.10	Relacionamento entre métodos do QuAGS.	33
5.1	Topologia dos experimentos da rede real.	37
5.2	Rede experimental - vazão média do nó origem.	38
5.3	Simulação da rede experimental, cenário com uma origem - vazão média por nó.	42

5.4	Simulação da rede experimental, cenário com três origens - vazão média por nó.	43
5.5	Topologia para simulações: topologia linear com 15 nós.	45
5.6	Simulação da rede linear - vazão média por nó.	45
5.7	Simulação da rede linear - histórico de uso dos <i>gateways</i>	46
5.8	Testes da Rede Linear - Vazão da rede no intervalo de 30s antes de cada transição no DynTun	47
5.9	Topologia para simulações: topologia em grade 7x7.	48
5.10	Topologia para simulações: topologia em grade 8x8.	49
5.11	Simulação da topologia em grade 7x7, cenário com uma origem e quatro <i>gateways</i> - vazão média por nó.	50
5.12	Simulação da topologia em grade 7x7, cenário com uma origem e quatro <i>gateways</i> - histórico de uso dos <i>gateways</i>	51
5.13	Simulação da topologia em grade 7x7, cenário com uma origem e três <i>gateways</i> - vazão média por nó.	52
5.14	Simulação da topologia em grade 7x7, cenário com duas origens e dois <i>gateways</i> - vazão média por nó.	53
5.15	Simulação da topologia em grade 8x8, cenário com origens no centro - vazão média por nó.	54
5.16	Simulação da topologia em grade 8x8, cenário com origens nos vértices - vazão média por nó origem.	55

Lista de Tabelas

2.1	Diferenças entre os padrões IEEE 802.11a, b e g.	4
5.1	Rede experimental - porcentagem de uso dos <i>gateways</i>	38
5.2	Parâmetros utilizados para a modelagem do tráfego nas simulações.	39
5.3	Parâmetros utilizados para o modelo de propagação <i>Shadowing</i> nas simulações.	41
5.4	Simulação da rede experimental, cenário com uma origem - porcentagem de uso dos <i>gateways</i>	42
5.5	Simulação da rede experimental, cenário com três origens - porcentagem de uso dos <i>gateways</i>	44
5.6	Simulação da rede linear - porcentagem de uso dos <i>gateways</i>	46
5.7	Simulação da topologia em grade 7x7, cenário com uma origem e quatro <i>gateways</i> - porcentagem de uso dos <i>gateways</i>	50
5.8	Simulação da topologia em grade 7x7, cenário com uma origem e três <i>gateways</i> - porcentagem de uso dos <i>gateways</i>	51
5.9	Simulação da topologia em grade 7x7, cenário com duas origens e dois <i>gateways</i> - porcentagem de uso dos <i>gateways</i>	53

Capítulo 1

Introdução

As redes em malha sem fio [7, 6] são compostas por roteadores sem fio tipicamente estacionários que formam um *backbone*. Esses roteadores cooperam para prover acesso à *Internet* para clientes móveis através de comunicação em múltiplos saltos. Geralmente, um dos roteadores do *backbone* é configurado como *gateway* para a *Internet*. Nesse cenário, a tendência é que o tráfego se concentre nos enlaces próximos ao *gateway* [5]. Quanto maior o número de clientes, maior a disputa pelo acesso ao meio nesses enlaces. Como os recursos do *gateway* são limitados, o número de clientes da rede é limitado pela capacidade do *gateway*.

Um problema característico nas redes em malha é que a vazão máxima obtida por um nó depende da sua distância, em número de saltos, para o *gateway* [4]. A vazão de um nó mais próximo ao *gateway* tende a ser maior que a vazão de um nó mais distante. Uma das causas deste problema provém do protocolo CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*) usado pelo padrão IEEE 802.11, geralmente empregado nas redes em malha sem fio. Esse protocolo garante que nós dentro do mesmo raio de transmissão têm a mesma probabilidade de conseguir acesso ao meio. O CSMA/CA, no entanto, não considera que os pacotes podem ter passado por outras disputas pelo meio anteriormente, em virtude da comunicação em múltiplos saltos. Isso gera injustiça entre os clientes, uma vez que os pacotes de clientes mais distantes experimentam mais disputas pelo meio até alcançarem o *gateway*.

O uso de múltiplos *gateways* para a *Internet* é uma técnica utilizada para aumentar a escalabilidade e a justiça em uma rede em malha sem fio [3, 22, 27]. Essa técnica permite o balanceamento do tráfego da rede entre os diferentes *gateways* e também reduz a injustiça, pois permite que clientes acessem a *Internet* através de um *gateway* mais próximo. Nesse cenário, os clientes podem experimentar trocas de *gateways* por causa de

dois fatores. Primeiro, os clientes devem ser capazes de escolher o melhor *gateway* e a melhor rota para ele em função de uma métrica de roteamento. Além disso, os clientes são potencialmente móveis, podendo trafegar por áreas cobertas por diferentes *gateways*.

Em uma rede em malha sem fio, as trocas de *gateway* podem ser um problema. Como tais redes, normalmente, são usadas para acesso à *Internet*, os *gateways* em geral usam a técnica NAT (*Network Address Translation*) [10] para isolar os endereços IP privados dos nós da rede em malha [8]. A troca de *gateways* em redes que utilizam NAT gera mudança no endereço IP de origem dos pacotes que deixam a rede, causando a quebra das conexões TCP ativas durante essa troca. Logo, é necessária uma solução que realize o balanceamento de carga entre *gateways* de uma rede em malha e que leve em conta também o problema da quebra de conexões TCP em virtude do NAT.

1.1 Objetivos do Trabalho

Este trabalho propõe um mecanismo de balanceamento de carga entre os múltiplos *gateways* de uma rede em malha sem fio. O mecanismo proposto, chamado de QuAGS (*Quality-Aware Gateway Selection*), seleciona probabilisticamente o *gateway* a ser usado por uma conexão de um cliente. A probabilidade de seleção de cada *gateway* é calculada a partir da métrica do caminho do cliente a esse *gateway*. O objetivo de utilizar probabilidades, e não somente a métrica de roteamento, para a seleção é fazer com que *gateways* que *a priori* não seriam utilizados possam ser selecionados, aumentando o uso dos recursos da rede.

O QuAGS é implementado em uma rede experimental, utilizando por base a ferramenta DynTun [8], o que garante que não há problemas de quebra de conexão em virtude do uso da técnica NAT. A quebra de conexão não ocorre devido ao DynTun forçar que todos os pacotes de uma determinada conexão utilizem sempre o mesmo *gateway* para se comunicar com redes externas. O QuAGS usa a marcação de pacotes e a criação de túneis lógicos dos nós aos *gateways*. Uma nova conexão de um cliente conectado a um nó, com destino externo à rede, seleciona um *gateway* com probabilidade proporcional à qualidade do caminho entre o nó e o *gateway*.

O mecanismo é analisado através de testes reais em uma rede experimental e de simulações em diferentes cenários e topologias. Resultados mostram que o QuAGS obtém um balanceamento de carga eficiente entre os *gateways* da rede, sem que a vazão seja comprometida e evita que o caminho entre um nó e um *gateway* seja saturado.

1.2 Organização do Texto

No Capítulo 2 serão abordados conceitos básicos em redes sem fio IEEE 802.11 e sobre redes em malha sem fio, ou simplesmente redes *mesh*. Será explicado o funcionamento do mecanismo CSMA/CA usado pelo IEEE 802.11 para o controle de acesso ao meio. Uma breve discussão sobre interferência na camada física do IEEE 802.11 também será apresentada, uma vez que a interferência influencia o desempenho dos protocolos de roteamento em uma rede *mesh*.

No Capítulo 3 o uso de múltiplos *gateways* em redes *mesh* será discutido. Será apresentado um resumo do funcionamento da técnica NAT e o problema gerado pelo seu uso em redes *mesh* com múltiplos *gateways*. Uma discussão sobre balanceamento de carga em redes *mesh* com múltiplos *gateways* também será realizada. Também serão apresentadas algumas soluções propostas na literatura para a resolução do problema de balanceamento de carga em redes *mesh* que usam múltiplos *gateways*. Esse capítulo apresentará o problema que este trabalho busca solucionar. Também será realizada nesse capítulo a classificação das soluções existentes na literatura e as características dessas soluções que são evitadas pela proposta deste trabalho, assim como o porquê de serem evitadas.

O Capítulo 4 explicará brevemente o mecanismo DynTun, que serve de base para a implementação do QuAGS. O mecanismo QuAGS será proposto e suas características e seu modo de funcionamento serão apresentados, além de sua implementação. Ainda neste capítulo, as diferenças entre os dois trabalhos serão evidenciadas.

No Capítulo 5 serão apresentados os experimentos realizados para validação do QuAGS, em simulações e em uma rede experimental. Será feita uma análise dos resultados e conclusões sobre a ferramenta, de acordo com os resultados dos testes, serão apresentadas.

Finalmente, o Capítulo 6 concluirá o que foi abordado no texto e apresentará possíveis trabalhos futuros.

Capítulo 2

Conceitos Básicos

As redes IEEE 802.11, também conhecidas como *Redes Wi-Fi*, são redes sem fio locais. Essa tecnologia é atualmente o padrão para WLANs (*Wireless Local Area Network*) [19] e se encontra em constante desenvolvimento e tem como uma das principais características ser de baixo custo. Assim, pode-se notar que alguns tipos de serviços podem ser implementados nesse tipo de rede. Entre esses serviços está o acesso à Internet.

2.1 Redes IEEE 802.11

O padrão IEEE 802.11 [19] trata das camadas física e de acesso ao meio (MAC) do modelo OSI (*Open Systems Interconnection*). Ele também é conhecido mais popularmente como *Wi-Fi*. Ele tem várias extensões, dentre as quais as mais usadas são os padrões 802.11a, 802.11b e 802.11g. A Tabela 2.1 apresenta algumas das características dos três padrões mais usados atualmente.

Os padrões têm muitas semelhanças, dentre elas o protocolo de acesso ao meio, a estrutura de quadro e a capacidade de mudar a taxa de transmissão de acordo com as condições do meio. O mais importante é que todos suportam o modo *ad-hoc*, fator interessante para este trabalho, de modo que em qualquer um desses padrões pode-se implementar uma rede em malha sem fio. A Seção 2.2 apresentará o modo *ad-hoc* e sua importância no contexto de redes em malha sem fio. Com o uso do protocolo de acesso ao meio, o CSMA/CA

Padrão	Faixa de frequência	Taxa de transmissão
802.11a	5,1 a 5,8 GHz	até 54 Mbps
802.11b	2,4 a 2,485 GHz	até 11 Mbps
802.11g	2,4 a 2,485 GHz	até 54 Mbps

Tabela 2.1: Diferenças entre os padrões IEEE 802.11a, b e g.

(*Carrier Sense Multiple Access with Collision Avoidance*), cada estação, antes de realizar a transmissão, deve escutar o canal e verificar se há alguma outra estação transmitindo. Assim, quanto mais transmissões um determinado pacote tem, mais vezes ele concorre ao meio com outros pacotes da rede. A Seção 2.1.2 apresenta detalhadamente o funcionamento do CSMA/CA. O seu objetivo é diminuir a probabilidade de colisão de quadros transmitidos por estações ao mesmo tempo.

2.1.1 Interferência em Redes IEEE 802.11

A interferência do espectro de radiofrequência é um fenômeno presente em redes sem fio IEEE 802.11. Ela ocasiona uma significativa perda de desempenho nesse tipo de rede. Para descrever interferência, deve-se levar em conta fatores específicos como condições do ambiente, *hardware* de transmissão e recepção, entre outros. Os tipos básicos de interferência que uma rede IEEE 802.11 pode sofrer podem ser classificados como:

- Interferência interna à rede

A interferência interna à rede refere-se à interferência gerada entre os nós de uma mesma rede sem fio. A transmissão realizada por um nó estará interferindo na transmissão de outro nó quando, no receptor, o nível do sinal estiver abaixo do limiar para detecção de portadora pelo CSMA/CA, mas é suficiente para servir como ruído para transmissão de outros nós.

- Interferência de outras redes IEEE 802.11

A interferência gerada por outras redes IEEE 802.11 acontece quando outras redes utilizam o mesmo canal ou um canal adjacente dentro da área de cobertura da rede, ou próximo a ela. Uma vez que as redes IEEE 802.11 a cada dia são mais populares, elas são utilizadas livremente pelos mais diversos tipos de usuários. Dos 11 canais definidos pelo padrão IEEE 802.11, somente os canais 1, 6 e 11 podem ser utilizados simultaneamente sem interferência, pois suas frequências de operação não se sobrepõem. Logo, até 3 redes podem operar dentro de uma mesma área de cobertura simultaneamente sem causarem interferência entre si, desde que cada uma esteja utilizando um desses 3 canais. Quando redes diferentes utilizam, dentro de uma mesma área de cobertura, o mesmo canal ou canais adjacentes, ocorre interferência, pois o meio de transmissão é compartilhado.

- Interferência externa

As redes IEEE 802.11 utilizam uma banda de frequências não-licenciada, como apresentada na Tabela 2.1. Por não utilizar uma faixa de frequências exclusiva, o padrão IEEE 802.11 foi desenvolvido de forma que a sua resistência a interferências fosse alta. Mesmo assim, diversos tipos de dispositivos (como fornos microondas e aparelhos sem fio domésticos) [24] e tecnologias de transmissão de dados (como *Bluetooth* [20]), que compartilham a mesma faixa de frequência das redes IEEE 802.11, geram interferências que podem degradar consideravelmente seu desempenho.

2.1.2 Controle de Acesso ao Meio em Redes IEEE 802.11

O padrão IEEE 802.11 oferece duas funções na subcamada de acesso ao meio (MAC), a DCF (*Distributed Coordination Function*) e a PCF (*Point Coordination Function*). A PCF necessita de um elemento centralizador realizando a coordenação de qual estação pode utilizar o meio em qual instante. Logo, pela necessidade desse nó coordenador, a PCF não é adequada para uma rede com múltiplos saltos, que é o tipo de rede utilizado neste trabalho.

O DCF é o método de acesso ao meio padrão do IEEE 802.11. Ele permite o compartilhamento do meio entre as estações utilizando o método de acesso CSMA/CA [25]. De acordo com esse método, cada estação, antes de transmitir qualquer quadro, deverá observar o estado do meio e determinar se existe alguma outra estação transmitindo no momento. Caso o meio esteja ocupado com alguma outra transmissão, a estação deve aguardar até o final desta transmissão.

O ato de evitar colisão é utilizado para aumentar o desempenho do CSMA, através da tentativa de divisão do meio sem fio de forma mais igualitária possível entre os nós pertencentes ao domínio. O CSMA/CA se diferencia do CSMA/CD (*Carrier Sense Multiple Access with Collision Detection*) [25], utilizado em redes cabeadas, por causa do meio utilizado. No espectro de radiofrequência, as colisões não podem ser detectadas quando ocorridas no nó transmissor. Um dos problemas de comunicação no meio sem fio é a impossibilidade de escutar enquanto se realiza uma transmissão, assim sendo impossível detectar uma colisão.

O intervalo de tempo entre os quadros é denominado IFS (*Interframe Space*). A estação transmissora deve se assegurar que o meio estará livre por um intervalo de tempo específico, utilizando o mecanismo de detecção da portadora. No DCF são utilizados dois tipos de IFS, o DIFS (*Distributed Interframe Space*) antes da transmissão de quadros de dados e o SIFS (*Short Interframe Space*) antes de um quadro ACK.

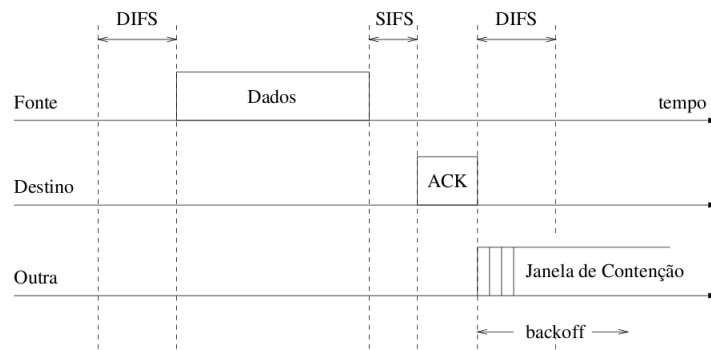


Figura 2.1: Acesso ao meio com DCF [31].

Antes de transmitir um quadro de dados, o nó transmissor espera um intervalo de tempo igual ao DIFS somado a um intervalo de tempo de *backoff* aleatório dado pela multiplicação entre um inteiro selecionado aleatoriamente dentro de um intervalo entre zero e CW (*Contention Window*), que varia de CW_{min} a CW_{max} , e um intervalo de tempo definido pelo padrão. A Figura 2.1 ilustra o acesso ao meio com DCF.

Após a escolha do tempo de *backoff*, o nó transmissor deve decrementar o contador de *backoff* enquanto o meio estiver livre. Caso o meio fique ocupado novamente, o contador será interrompido. Ele volta a ser decrementado após o meio ficar livre novamente por um intervalo de tempo igual a DIFS. Quando o contador é zerado, o nó realiza a transmissão de um quadro. Na primeira transmissão realizada pelo nó, o valor da janela de contenção é CW_{min} , que é chamado de janela de contenção mínima. A cada transmissão sem sucesso, o valor da janela de contenção é dobrado até CW_{max} . Após receber com sucesso um quadro, o nó receptor aguarda um tempo SIFS e então envia ao nó transmissor um quadro ACK. Caso o nó transmissor não receba o ACK dentro de um intervalo de tempo pré-definido, ou detecte a transmissão de um quadro diferente no canal, a transmissão é considerada mal sucedida e o quadro é retransmitido de acordo com as regras de *backoff* descritas acima.

2.2 Redes em Malha Sem Fio

A unidade básica do padrão IEEE 802.11 é o BSS (*Basic Service Set*). Um BSS pode ser configurado como independente (IBSS) ou infra-estruturado. No modo infra-estruturado, os nós comunicam-se através de um nó central denominado de ponto de acesso (*Access Point* - AP). No modo independente, popularmente conhecido como modo *ad-hoc*, os nós se comunicam de forma direta sem a necessidade de um elemento intermediário de

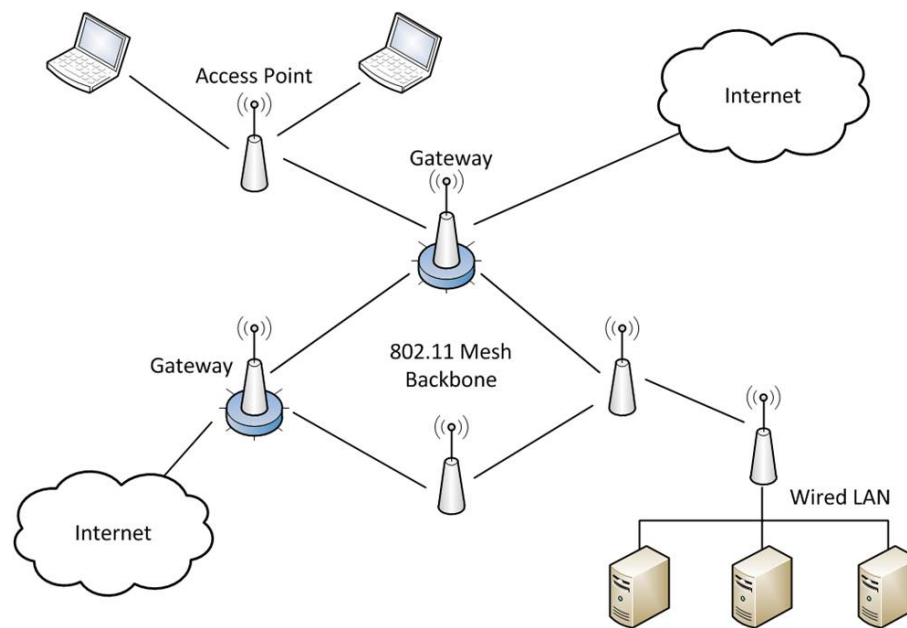


Figura 2.2: Exemplo de topologia de uma rede *mesh*.

comunicação.

As redes em malha sem fio (ou redes *mesh*) consistem de roteadores e clientes *mesh* configurados no modo *ad-hoc* com o objetivo de prover serviços em uma área extensa que não poderia ser coberta por um único AP. Os roteadores *mesh* são fixos e constituem o *backbone* da rede. Roteadores *mesh* podem ser configurados como *gateways*, que são os nós responsáveis pela comunicação com outras redes (por exemplo, a Internet). Na Figura 2.2 pode-se ver um exemplo de rede *mesh*, composta por clientes *mesh*, roteadores e dois *gateways* para a Internet.

Nesse tipo de rede, os nós se comunicam por múltiplos saltos, de forma que todos os nós da rede estão em um mesmo nível de hierarquia. Assim, um pacote é encaminhado salto a salto até que chegue ao destino. Esse tipo de rede ainda tem como características importantes:

- Auto-configuração: todos os nós estão em um mesmo nível de hierarquia, sem necessidade de um ponto de acesso central, de forma que novos nós adicionados (desde que usem parâmetros, como identificador da rede e canal, iguais aos outros nós da rede) são automaticamente reconhecidos e incorporados à rede;
- Tolerância a falhas: a falha ou movimentação de um nó comum não compromete a rede como um todo, sendo os *gateways* os únicos nós críticos da rede;

- Suporte à mobilidade: dentro da área de cobertura da rede, um cliente pode estar conectado a qualquer nó, mesmo não sendo o nó que ele se conectou originalmente, que ele continuará com acesso à rede;

Apesar de redes em malha sem fio poderem ser híbridas [2], utilizando vários padrões, como *Wi-Fi* (802.11), *WiMax* (802.16) [21], redes de sensores, entre outros, este trabalho se restringe ao estudo de redes *mesh* no padrão IEEE 802.11, que, como dito anteriormente, é o padrão mais usado e difundido na atualidade.

As redes em malha sem fio, por causa de sua natureza de comunicação em múltiplos saltos, e pela alta variação nas rotas existentes, têm como uma de suas características a escolha dinâmica de rotas. Assim, dois nós quaisquer pertencentes à rede poderão estar conectados através de diferentes caminhos. Logo, se faz necessário que um dos caminhos seja selecionado para que os pacotes de uma transmissão entre dois nós seja realizada. Porém, a qualidade dos enlaces tende a sofrer variações e, consequentemente, a qualidade do caminho com múltiplos saltos. Isso ocorre por causa das interferências descritas na Seção 2.1.1 as quais o meio sem fio é suscetível. Então, se faz necessária a constante avaliação da qualidade dos enlaces em uma rede em malha sem fio para se determinar a melhor rota entre duas estações. O agente responsável por essa avaliação é chamado de protocolo de roteamento.

O protocolo de roteamento precisa ter um critério para qualificar as possíveis rotas entre os nós da rede. Esse critério é chamado de métrica de roteamento. De acordo com a métrica, as rotas são avaliadas e o melhor caminho é selecionado, usando um algoritmo de caminho mínimo, para a transmissão de pacotes entre dois nós quaisquer da rede. Geralmente as métricas coletam estatísticas dos enlaces, como perda de pacotes, número de saltos, tempo de transmissão, entre outros. Então, as métricas usam algum tipo de fórmula para ponderar as estatísticas e atribuir pesos aos enlaces da rede. O protocolo de roteamento OLSR-ML (*Optimized Link State Routing Protocol - Minimum Loss*) utilizado na rede dos testes deste trabalho e outros protocolos de roteamento são descritos por Campista *et al.* [6].

Assim, a vazão máxima obtida por um nó depende da sua distância, em número de saltos, para o *gateway* [4]. O mecanismo CSMA/CA é um dos responsáveis pela vazão de um nó mais próximo ao *gateway* tender a ser maior que a vazão de um nó mais distante. Como o algoritmo do CSMA/CA não considera o número de transmissões que um pacote se submete, esses pacotes mais distantes sofrem mais disputas pelo meio. Em uma rede em malha sem fio com múltiplos saltos, esse fator é importante. Como o protocolo de

roteamento tende a diminuir a métrica de caminhos que tenham um número grande de saltos, usar a métrica como fator de escolha de caminhos para evitar muitas disputas pelo meio torna-se uma idéia interessante.

Capítulo 3

Descrição do Problema

Devido ao seu baixo custo e facilidade de instalação, o uso de redes em malha sem fio tem crescido atualmente. Soluções universitárias, como o Projeto ReMOTE [18], e comerciais, como a da Cisco [15], são propostas. Porém, com o aumento da área de cobertura de uma rede, e consequentemente, com o aumento do número de nós, problemas podem surgir na rede. Alguns desses problemas são o balanceamento de carga entre os múltiplos *gateways* da rede, a subutilização de recursos e a quebra de conexões TCP em virtude do uso de mais de um *gateway* e das frequentes mudanças de rotas. Na literatura, existem algumas propostas de soluções para esses problemas, que são descritas a seguir.

3.1 Uso de Múltiplos *Gateways* em Redes *Mesh*

Uma das vantagens de uma rede em malha sem fio é o seu crescimento de forma escalável [2]. A Figura 3.1a apresenta uma rede em malha sem fio com um *gateway*, nó A, e vários nós, cobrindo a área denotada pela linha oval externa.

Como os nós se comunicam de forma *ad-hoc*, se for necessário que essa rede cubra uma nova área maior, a simples adição de nós sem necessidade da reconfiguração dos nós anteriores é suficiente para que a rede cubra essa nova área maior, representada pela Figura 3.1b. Porém, a adição dos nós pode trazer problemas em relação à justiça em uma rede em malha sem fio. Na mesma figura, os nós U, W, X, Y e Z, por exemplo, estão pelo menos a quatro ou cinco saltos do *gateway*, enquanto outros nós estão a um ou dois saltos. Clientes conectados a esses nós podem ter vazão inferior e atraso superior se comparado com clientes conectados a outros nós mais próximos do *gateway*, pois um número maior de saltos significa que um pacote sofre concorrência pelo meio de acesso um número maior de vezes. Um pacote originado de um nó distante a um *gateway* também

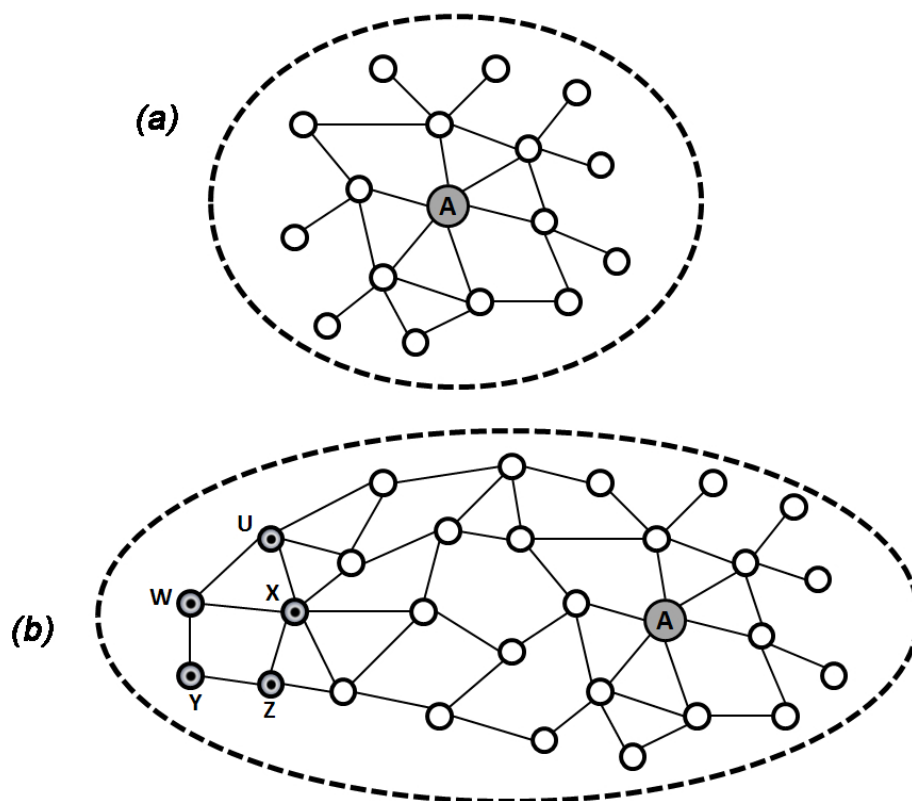


Figura 3.1: Escalabilidade em redes em malha sem fio.

tem maior chance de sofrer perda e um maior número de transmissões.

Assim, uma técnica que se faz necessária para resolução desse problema é o uso de múltiplos *gateways* na rede. A Figura 3.2 segue o exemplo anterior. Ao tornar o nó X da Figura 3.1b em um *gateway*, chamado agora de nó B, o problema de justiça pode ser amenizado, uma vez que os clientes terão aproximadamente o mesmo número de saltos até o *gateway* mais próximo, que provavelmente deverá ser o mais utilizado.

Alguns trabalhos na literatura falam acerca da importância do uso de múltiplos *gateways*. Akella *et al.* [1] quantificam o potencial aumento de desempenho na rede com o uso de múltiplos *gateways* através da análise de registros de tráfego real da Internet. Eles concluem que o desempenho em redes com dois *gateways* pode aumentar em 25%. Porém, esse estudo foi realizado em uma rede cabeada infra-estruturada.

No artigo de Tsai e Moors [32] é discutida a importância da estratégia de localização de múltiplos *gateways*. O trabalho avalia através de simulações um algoritmo para posicionamento de *gateways*. Porém não apresenta estratégias para balanceamento de carga e para solução do problema de quebra de conexão.

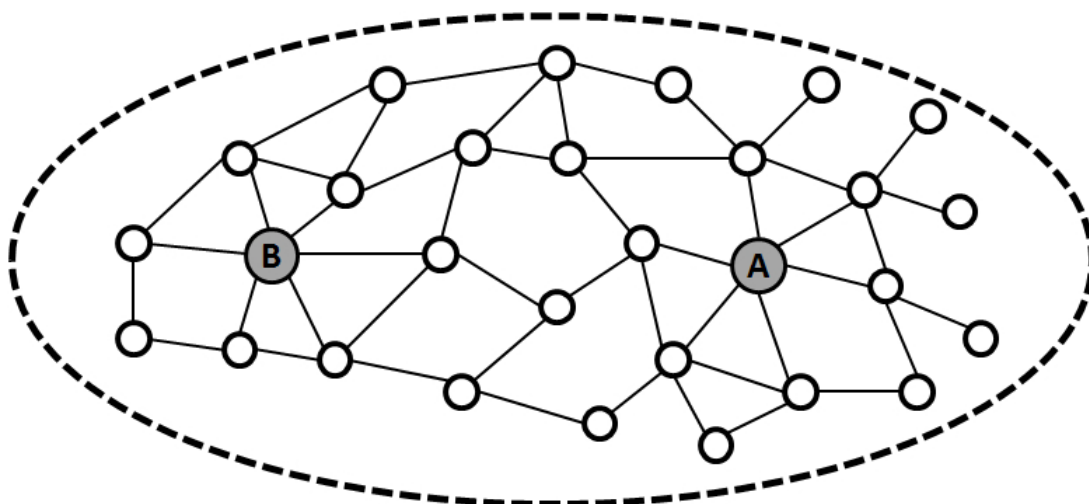


Figura 3.2: O problema da justiça em redes em malha sem fio com 2 *gateways*.

3.2 A Técnica NAT

Uma técnica que é conhecida e amplamente utilizada no contexto de redes em geral é o NAT (*Network Address Translation* [10]), por causa da escassez de endereços IPv4 na Internet. Essa técnica resumidamente consiste em mapear endereços IPs privados para um único endereço IP público da rede para se conectar à Internet. A ideia principal do NAT é que estações que possam se comunicar umas com as outras na Internet não têm a necessidade de terem endereços IP públicos únicos. Com o uso do NAT, uma estação pode ter um endereço IP privado único dentro de seu escopo de rede local, mas que não é necessariamente único em um escopo global.

O funcionamento do NAT é apresentado na Figura 3.3. A partir do momento em que uma estação da rede em malha envia um pacote para outra estação externa através de uma porta, formando uma tupla (IPprivA, PortaW), o roteador responsável pelo NAT mudará o endereço de origem do nó e a porta para uma tupla (IPpúblico, PortaY) para se comunicar com o servidor externo e armazena essa informação em uma tabela. O pacote de resposta do servidor externo chega ao roteador responsável pelo NAT com a tupla de destino (IPpúblico, PortaY). O roteador consulta a tabela e verifica a qual dos clientes privados pertence esse pacote. Consultando a tabela, a entrada pela qual corresponde a tupla do pacote é (IPprivA, PortaW). O cabeçalho do pacote é modificado: a tupla de destino (IPpúblico, PortaY) é substituída por (IPprivA, PortaW). O pacote então é encaminhado ao cliente privado que originou a conexão.

Logo, pode-se ver que tanto o uso de múltiplos *gateways* quanto o uso da técnica NAT

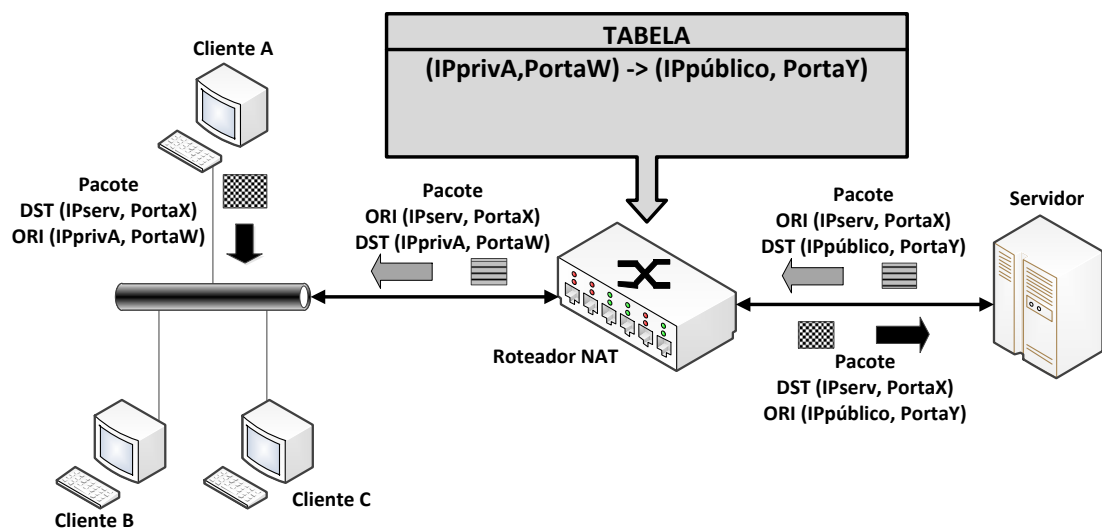


Figura 3.3: O funcionamento da técnica NAT.

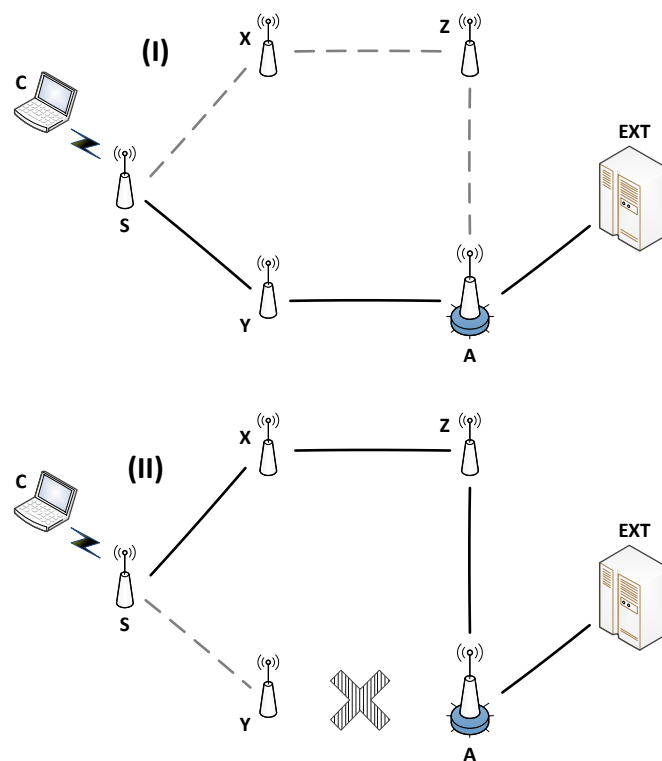


Figura 3.4: Mudança de rotas em redes *mesh* com um *gateway*.

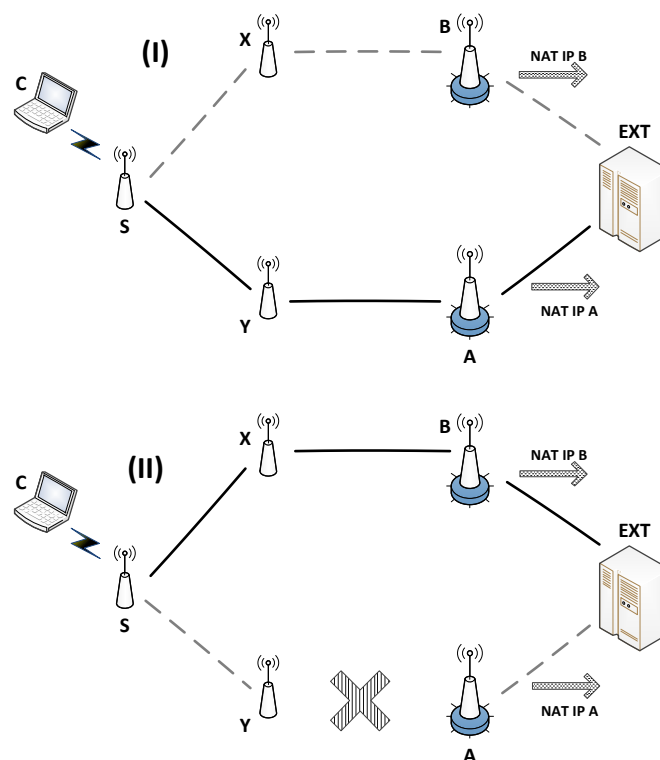


Figura 3.5: Mudança de rotas em redes *mesh* com dois *gateways*.

são técnicas que podem ser utilizadas em conjunto no contexto de redes em malha sem fio. Porém, outro problema ocorre ao usar as duas técnicas simultaneamente. A Figura 3.4.(I) apresenta uma rede em malha com vários nós, sendo um deles um nó *gateway* que realiza NAT, denotado por A, um cliente C conectado ao nó S e um servidor externo EXT, que pode ser um servidor na Internet.

A rota $C \rightarrow EXT$ é denotada na Figura 3.4.(I) pelos enlaces em negrito, através do *gateway* A. Como dito anteriormente na Seção 2.2, as rotas em redes em malha sem fio mudam dinamicamente, por causa de interferências, ou simplesmente por variações na métrica do protocolo de roteamento. Com a falha no enlace $Y \rightarrow A$, ilustrada na Figura 3.4.(II), o protocolo de roteamento recalcula uma outra rota para EXT através do *gateway* A, passando pelo nó B. Nesse exemplo, o problema seria a perda momentânea de pacotes.

A Figura 3.5.(I) apresenta uma rede com topologia semelhante à do exemplo anterior, porém o nó B é um segundo *gateway* da rede. Ambos os *gateways* realizam NAT, cada um com um endereço IP público diferente. Ocorrendo o mesmo tipo de falha no enlace do exemplo com apenas um *gateway*, o protocolo de roteamento recalcula a rota e o *gateway* B é uma opção melhor do que o *gateway* A para o acesso ao servidor EXT, como mostra a

Figura 3.5.(II). Como ambos os *gateways* da rede estão usando a técnica NAT, o endereço IP no cabeçalho dos pacotes do cliente muda, de forma que isso infringe a semântica da conexão TCP [29] e a conexão é quebrada, que é um problema de escala maior do que a perda momentânea de pacotes, pois é necessário que outra conexão seja iniciada para que pacotes sejam transmitidos novamente.

3.3 Balanceamento de Carga e Utilização de Recursos em Redes em Malha Sem Fio

Um outro problema que pode ocorrer em uma rede em malha sem fio é que os recursos da rede podem ser subutilizados. Os nós mais próximos do *gateway* fazem parte de caminhos de outros nós para o *gateway*. Assim, os nós mais próximos encaminham mais pacotes que os nós mais distantes e, por isso, podem ter falta de recursos, como filas de pacotes cheias e excesso de processamento. Por outro lado, nós mais distantes, em geral, encaminham apenas pacotes de seus clientes e, assim, podem ser subutilizados. Os *gateways* com mais nós próximos geralmente serão mais utilizados que outros, assim como o meio de sua área de alcance será mais disputado. Assim, se faz importante transferir uma parcela do tráfego para outros *gateways*, que mesmo sendo mais distantes e aumentando o número de saltos, pode aliviar nesta área mais utilizada a carga dos nós e evitar a saturação dos recursos. Com essa ação, os nós em área mais esparsas também não sofreriam de subutilização.

Como dito em [12], associar nós ao *gateway* mais próximo em termos de métrica de roteamento pode causar desbalanceamento da carga, pois as métricas geralmente não são cientes da carga da rede. Vários fatores podem causar essa situação, como demanda de tráfego diferente entre as “regiões” de uma área coberta pela rede, falha de *gateways* e mau posicionamento dos *gateways* da rede por falta de planejamento.

3.3.1 Soluções Existentes na Literatura

Na literatura, vários trabalhos propõem soluções para realizar o balanceamento de carga em redes em malha sem fio com múltiplos *gateways*. Porém, nenhuma delas tem abordagem semelhante a este trabalho. Essa seção apresenta algumas dessas propostas e apresenta as diferenças delas para a solução proposta neste trabalho.

3.3.1.1 Soluções que necessitam um nó central conectado aos *gateways* da rede

Ancillotti *et al.* [3] apresentam duas estratégias adaptativas para balanceamento de carga e seleção de *gateways*, que se baseiam nas diferentes capacidades de cada *gateway*. A MRC (*Maximum Residual Capacity*) propõe que um nó cliente selecione o *gateway* com a máxima capacidade disponível. Assim, outros *gateways* distantes do cliente podem ser escolhidos. Por outro lado, a MNC (*Maximum Normalized Cost*) define que cada nó cliente deve calcular uma função utilidade para cada *gateway*. Quanto maior o custo entre um cliente e o *gateway*, menor a utilidade que o nó associa a ele. Os desempenhos do MNC e do MRC são comparados com um algoritmo convencional de escolha do menor caminho através de testes em uma rede experimental com *streaming* de vídeo. Os resultados apontam um aumento da capacidade da rede com o MNC e o MRC, através do aumento do número de vídeos por cliente. O tempo de rebufferização dos vídeos para até 5 vídeos por cliente é mínimo, se comparado com o protocolo de roteamento tradicional.

O trabalho de Lakshmanan *et al.* [26] ressalta a importância de um nó cliente poder usar múltiplos *gateways*, sem particionamento da rede, para alcançar o balanceamento de carga entre eles. Eles propõem um modelo para redes em malha sem fio chamado MGA (*Multiple-Gateway Association*), no qual é considerado que cada cliente pode escolher mais de um *gateway* para realizar comunicação com outras redes. A escolha do número de *gateways* e quais são escolhidos por um cliente são realizados segundo um algoritmo que tenta identificar as associações e os fluxos na rede de maneira gulosa. Esse algoritmo é modelado para maximizar a vazão agregada da rede, tem como entrada a demanda de um fluxo e como saída um vetor de alocação para cada par fluxo-*gateway*. O trabalho é avaliado através de simulações, comparando com um modelo SGA (*Single Nearest Gateway Association*), que consiste em todo nó cliente, para todo tráfego que ele gera, escolher o *gateway* mais próximo a ele. Os resultados mostram um ganho de vazão em torno de 10% do MGA sobre o SGA. Outro resultado que o trabalho mostra é o aumento da vazão da rede ao realizar a adição de *gateways*. Com uma carga de 1Mbps na rede, esse valor de vazão só é alcançado com a inserção do quarto *gateway*. O MGA tem uma perda de pacotes superior ao SGA, porém os autores apontam que, mesmo assim, a vazão com o MGA é superior ao SGA.

Ito *et al.* [23] propõem um método baseado em sessão com o objetivo de maximizar a vazão de fluxos TCP em comunicações entre redes em malha sem fio com múltiplos *gateways* e redes externas. Ser baseado em sessão significa que os nós memorizam o rela-

cionamento entre uma sessão de uma comunicação e o *gateway* escolhido para ela, ao invés do melhor *gateway* ser escolhido por cada pacote gerado. Assim, todos os pacotes dessa sessão passarão pelo mesmo *gateway*. Para que o algoritmo de escolha possa executar, os *gateways* da rede precisam enviar os pacotes para um nó central que também memoriza o relacionamento entre a sessão e o *gateway* escolhido por ela e redireciona os pacotes para a rede externa. Os modelos de distribuição por sessão e distribuição por pacote são comparados através de simulações, que apontam um ganho de vazão de até três vezes, sem causar problemas de injustiça na rede com o aumento do número de clientes.

Esses trabalhos diferem do QuAGS pela necessidade da existência do nó central conectado aos *gateways* da rede. Apesar do NAT poder ser executado por esse nó gerenciador da rede, o que solucionaria o problema de quebra de conexões TCP, o uso desse nó gera dois problemas. Em primeiro lugar, o nó gerenciador é um ponto central de concentração do tráfego da rede, limitando a capacidade de escoamento de tráfego. Além disso, o nó gerenciador é um ponto de falha, pois sem ele a rede se torna inacessível. O QuAGS, por outro lado, age de forma distribuída, onde cada conexão de cada cliente seleciona o seu melhor *gateway*, de forma a evitar a concentração de tráfego da rede e a inserção do ponto central crítico de falha.

3.3.1.2 Outras Abordagens

Gálvez *et al.* [12] propõem o GWLB (*Gateway Load-Balancer*), um algoritmo adaptativo para balanceamento de carga entre *gateways*. O trabalho tem como objetivos: aliviar o congestionamento da rede, através de re-roteamento de tráfego de domínios congestionados para domínios menos congestionados; melhorar a justiça de fluxo inter-domínio através do balanceamento de carga entre os domínios, quando for benéfico. Os autores discutem estratégias para evitar oscilações na rede, analisando-as em tempo real e tomando decisões de acordo com o estado da mesma. Porém, o GWLB realiza o balanceamento de carga na rede apenas para o tráfego de entrada. O tráfego de saída da rede é direcionado sempre para o *gateway* mais próximo. A proposta deste trabalho, QuAGS, por outro lado, lida com o balanceamento de carga no tráfego de saída da rede.

Iqbal *et al.* [22] propõem o LBGD-AODV (*Load-Balanced Gateway Discovery-AODV*), uma modificação no protocolo de roteamento AODV para proporcionar balanceamento de carga em redes com múltiplos *gateways*. Quando um novo nó entra na rede, causando mudança na topologia, o algoritmo reselectiona o melhor *gateway* para cada nó. O algoritmo usa informações como número de saltos e carga de cada *gateway*. No entanto, esse

trabalho se diferencia do QuAGS pela escolha do *gateway* para cada nó cliente ser fixa enquanto a topologia não é modificada, ou seja, a rede é particionada entre os *gateways*.

Nandiraju *et al.* [27] propõem um algoritmo que constrói uma lista de *gateways* para cada nó, ordenando-a pela qualidade da rota. Cada nó envia seu tráfego para o *gateway* que possui a rota de maior qualidade. Quando a fila de pacotes de um *gateway* ultrapassa um determinado limiar, ele envia uma mensagem de notificação de congestionamento para os nós conectados a ele. Ao receber essa notificação, um nó seleciona o próximo *gateway* da sua lista e passa a usá-lo. O problema deste algoritmo é que a rede pode sofrer com a oscilação de tráfego entre seus diferentes *gateways*. O QuAGS busca realizar um balanceamento de carga mais eficiente entre os diferentes *gateways*, evitando que os nós somente realizem trocas quando há saturação.

Capítulo 4

O Mecanismo QuAGS

Este capítulo apresenta o mecanismo QuAGS (*Quality-Aware Gateway Selection*) [9], que tem como objetivo aumentar o uso de recursos da rede, evitando a saturação de um caminho para um *gateway* por um nó. Para entender melhor o seu funcionamento, inicialmente é apresentado o mecanismo DynTun, um trabalho anterior que aborda o problema de quebra de semântica de conexões TCP em redes em malha com múltiplos *gateways* e que serviu de inspiração para a presente proposta. Detalhes acerca da implementação do QuAGS também são apresentados nesse capítulo.

4.1 O Mecanismo DynTun

O mecanismo DynTun (*Dynamic Tunnels*) [8] se baseia na criação dinâmica de túneis lógicos e na marcação de pacotes pertencentes a uma mesma conexão, de forma transparente para o usuário. Seu principal objetivo é solucionar o problema da quebra da semântica das conexões TCP que ocorre com a alternância entre múltiplos *gateways* que realizam NAT.

A Figura 4.1 apresenta um exemplo de topologia de uma rede em malha sem fio. Ela é composta por sete nós, sendo dois *gateways* A e B. Ambos os *gateways* fazem NAT e estão conectados a um servidor EXT externo à rede. Um cliente C está conectado ao nó X da rede. Essa topologia será utilizada ao longo desse exemplo.

O nó X está executando o DynTun, que é um processo contínuo que executa em *loop* uma sequência de passos. O primeiro passo é identificar quantos e quais são os *gateways* da rede. Essa identificação é feita através da mensagem HNA (*Home Network Announcement*) que os *gateways* enviam por *broadcast*. Essa mensagem é específica do protocolo de roteamento OLSR, mas o DynTun pode ser implementado sobre qualquer

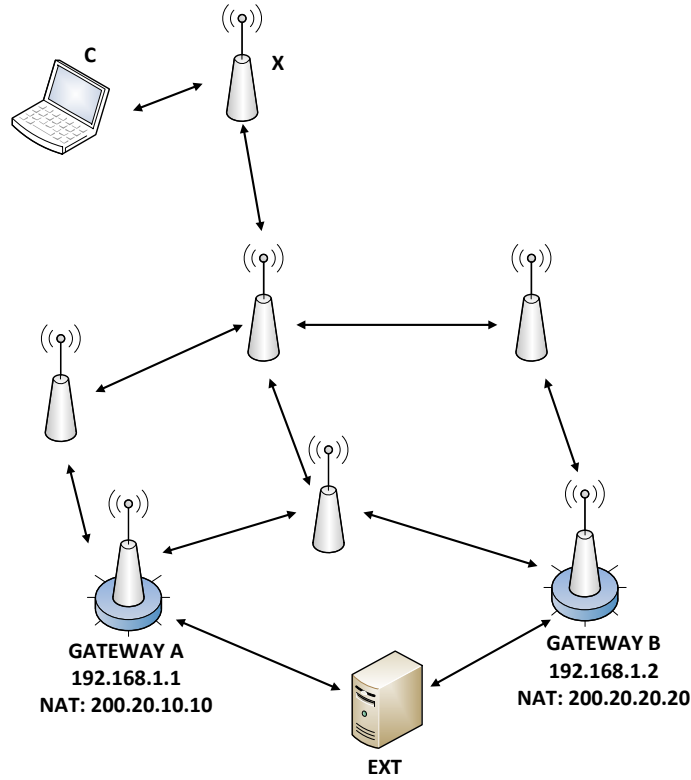


Figura 4.1: Topologia para o exemplo do funcionamento do DynTun.

protocolo, desde que seja implementado um sistema de anúncio de *gateways*. Conforme a Figura 4.2 apresenta, as mensagens HNA de ambos os *gateways* da rede eventualmente chegam ao nó X.

Com o conhecimento de quem são os *gateways* da rede, o DynTun cria um túnel lógico através de interfaces virtuais de rede entre um dado *gateway* e o nó que está executando o mecanismo. Também é criado no nó que executa o DynTun uma tabela de roteamento para cada *gateway* da rede. Cada tabela tem apenas um registro de endereço IP, que é o endereço IP lógico do túnel na extremidade do *gateway*.

A Figura 4.3 segue o exemplo, demonstrando o passo explicado acima. Nela, após o nó X receber as mensagens HNA dos *gateways*, ele cria duas interfaces virtuais X_a e X_b (uma referente a cada *gateway* da rede), que são as extremidades de dois túneis lógicos na rede. Respectivamente, X_a é a interface do túnel lógico T_a no nó X e X_b é a interface do túnel lógico T_b no nó X. As interfaces virtuais A_x e B_x são criadas respectivamente nos *gateways* A (sendo a interface do túnel T_a em A) e B (sendo a interface do túnel T_b em B) da rede. Por fim, é criado no nó X uma tabela de roteamento para cada túnel da rede. A tabela referente ao túnel T_a tem como única entrada o endereço IP da interface A_x e a

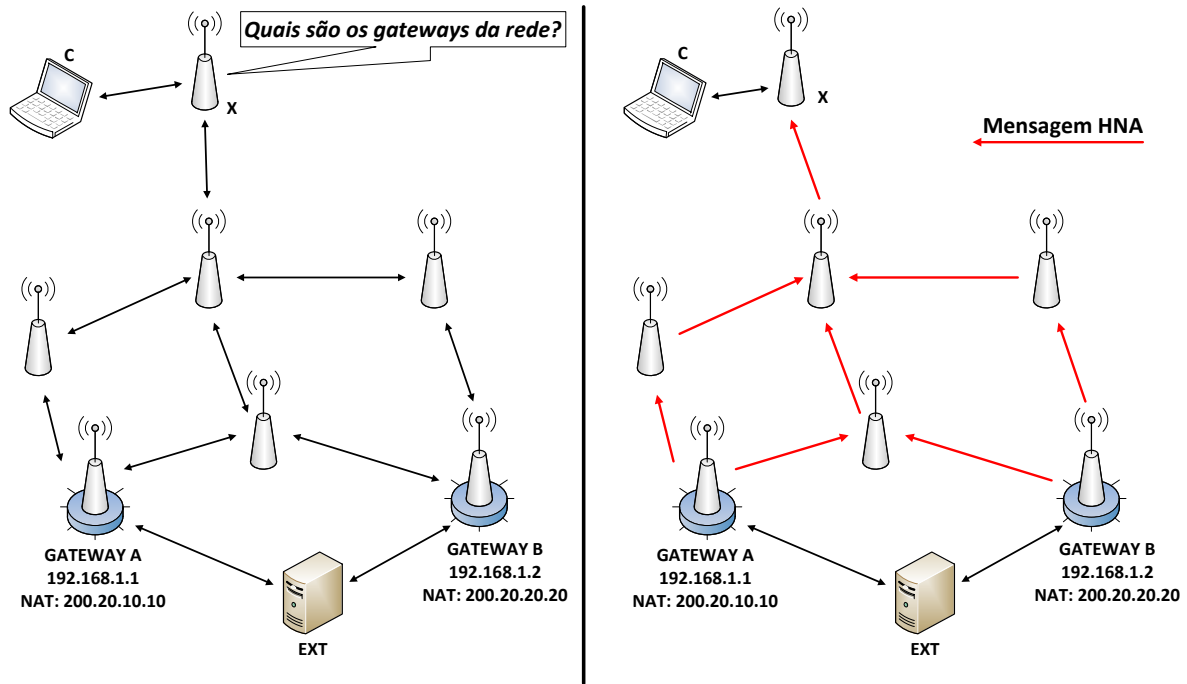


Figura 4.2: Descoberta de *gateways* com o DynTun.

tabela referente ao túnel T_b tem como única entrada o endereço IP da interface B_x .

Quando o nó identifica uma nova conexão do cliente que está conectado a ele, o nó verifica nesse momento qual é o *gateway* de melhor qualidade¹. E então ele modifica a regra de *firewall* (que é responsável pela marcação dos pacotes) do roteador, adicionando uma entrada na qual todos os pacotes desta conexão receberão a mesma marcação nos seus cabeçalhos. Essa marca identifica qual foi o *gateway* de melhor qualidade no momento do estabelecimento da conexão. Quando um pacote dessa conexão chega, a partir do cliente, no nó, ele será identificado. O pacote é encapsulado em outro pacote IP e recebe a marca referente a essa conexão. Todos os pacotes com a marca de um *gateway* trafegarão pelo túnel lógico correspondente a esse *gateway*. Essa definição de qual pacote trafega por qual túnel é feita nas tabelas de roteamento criadas com o DynTun no nó que está executando o mecanismo.

Continuando o exemplo, quando X identifica uma nova conexão proveniente de C para o servidor EXT , X verifica qual é o *gateway* de melhor qualidade no momento do estabelecimento dessa conexão. No exemplo, o *gateway* A tem o caminho com menor custo. Assim, o nó X determina que todos os pacotes dessa conexão receberão a marca

¹No restante deste trabalho, o termo “qualidade de um *gateway*” é usado para se referir de forma abreviada ao custo, de acordo com a métrica de roteamento, do caminho entre um cliente e um *gateway*. Portanto, *gateways* de melhor qualidade são aqueles que possuem caminhos de menor custo.

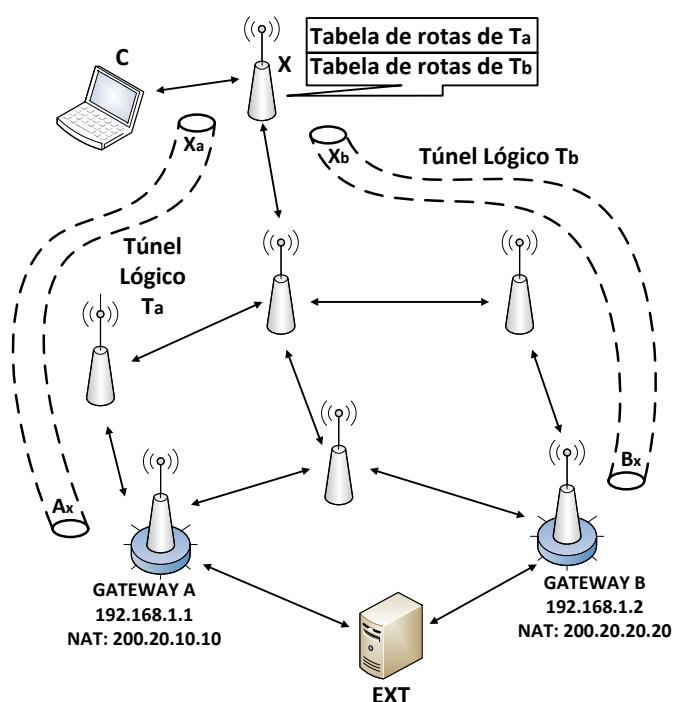


Figura 4.3: Criação de tabelas e túneis lógicos com o DynTun.

referente ao túnel T_a . Pacotes subsequentes dessa conexão receberão a mesma marca e trafegarão pelo mesmo túnel.

Não está exemplificado nas figuras, mas uma suposta segunda conexão poderia usar o *gateway* B através do túnel T_b , caso no momento do seu estabelecimento, o *gateway* B tivesse melhor qualidade do que o *gateway* A. Caso o *gateway* B fosse melhor que A, todos os pacotes da primeira conexão teriam a marcação do túnel T_a e sairiam da rede pelo *gateway* A, enquanto os pacotes da segunda conexão teriam a marcação do túnel T_b e sairiam da rede pelo *gateway* B.

4.2 Motivação para o Desenvolvimento do QuAGS

Durante o estudo do DynTun, foi verificado o potencial do uso do DynTun para realização de balanceamento de carga na rede. O desejo de aproveitar o potencial do DynTun levou à criação do mecanismo QuAGS. Essa seção aborda os principais aspectos do DynTun quanto ao seu potencial e ao seu problema e serve de motivação para o uso do QuAGS.

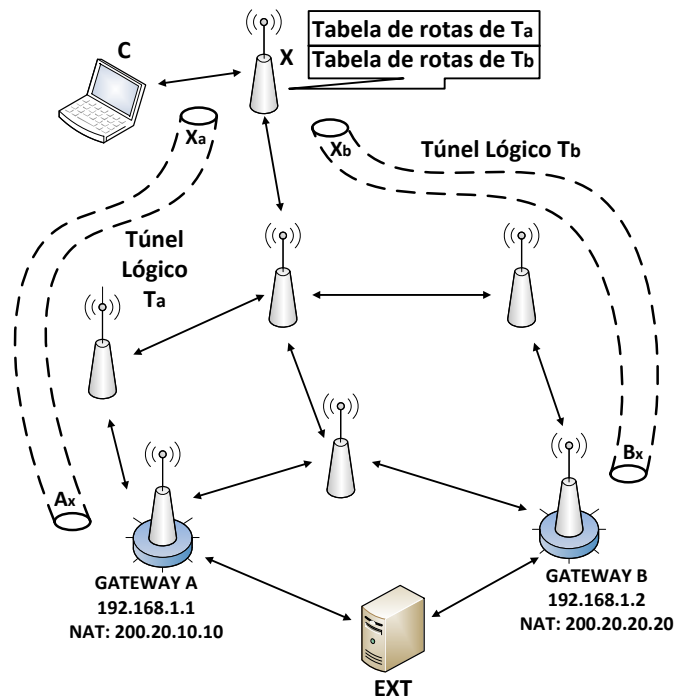


Figura 4.4: Seleção de *gateway* e marcação de pacotes com o DynTun.

4.2.1 Potencial do DynTun

Analisando o DynTun, pode-se perceber o potencial do uso dessa ferramenta para prover o balanceamento de carga em redes em malha sem fio. Com o DynTun, se uma conexão com um servidor externo é aberta por um cliente da rede em malha através de um *gateway*, essa conexão sempre usa esse mesmo *gateway* enquanto estiver ativa. Com o uso desta rota entre o cliente e o *gateway*, a qualidade da mesma pode cair, como retrata a Figura 4.5. Essa figura mostra a variação da qualidade dos melhores caminhos de um cliente para dois *gateways* diferentes em função do tempo. Essas curvas foram obtidas na rede real utilizada nos testes deste trabalho. As quedas observadas na figura ocorrem durante a atividade das conexões durante a realização de um dos testes apresentados no Capítulo 5.

O fenômeno se deve ao fato do tráfego de dados competir com os pacotes de controle utilizados para avaliar a qualidade dos enlaces, em especial, no caso de protocolos de roteamento que realizam medidas ativas de qualidade [6]. Com o DynTun, mesmo que o *gateway* de melhor qualidade se altere durante o tempo de atividade de uma conexão, a ferramenta irá utilizar o *gateway* originalmente selecionado para essa conexão. Outras conexões abertas neste período, no entanto, poderão selecionar outros *gateways* de melhor qualidade. Portanto, o DynTun realiza de forma colateral o balanceamento de carga,

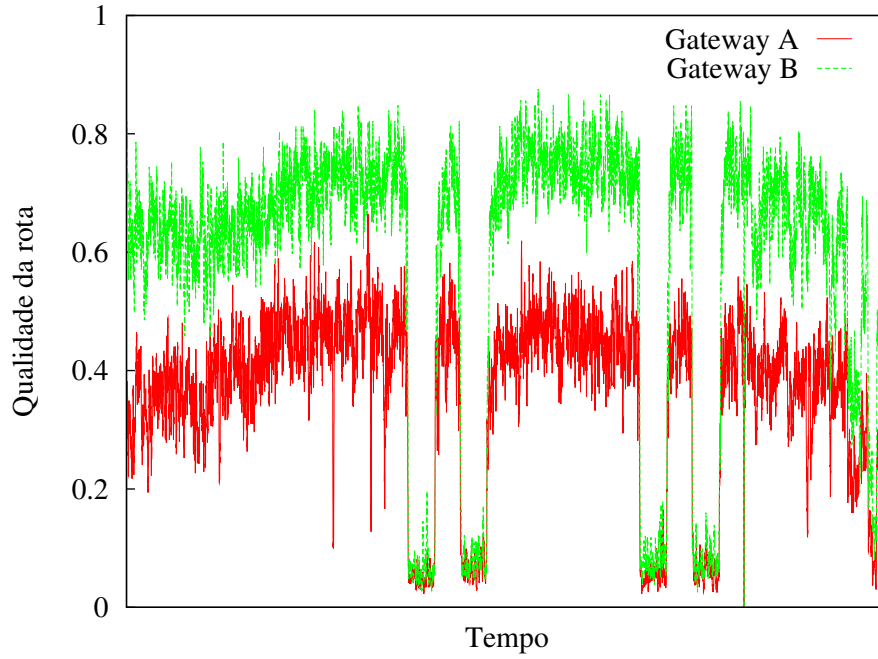


Figura 4.5: Qualidade dos melhores caminhos entre um nó e os *gateways* da rede de testes.

mesmo não sendo o seu objetivo principal. Para demonstrar esse potencial, é apresentado o seguinte exemplo. A Figura 4.6 apresenta uma topologia simples no qual o *backbone* da rede em malha é composto por cinco nós, sendo A e B *gateways*. O cliente C está conectado ao nó Y e EXT é um servidor externo à rede, que se comunica com os dois *gateways*.

Supondo que a rota de melhor caminho entre o cliente C e o servidor EXT seja $C \rightarrow Y \rightarrow X \rightarrow A \rightarrow EXT$, uma nova conexão do cliente C usará o *gateway* A para se comunicar com o servidor. Com o DynTun, todos os pacotes desta conexão usarão essa rota para chegarem ao servidor. Conforme a Figura 4.7 apresenta, com o tráfego no sentido do *gateway* A, acontece a disputa entre os pacotes de controle do protocolo de roteamento com os pacotes de dados e também ocorre a concorrência pelo meio através do CSMA/CA. Por consequência, os enlaces sem fio do caminho da conexão, $Y \rightarrow X$ e $X \rightarrow A$, provavelmente terão uma diminuição na qualidade do caminho.

Assim, com o passar do tempo, o protocolo de roteamento, realizando medidas ativas, pode verificar que a qualidade da rota entre Y e B pode estar superior à rota entre Y e A. Logo, por saturação do caminho $Y \rightarrow A$, com o uso do DynTun, as próximas conexões poderão passar a usar o *gateway* B a partir do momento que a rota através dele seja superior à rota através de A, conforme a Figura 4.8 apresenta.

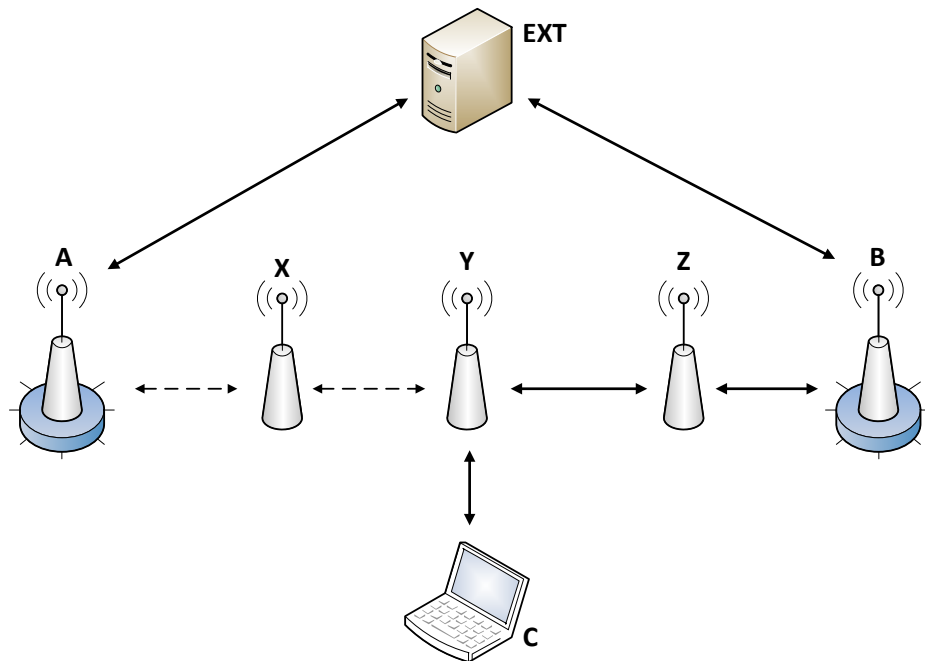


Figura 4.6: Topologia para o exemplo do potencial do DynTun.

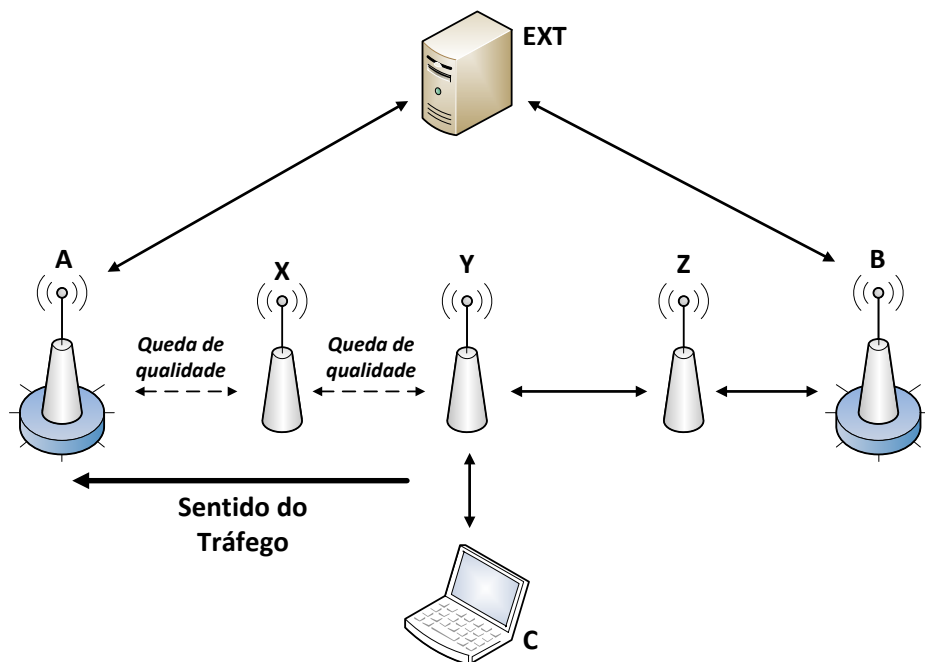


Figura 4.7: Queda da qualidade dos enlaces por causa de tráfego na rede.

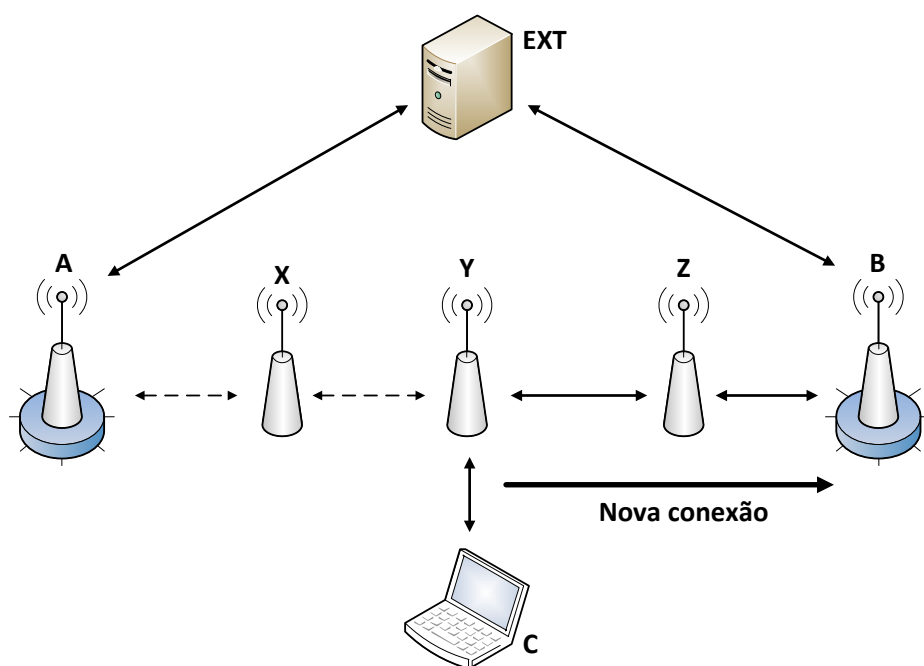


Figura 4.8: Troca da escolha do melhor *gateway* com o DynTun.

Por causa dessa mudança de qualidade dos caminhos até os *gateways*, percebe-se que o balanceamento de carga com o DynTun é realizado de forma colateral. Se a situação ilustrada no exemplo continuar, com o tempo a métrica do caminho $Y \rightarrow B$ provavelmente diminuirá, enquanto a métrica do caminho $Y \rightarrow A$ estará se recuperando da carga anterior e provavelmente voltará a aumentar. Assim, o cenário se inverte e quando a métrica do caminho $Y \rightarrow A$ for maior que a métrica do caminho $Y \rightarrow B$, o *gateway* A novamente será o melhor candidato para novas conexões do cliente C. A repetição da queda e recuperação das métricas para os *gateways* fazem o DynTun realizar o balanceamento de carga na rede.

4.2.2 Limitação do DynTun

Como visto na seção anterior, com o DynTun, um nó sempre seleciona o *gateway* com o caminho de maior qualidade no instante de abertura da conexão de um cliente. Essa característica é herdada do protocolo de roteamento em uso na rede. Porém, é possível que *gateways* com caminhos de menor qualidade nunca sejam selecionados por um cliente, o que reduz a eficiência do uso de múltiplos *gateways*, caso esses *gateways* nunca se tornem o melhor candidato para uma nova conexão. Nesse cenário, tem-se um menor aproveitamento dos recursos da rede. O QuAGS tem como um dos seus objetivos, então,

aproveitar de forma mais eficiente os *gateways* da rede.

Também existe a possibilidade de saturação de um subconjunto dos *gateways*. O exemplo da Seção 4.2.1 mostra que a troca do melhor *gateway* só ocorre no DynTun porque ele causa a saturação na rota entre o nó que executa o mecanismo e o *gateway* de melhor qualidade. Esse é um problema que o QuAGS busca evitar. É visível que, caso a carga da rede seja alta, evitar a saturação dos *gateways* se torna uma tarefa difícil, mas o QuAGS busca evitar a saturação dos *gateways*.

4.3 Proposta: Seleção Probabilística dos *Gateways*

Em geral, usando apenas a métrica de roteamento, um cliente seleciona um *gateway* de acordo com o custo do caminho para alcançá-lo. Nessa situação, é possível que apenas poucos *gateways* sejam usados pelas conexões de saída ou ainda que, durante o seu período de atividade, uma conexão experimente uma mudança de *gateway*. Essa dissertação propõe o mecanismo *Quality-Aware Gateway Selection* (QuAGS) com o objetivo de solucionar ambos os problemas.

O QuAGS difere do DynTun por buscar realizar um balanceamento de carga de forma pró-ativa. O mecanismo usa como base a ferramenta DynTun para se aproveitar da solução eficiente para o problema de quebra da semântica das conexões TCP. O QuAGS, entretanto, utiliza a qualidade dos enlaces e dos caminhos para calcular a probabilidade de seleção de um *gateway* e, assim, prover o balanceamento de carga entre os múltiplos *gateways* de uma rede em malha sem fio.

Para lidar com as mudanças de *gateways*, o QuAGS usa a ferramenta DynTun e, assim, todos os pacotes de uma conexão terão a marca do *gateway* escolhido e serão roteados por um túnel lógico até ele. Para prover o balanceamento de carga, o QuAGS define um valor para a probabilidade de seleção associada a cada *gateway*. Com base nesse valor, determina-se qual dos N *gateways* da rede será usado por essa conexão. A probabilidade de seleção de um *gateway* g_j por um cliente c_i é calculada localmente pelo cliente de acordo com qualidade de g_j . Supondo que q_{c_i, g_j} é a qualidade do *gateway* g_j para o cliente c_i , a probabilidade P_{c_i, g_j} de c_i selecionar g_j é dada por:

$$P_{c_i, g_j} = \frac{q_{c_i, g_j}}{\sum_{k=1}^N q_{c_i, g_k}}. \quad (4.1)$$

Assim, quanto maior a qualidade do caminho para um dado *gateway*, maior a probabilidade que esse *gateway* seja escolhido no instante do estabelecimento de uma nova conexão. Logo, a carga da rede é balanceada de acordo com a qualidade dos *gateways*. Em geral, um cliente terá a maioria de suas conexões roteadas pelo seu *gateway* de melhor qualidade, que geralmente é aquele mais próximo geograficamente. Porém, algumas conexões deste cliente também usarão outros *gateways*. Consequentemente, os recursos da rede são usados de forma mais eficiente. Estatisticamente, os *gateways* mais bem localizados receberão mais carga. Porém, os *gateways* com localização (ou enlaces) piores não serão subutilizados, visto que uma porcentagem de conexões podem ser roteadas por eles.

É importante ressaltar que o cálculo da qualidade do caminho para um dado *gateway* q_{c_i, g_k} depende da métrica usada na rede. Com o passar do tempo, é possível que a qualidade dos enlaces dos *gateways* de melhor qualidade diminua, pois, como dito anteriormente, o tráfego de dados compete com os pacotes de controle usados para medir a qualidade dos enlaces. Logo, a probabilidade de seleção desses *gateways* para novas conexões diminui. Assim, os outros *gateways* terão sua probabilidade incrementada. Ainda assim, nenhum *gateway* é subutilizado, apenas as suas probabilidades de escolha mudam de acordo com a carga da rede. O valor da probabilidade de cada *gateway* é recalculado periodicamente para que as mudanças da métrica dos enlaces se reflitam nas escolhas dos *gateways*. Para atenuar variações, em um dado instante de atualização t , o valor da probabilidade de seleção atual $P_{c_i, g_j}(t - 1)$, que foi observado em um instante de atualização anterior $t - 1$, será substituído pelo valor t observado em $P_{c_i, g_j}(t)$ se, e somente se:

$$|P_{c_i, g_j}(t) - P_{c_i, g_j}(t - 1)| > \varepsilon, \quad (4.2)$$

onde ε é o erro aceitável do cálculo da probabilidade. Do contrário, $P_{c_i, g_j}(t) = P_{c_i, g_j}(t - 1)$. O parâmetro ε é uma constante que pode ser modificada de acordo com a sensibilidade desejada para a variação do cálculo da probabilidade de seleção.

4.4 Algoritmo da Proposta

O QuAGS executa um algoritmo de forma cíclica. Enquanto o protocolo de roteamento estiver funcionando, assim também estará o algoritmo. A Figura 4.9 apresenta a máquina de estados do algoritmo do QuAGS.

Ao receber pela primeira vez a lista de *gateways* da rede e a qualidade da rota entre o nó e cada um deles, o QuAGS realiza o cálculo das probabilidades segundo a Equação 4.1.

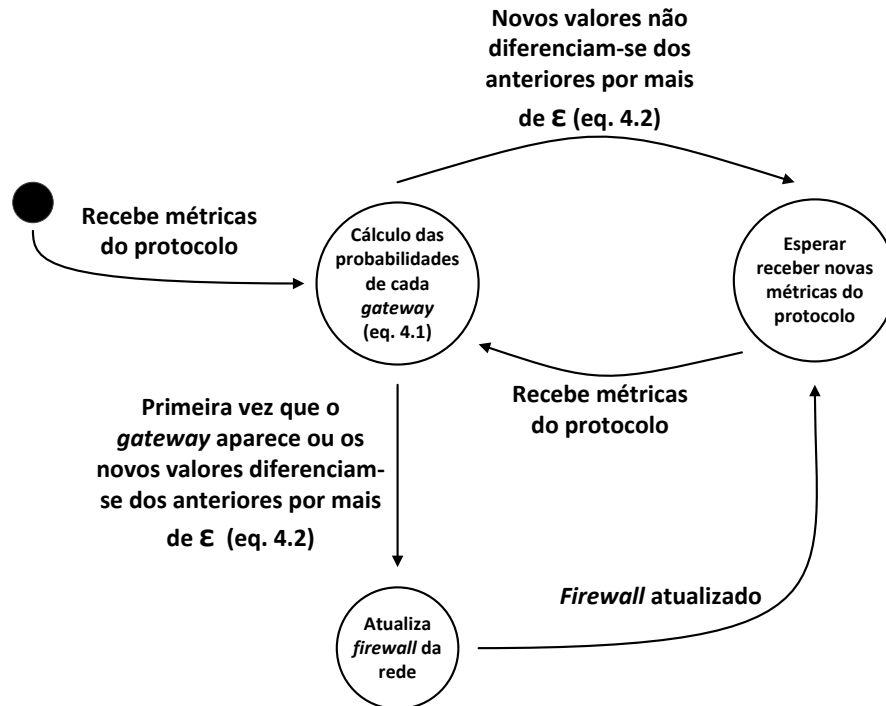


Figura 4.9: Máquina de estados do QuAGS.

Como é a primeira vez que os *gateways* aparecem para o mecanismo, eles são “cadastrados” no *firewall* da rede. Esse “cadastro” se dá por meio da inserção das regras de marcação das conexões no *firewall*, que são apresentadas na Seção 4.5, além da criação das interfaces virtuais no nó, uma por *gateway* cadastrado. Ainda são criadas, para cada *gateway*, as tabelas de roteamento contendo apenas a entrada do *gateway*, de forma semelhante à feita pelo DynTun, como explicado na Seção 4.1.

Após o “cadastro” do *gateway* ser realizado, o algoritmo entra em estado de espera por novos valores de métrica do protocolo de roteamento. O responsável pela escolha do *gateway* para uma nova conexão e marcação de todos os pacotes dessa conexão são as regras do *firewall*. Ou seja, a função do QuAGS é atualizar o *firewall* de acordo com as métricas dos caminhos entre o nó e cada *gateway*, além de criar toda a estrutura de túneis e interfaces necessária para que os pacotes marcados pelo *firewall* possam trafegar.

O QuAGS permanece no estado de espera até que novas métricas sejam enviadas pelo protocolo de roteamento. Caso haja algum novo *gateway* na lista, algum *gateway* saia da rede ou o novo valor da métrica de pelo menos um dos *gateways* seja diferente do valor antigo por ϵ (Equação 4.2), o algoritmo entra novamente na fase de atualização do *firewall*. Nessa nova fase de atualização do *firewall*:

- *Gateways* novos são “cadastrados”, segundo o processo descrito no parágrafo anterior.
- As regras e estruturas (tabela de roteamento, túneis e interfaces) referentes a *gateways* que saiam da rede serão removidas.
- Caso o valor da probabilidade de algum *gateway* já cadastrado tenha que ser atualizado (Equação 4.2), as interfaces, túneis e tabelas de roteamento não são modificados. Apenas as regras de *firewall* que realizam o cálculo da probabilidade dos *gateways* são atualizadas, com os novos valores das probabilidades.

No estado de cálculo das probabilidades de cada *gateway*, caso os valores novos não se diferenciem dos antigos por ε e não tenha nenhuma mudança na lista dos *gateways* da rede, o mecanismo retorna ao estado de espera.

4.5 Implementação

O QuAGS é implementado, na validação deste trabalho, como um módulo do protocolo de roteamento OLSR em roteadores *mesh* executando o sistema operacional *OpenWrt* [14]. É importante ressaltar que o QuAGS é independente do protocolo de roteamento usado na rede e a escolha do OLSR é somente para fins de implementação do mecanismo.

Com o QuAGS, um nó do *backbone* da rede, mantém uma lista de *gateways*. Essa lista contém o endereço IP de cada *gateway* e a qualidade do caminho até ele. Para tanto, foi feita uma pequena modificação no protocolo de roteamento OLSR, para que esta lista fosse guardada. Embora esta mudança tenha sido feita sobre o OLSR-ML [28], diferentes protocolos e métricas de roteamento podem ser utilizados, desde que o valor da métrica seja visível ao QuAGS.

4.5.1 Módulos Necessários

Para o funcionamento do QuAGS, é necessária a instalação de alguns módulos da ferramenta *Iptables* [16], que é responsável pela marcação dos pacotes e escolha dos *gateways*. Além disso, também é necessária a instalação da estrutura necessária para a criação dos túneis. Apenas os nós que executam o QuAGS precisam ter esses módulos instalados. Nos *gateways* da rede, precisa-se instalar apenas a parte responsável pelos túneis, já que eles são a outra extremidade dessa estrutura. Os módulos utilizados são:

- *Iproute2* [17]

É instalado através do programa *ip*. Executa como um comando do *Linux* e é o responsável pela criação, modificação e destruição dos túneis. Também é o responsável pela criação das múltiplas tabelas de rotas, uma para cada *gateway* da rede.

- *ip_gre*

É o módulo que permite a criação de túneis do tipo GRE (*Generic Routing Encapsulation*) [11] que o mecanismo utiliza. Esses túneis são construídos através de encapsulamento IP sobre IP. Com esse módulo, o *Iproute2* pode criar túneis GRE através do comando *ip tunnel add _NAME_ mode gre*. Ambos os *gateways* devem ter o módulo *ip_gre* instalado. Esse módulo é nativo do *firmware OpenWrt* que está instalado nos roteadores da rede real usada nos testes deste trabalho.

- *xt_connmark*

É um módulo da ferramenta *Iptables*. É o responsável por gerenciar a marcação dos pacotes para cada conexão. É o responsável pela verificação de qual conexão um pacote pertence. Caso seja um pacote de uma conexão anteriormente marcada e com o *gateway* de destino definido, ele restaura a marca dessa conexão e insere essa marca no pacote. Caso seja o primeiro pacote marcado de uma conexão, ele salva a marca para que os pacotes seguintes dessa conexão possam ter a mesma marca, através do processo de restauração citado anteriormente.

- *xt_MARK*

Também é um módulo da ferramenta *Iptables*. Ele define qual será a marca de cada conexão e salva essa informação para que o módulo *xt_connmark* possas marcar os pacotes.

- *xt_statistic*

Outro módulo da ferramenta *Iptables*. É o responsável pela seleção probabilística dos *gateways* da rede. Ele define uma probabilidade no *firewall* de uma marca ser escolhida. Caso a variável aleatória sorteada esteja dentro da porcentagem, a regra do *firewall* é executada.

4.5.2 Métodos

O QuAGS tem vários métodos no seu código. É escrito na linguagem C e realiza chamadas de sistema através da função *system*. É aproveitada a estrutura de métodos do DynTun,

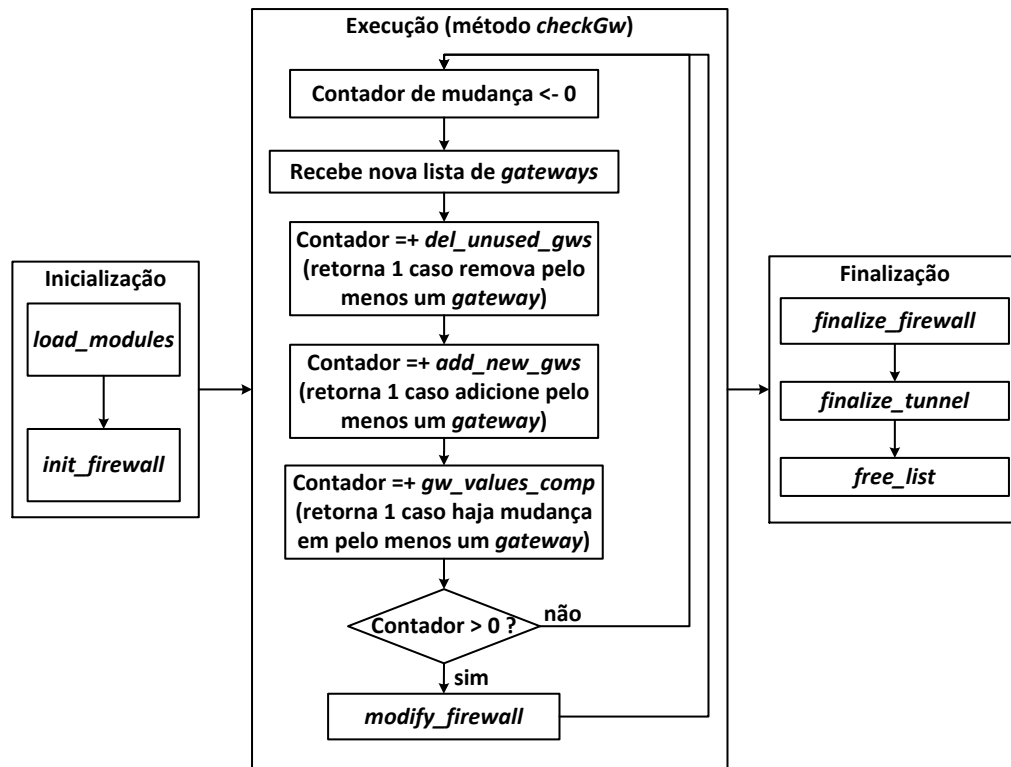


Figura 4.10: Relacionamento entre métodos do QuAGS.

com modificação de alguns e adição de outros. A Figura 4.10 apresenta o relacionamento entre os métodos do QuAGS, na forma que o mecanismo é executado. Os principais métodos estão a seguir:

- Métodos para manipulação da lista de *gateways*

O QuAGS tem um conjunto de métodos que manipula a lista de *gateways*. O método *add_to_list* adiciona um *gateway* à lista, enquanto o *del_from_list* apaga um *gateway* dela. O método *find_in_list* procura um *gateway* na lista e o método *free_list* a apaga.

- Métodos para verificação de mudanças nos *gateways* da rede

O QuAGS utiliza dois métodos que são os responsáveis pela verificação de adição ou remoção de *gateways* na rede, além de um método para verificação de mudanças das métricas (e consequentemente das probabilidades de escolha) dos *gateways* existentes na rede.

O método *add_new_gws* compara a nova lista de *gateways* enviada pelo protocolo de roteamento com a antiga lista de *gateways* da rede. Caso haja *gateways* novos,

o método chama outro método responsável pelo “cadastro” dos novos *gateways* no mecanismo, com a criação dos túneis, interfaces e regras de *firewall* para eles.

O método *del_unused_gws* é o responsável pela ação inversa: ao comparar a nova e a antiga listas de *gateways*, caso algum *gateway* tenha saído da rede, ele invoca outro método responsável pela remoção de toda a estrutura (regras de *firewall*, túneis e interfaces) dos *gateways* que saiam da rede.

O método *gw_values_comp* compara o valor das métricas das listas antiga e nova de *gateways* da rede, segundo a Equação 4.2. Caso alguma probabilidade tenha que ser modificada, todas são recalculadas segundo a Equação 4.1.

- Métodos para manipulação do *firewall*

Esse conjunto de métodos é responsável pela manipulação do *firewall*. São os responsáveis por escrever e modificar o *firewall* da rede, para que ele possa realizar a seleção de *gateways* e marcação de pacotes.

O método *init_firewall*, invocado apenas no início da execução do mecanismo, cria as regras de forma genérica e as escreve no *firewall* da rede. Elas são escritas de forma genérica porque o método *modify_firewall* as sobrescreve com as informações específicas de seleção e marcação dos *gateways* da rede.

O método *modify_firewall* é o método que é chamado pelos métodos *add_new_gws* e *del_unused_gws* para adição e remoção, respectivamente, das regras de *firewall* quando a lista de *gateways* na rede sofre alguma alteração. Esse método também é invocado quando pelos menos uma das métricas dos caminhos $\text{nó} \rightarrow \text{gateways}$ enviada pelo protocolo de roteamento diferencia-se da anterior segundo a Equação 4.2: nenhuma regra é retirada ou adicionada, mas as regras dos *gateways* existentes são modificadas.

O método *finalize_firewall* é invocado apenas quando a execução do mecanismo é terminada. Ele é o responsável por apagar as regras adicionadas pelo mecanismo do *firewall* da rede, retornando-o ao estado anterior à execução do QuAGS.

- Métodos para manipulação de túneis, tabelas de roteamento e interfaces virtuais

Esse conjunto de métodos trata das estruturas necessárias para que os pacotes marcados possam trafegar corretamente pela rede.

O método *add_tunnel_addr* é o método responsável pela criação e adição de toda essa estrutura de túneis, tabelas e interfaces. É invocado sempre que um *gateway* novo é adicionado à lista. Cria uma interface virtual de nome “overX” (onde X é

um identificador único do *gateway*) com a ferramenta de sistema *ifconfig*, além do túnel GRE de mesmo nome com a ferramenta *Iproute2*. Também com o *Iproute2* ele cria a tabela de rotas com a marca do seu *gateway*, contendo apenas a entrada dele. Quando o *firewall* verifica a marca o pacote, é redirecionado para a tabela com a mesma marca. Como essa tabela só tem uma entrada, ela é utilizada pelo pacote, automaticamente redirecionando para o *gateway*.

Os métodos *del_tunnel_addr* e *finalize_tunnel* são responsáveis por remover essa estrutura. O primeiro apaga a tabela de rotas da marca do *gateway*, enquanto o segundo, para cada *gateway*, remove o túnel e derruba a interface virtual criada.

- Método para carregar os módulos necessários para a execução do mecanismo

O método *load_modules* é o responsável por realizar as chamadas de sistema que carregam os módulos descritos na Seção 4.5.1.

- Método de execução do mecanismo

Finalmente, o método *checkGw* é o responsável por executar em repetição as ações que unem os métodos citados acima, após as estruturas serem inicializadas. Primeiro, através do método *del_unused_gws*, ele desfaz os túneis relativos aos *gateways* que não existem mais na rede. Em seguida, ele adiciona os túneis relativos a novos *gateways* com o método *add_new_gws*. O próximo passo é verificar se os valores dos *gateways* existentes sofreram alguma modificação com o método *gw_values_comp*. Os métodos *del_unused_gws*, *add_new_gws* e *gw_values_comp* retornam a informação se houve mudanças na lista de *gateways* da rede e/ou nos seus valores de probabilidade através de uma *flag* de modificação. Caso essa *flag* informe que houve alguma dessas modificações, os métodos de modificação do *firewall* e dos túneis da rede são invocados. Por fim, ele apaga a lista antiga de *gateways* e “envelhece” a lista nova.

Capítulo 5

Avaliação de Desempenho

Este capítulo apresenta os testes e a avaliação de resultados realizados neste trabalho. Esta avaliação teve por objetivo comparar o desempenho do QuAGS com o desempenho do DynTun, analisando diferentes fatores, como por exemplo vazão e percentual de uso de *gateways*.

A avaliação foi dividida em duas partes. A primeira é experimental, realizada em uma rede real, enquanto a outra é baseada em simulações através da ferramenta *NS-2* [13]. O objetivo desta abordagem é utilizar a simulação como uma forma de avaliar o mecanismo em cenários que seriam difíceis de serem configurados na rede real disponível.

As seções deste capítulo descrevem os cenários e resultados obtidos. Os resultados dos experimentos reais são apresentados na Seção 5.1, enquanto os resultados das simulações são apresentados na Seção 5.2. Para todos os gráficos nesse capítulo, são calculados intervalos de confiança para um nível de confiabilidade de 90%.

5.1 Experimentos na Rede Real

Os testes experimentais foram realizados na rede interna do projeto ReMoTE [18], composta por treze roteadores *LinkSys WRT54G* com o *firmware OpenWrt* [14]. Ela se localiza ao longo de dois andares de um dos prédios da Escola de Engenharia da Universidade Federal Fluminense. Os nós da rede usam como protocolo de roteamento o OLSR com a métrica ML [28]. A topologia desta rede é a ilustrada na Figura 5.1.

O objetivo dos primeiros testes é verificar a vazão obtida por um cliente da rede com o uso do QuAGS e do DynTun. Para realizar as medições de vazão, foi utilizada uma ferramenta de geração de fluxo TCP. Um cliente, que gera o tráfego, se conecta por interface cabeada com o Nó 6. Os nós 0 e 12 fazem papel de *gateways* da rede. O

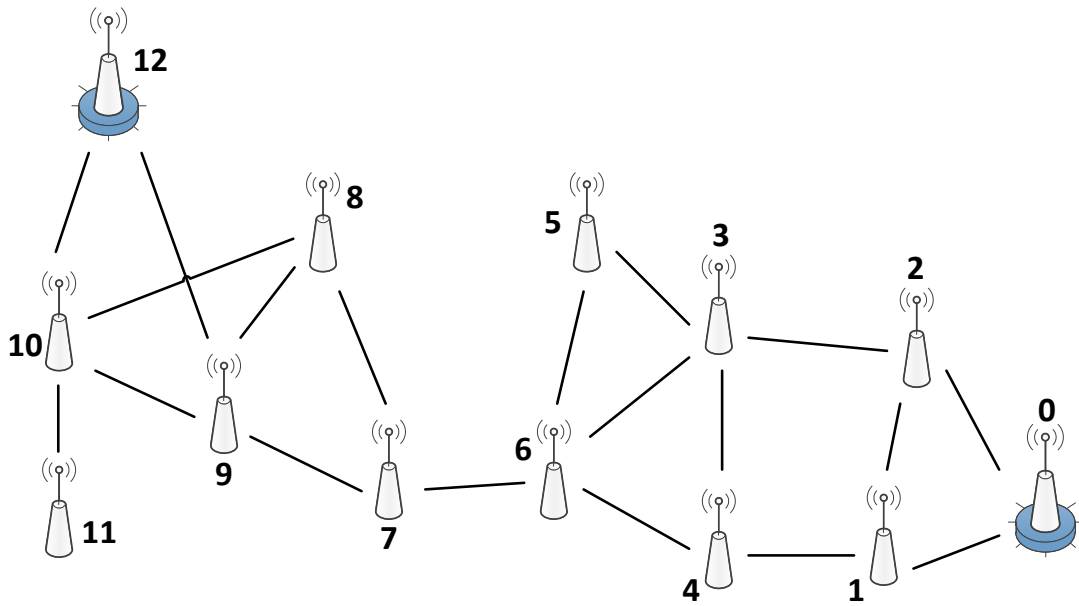


Figura 5.1: Topologia dos experimentos da rede real.

mecanismo de seleção de *gateways* avaliado (DynTun ou QuAGS) é executado no Nó 6. O cliente abre quatro conexões simultâneas, todos com destino a um servidor localizado fora da rede. Cada conexão dura 60 s. Após o término das conexões, espera-se mais 120 s e mais quatro conexões são abertas simultaneamente. Esse passo repete-se um total de 30 vezes. No total, 120 conexões são abertas pelo cliente. Esse período de 120 s entre as rodadas é feito para que as métricas dos enlaces possam ser restabelecidas para os valores típicos da rede. O valor de 120 s é selecionado através do tempo de envio dos pacotes de controlos do protocolo de roteamento da rede.

5.1.1 Resultados Experimentais

A Figura 5.2 apresenta a vazão média das 120 conexões TCP abertas pelo nó origem, com o DynTun e com o QuAGS. A Tabela 5.1 apresenta a porcentagem média de uso dos *gateways* nos testes realizados. No total, foram oito dias de teste, sendo uma rodada de teste por dia, sempre ocorrendo durante o período entre 0h e 6h, onde a rede é sujeita a menos interferência.

Na média, o QuAGS teve uma vazão média de 50,5 kbps enquanto o DynTun teve uma vazão média de 49,4 kbps. Porém, pode-se ver que de acordo com a porcentagem de uso dos *gateways* da rede, o QuAGS faz uma melhor distribuição da carga. Essa é uma

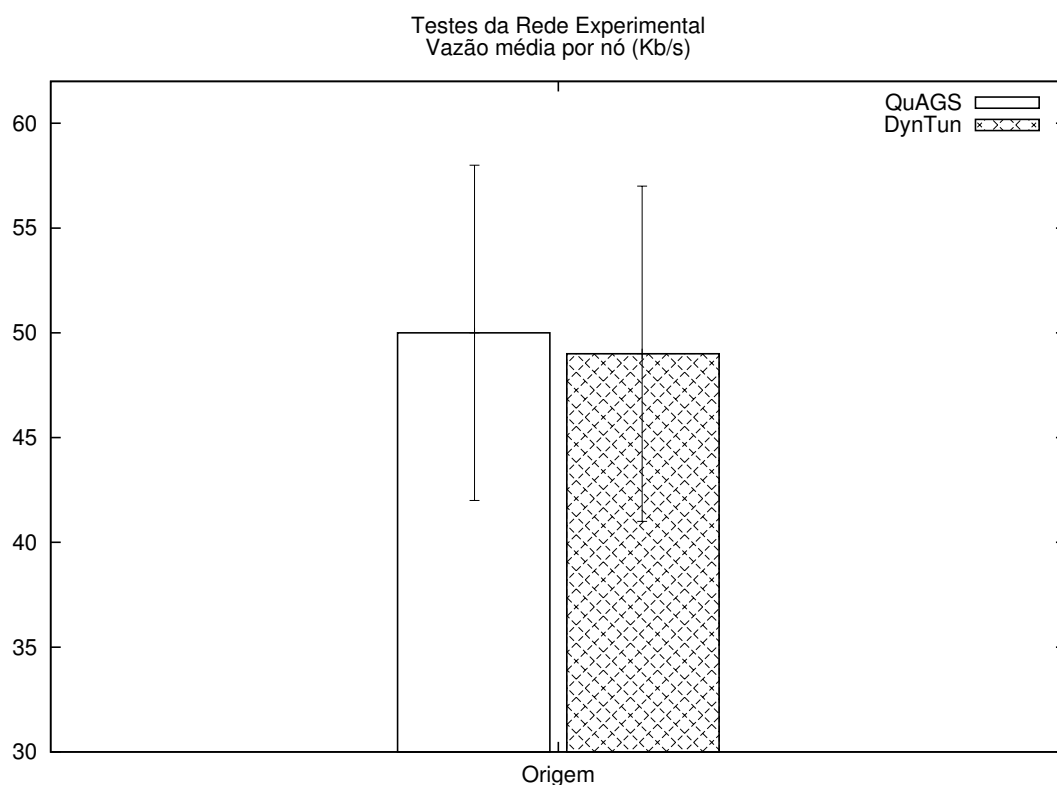


Figura 5.2: Rede experimental - vazão média do nó origem.

Mecanismo	Escolha <i>Gateway</i> 0	Escolha <i>Gateway</i> 12
QuAGS	58%	42%
DynTun	92%	8%

Tabela 5.1: Rede experimental - porcentagem de uso dos *gateways*.

característica desejada em redes onde a capacidade dos *gateways* é limitada. Outro tipo de rede em que se deseja essa característica é uma rede de sensores, na qual o baixo uso de um nó resulta na economia de energia do mesmo, sendo energia um fator preponderante nesse tipo de rede. Assim, percebe-se que o QuAGS consegue utilizar a rede de forma mais balanceada que o DynTun, sem comprometer a vazão da mesma por utilizar caminhos que são de pior qualidade.

Outro dado que aponta o uso balanceado da rede pelo QuAGS é a média de trocas de escolha de *gateway* para conexões consecutivas pelos mecanismo avaliados. Durante os oito dias de testes, o QuAGS apresenta uma média de 25,375 trocas, enquanto o DynTun apresenta uma média de 5 trocas. Assim, o DynTun realiza o envio conexões seguidas para o mesmo *gateway* com mais frequência se comparado ao QuAGS, que resulta na concentração de tráfego na região de um dos *gateways* da rede.

Parâmetro	Valor
Início da primeira rodada	200 s
Duração das conexões	300 s
Conexões abertas por rodada	10
Intervalo entre rodadas	30 s
Número de rodadas	10

Tabela 5.2: Parâmetros utilizados para a modelagem do tráfego nas simulações.

A baixa vazão obtida pelos dois mecanismos pode ser explicada pela capacidade limitada da rede sem fio experimental. Como o número de saltos entre o nó origem e o nó destino tende a diminuir a vazão da mesma, acredita-se que nessa rede experimental específica, com o número de saltos entre o nó escolhido como origem e o nós escolhidos como *gateways*, os testes podem ter chegado a um limite superior de vazão possível.

5.2 Resultados de Simulações

Nas simulações para avaliar o QuAGS, foram idealizadas quatro topologias diferentes. Cada uma das topologias representa diferentes cenários, para que possam ser avaliadas diversas combinações de tráfego de rede, número de nós fonte e número de *gateways*.

Nas simulações, a implementação foi realizada de forma diferente dos testes na rede experimental, a fim de facilitar a modelagem de uma rede com múltiplos *gateways*. Como o DynTun (e consequentemente o QuAGS) certifica que não há quebra de conexões TCP na rede [8], a simplificação foi feita da seguinte forma:

- Os geradores dos fluxos TCP são os nós da rede, ao contrário dos testes na rede experimental, onde quem gera o fluxo são os clientes conectados aos nós;
- O destino dos fluxos TCP não é um servidor externo e sim os *gateways*. Assim, as conexões continuam sendo obrigadas a passar pelos *gateways* da rede. Como o objetivo dos testes não é verificar a taxa de quebra de conexão e sim o mecanismo de escolha dos *gateways*, essa abordagem é válida.

O tipo de fluxo TCP escolhido para as simulações foi o FTP (*File Transfer Protocol*). Para todas as simulações realizadas, os parâmetros utilizados para definição do tráfego nos cenários, assim como seus valores escolhidos, estão apresentados na Tabela 5.2.

O parâmetro *início da primeira rodada* é necessário para que as métricas de roteamento da rede possam se estabilizar. Se as conexões iniciarem no tempo 0 s de simulação,

as métricas estarão com valores que não refletem a situação real da rede e os resultados seriam comprometidos. De acordo com simulações preliminares, foi verificado que em torno de 190 s as métricas da rede se estabilizam. Assim, o valor de 200 s foi escolhido para esse parâmetro.

Uma “rodada” é o nome dado a um grupo de conexões TCP abertas simultaneamente, por cada um dos nós geradores de tráfego na rede. Inicialmente, a ideia do teste era ser realizado de forma semelhante ao teste da rede experimental. Logo, nas simulações preliminares foi definido que uma rodada iniciaria apenas depois que as conexões da rodada anterior terminassem. Porém, foi verificado que as métricas da rede diminuem nas simulações de forma muito mais branda que nos testes na rede experimental, não gerando o impacto desejado no uso do DynTun ou do QuAGS. Então, foi definido que nas simulações, a rede precisaria ter um número maior de conexões simultaneamente para que a métrica pudesse mudar mais rapidamente. Mas aumentar o parâmetro *conexões abertas por rodada* não foi uma solução que trouxe resultado desejado, pois seria necessário um número grande de conexões simultâneas para gerar o efeito desejado. Um grande número de conexões simultâneas aliado a um grande número de rodadas torna a simulação longa, de forma que muito tempo real seria gasto para gerar resultados. A solução encontrada foi “mesclar” as rodadas, de forma que uma rodada não necessariamente precisa a anterior termine para ela poder iniciar. Assim, o efeito desejado aparece e a simulação dura um tempo real razoável. Para esse fim, como as conexões duram 300 s e são 10 rodadas de conexões, foi definido que o *intervalo entre rodadas* é de 30 s.

É importante frisar o fato de que esses valores são locais e iguais para todos os nós que originam tráfego na rede. Logo, seguindo os valores da Tabela 5.2, cada nó origem gera um total de 100 conexões TCP de 300 s de duração. Consequentemente, o total de conexões em uma simulação é o número de nós origem vezes 100.

As seções a seguir descrevem cada topologia usada nas simulações, assim como os cenários configurados em cada uma.

5.2.1 Topologia da Rede Interna

A primeira topologia usada na avaliação dos mecanismos é semelhante à rede real usada nos testes experimentais (Figura 5.1), sendo composta por treze nós.

Para que a topologia pudesse refletir com a maior precisão possível a rede real, foi necessário escolher de um modelo de propagação e ajustar os seus parâmetros. O modelo

Parâmetro	Valor
Expoente de Perda	2,5
Desvio Padrão	1,4
Distância de Referência (m)	0,5

Tabela 5.3: Parâmetros utilizados para o modelo de propagação *Shadowing* nas simulações.

selecionado foi o *Shadowing* [30]. Os modelos mais utilizados normalmente, como o *Free Space* e o *Two-ray Ground*, são baseados em uma função relativa à distância entre os nós. Por outro lado, o *Shadowing* é um modelo estatístico, no qual os nós podem comunicar-se probabilisticamente quando perto da borda da área de comunicação. A potência do sinal recebido em um ponto é modelada como uma variável aleatória de distribuição log-normal, com média calculada a partir de uma distância de referência.

No *NS-2*, para utilizar o modelo de propagação *Shadowing*, é necessário especificar quatro parâmetros:

- o expoente de perda do percurso;
- o desvio padrão da distribuição log-normal (em dB);
- a distância de referência (em metros);
- e a semente para a geração de números pseudoaleatórios.

A Tabela 5.3 lista os valores escolhidos, a fim de que a topologia pudesse estar o mais próximo possível da rede experimental, para os três primeiros parâmetros. O quarto parâmetro (a semente) foi variado de 0 a 9 nas simulações realizadas.

Para essa topologia, dois cenários diferentes foram idealizados, cada um com um número diferente de nós fonte:

- Um no qual apenas um nó é o gerador de tráfego TCP, o nó 6, que está localizado geograficamente no centro da rede.
- Um com três nós que geram o tráfego TCP: além do nó 6, dois nós deslocados para cada uma das pontas da rede (nós 4 e 7). Em ambos os cenários, dois nós destino (que fazem o papel dos *gateways*) são especificados, que são os nós 0 e 12. Eles foram escolhidos por serem os nós *gateways* da rede experimental.

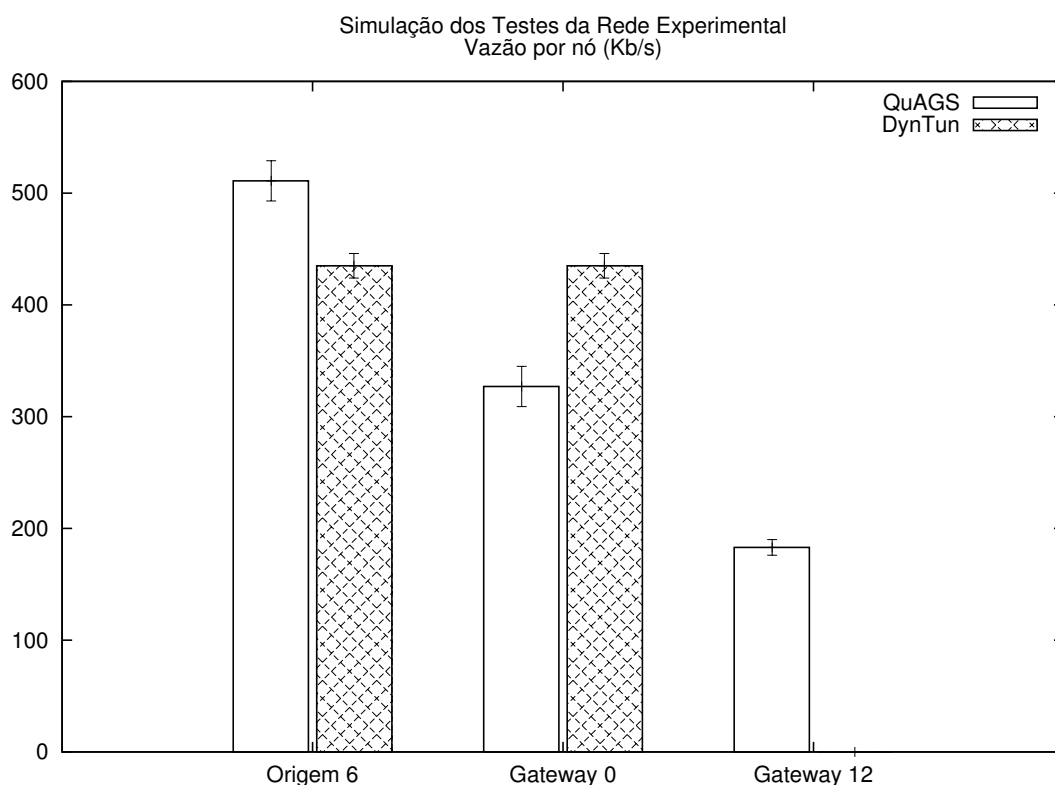


Figura 5.3: Simulação da rede experimental, cenário com uma origem - vazão média por nó.

Mecanismo	Escolha <i>Gateway 0</i>	Escolha <i>Gateway 12</i>
QuAGS	60%	40%
DynTun	100%	0%

Tabela 5.4: Simulação da rede experimental, cenário com uma origem - porcentagem de uso dos *gateways*.

A Figura 5.3 apresenta a vazão média das 100 conexões TCP abertas no cenário com apenas um nó origem, com o DynTun e com o QuAGS, nas dez simulações. A Tabela 5.4 apresenta a porcentagem média de uso dos *gateways* nos testes realizados.

Em comparação com o resultado da rede experimental, pode-se primeiramente verificar que a vazão média da rede, utilizando DynTun ou QuAGS, é de 9 a 10 vezes superior nas simulações. O QuAGS teve uma vazão média de 511,5 kbps, enquanto o DynTun teve a vazão média de 435,8 kbps. Isso significa um ganho de 17% na vazão da rede. Enquanto o DynTun utilizou, em todas as sementes, apenas um dos *gateways* da rede, o QuAGS fez uma distribuição da carga entre eles.

Esse resultado demonstra que, em condições de rede ideais, o ganho do QuAGS sobre

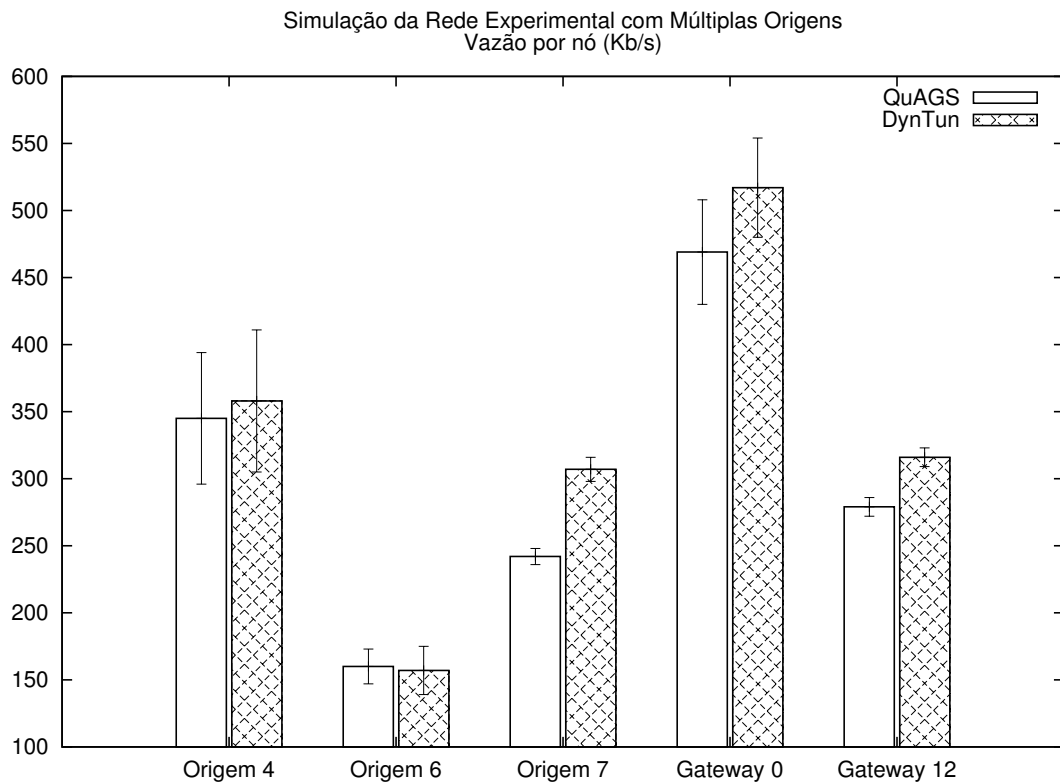


Figura 5.4: Simulação da rede experimental, cenário com três origens - vazão média por nó.

o DynTun tende a ser maior, pois na rede experimental a vazão da rede em geral para ambos é baixa. O uso de caminhos diferentes para as conexões da rede mostram que, além da potencial economia de energia do nó e de alívio de sua capacidade, o QuAGS também pode aumentar a vazão agregada da rede.

Porém, é necessário verificar a tendência de comportamento do QuAGS em uma rede com múltiplas fontes. Os resultados a seguir referem-se ao cenário com três nós origem e dois *gateways* na rede. A Figura 5.4 apresenta a vazão média das 100 conexões TCP abertas no cenário em cada nó de origem e cada *gateway*, com o DynTun e com o QuAGS, nas dez simulações. A Tabela 5.5 apresenta a porcentagem média de uso dos *gateways* nos testes realizados, para cada nó origem da rede.

O Nó 4, localizado perto do *Gateway* 0, teve uma vazão média de 345,2 kbps com o QuAGS e 358,5 com o DynTun. O Nó 6, geograficamente central na rede, obteve resultados de vazão de 160,7 kbps e 157,4 com o QuAGS e DynTun, respectivamente. E o Nó 7, localizado perto do *Gateway* 12, teve uma vazão média de 242,6 com o QuAGS e de 307,1 com o DynTun. Nesse cenário, percebe-se que o uso do QuAGS não é recomendável para nós que estejam geograficamente localizados perto de um dos *gateways* da rede. A

Mecanismo (nó origem)	Escolha <i>Gateway</i> 0	Escolha <i>Gateway</i> 12
QuAGS (4)	72%	28%
DynTun (4)	100%	0%
QuAGS (6)	56%	44%
DynTun (6)	99%	1%
QuAGS (7)	42%	58%
DynTun (7)	9%	91%

Tabela 5.5: Simulação da rede experimental, cenário com três origens - porcentagem de uso dos *gateways*.

característica do QuAGS de fazer o uso de acordo com as métricas dos diferentes *gateways* da rede nesses nós é o responsável por isso. Como na simulação as métricas são mais estáveis, a qualidade dos caminhos para os *gateways* da rede tende a ser de uma ordem de grandeza semelhante. O uso desta estatística no QuAGS faz com que o mecanismo enxergue que a qualidade dos caminhos seja de uma ordem de grandeza semelhante, o que pode ser verificado na Tabela 5.5. Assim, para algumas conexões, o QuAGS faz com que os nós em uma extremidade da rede utilizem os *gateways* que estão localizados geograficamente longe deles. Nestes casos, o uso de caminhos com mais saltos resulta na queda da vazão. Para esses nós, recomenda-se o uso do DynTun, por ser um mecanismo que tem uma probabilidade menor de utilizar um *gateway* nessas condições.

Para o Nó 6, os resultados assemelham-se aos resultados do teste na rede experimental com apenas um nó de origem. O tráfego intenso nas extremidades da rede, nas regiões de ambos os *gateways*, limita a vazão máxima do Nó 6, pois ele realiza mais saltos e mais disputas pelo meio. Assim, o ganho com o uso do QuAGS é limitado pela vazão máxima desse nó nas condições do teste.

5.2.2 Topologia Linear

Outra topologia utilizada na avaliação da proposta foi a topologia linear. Nessa topologia, 15 nós são dispostos em linha, com 20 metros de distância um do outro. O raio de alcance de cada nó é suficiente para comunicar-se apenas com os vizinhos mais próximos. Essa topologia está ilustrada na Figura 5.5. Esse tipo de topologia foi escolhido por ser uma rede esparsa e com poucas opções de escolha de rotas.

O cenário para essa topologia contém um nó origem e dois *gateways*. O nó escolhido para ser origem é o 1, que é o nó geograficamente central da rede. Ele se encontra a sete saltos dos dois *gateways* escolhidos, que são os nós 0 e 2. O objetivo da avaliação

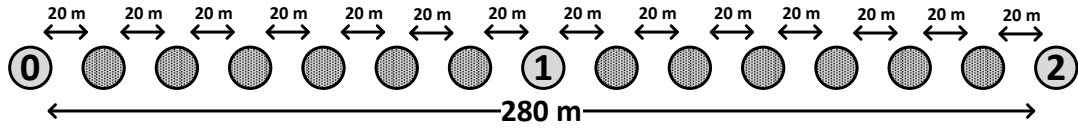


Figura 5.5: Topologia para simulações: topologia linear com 15 nós.

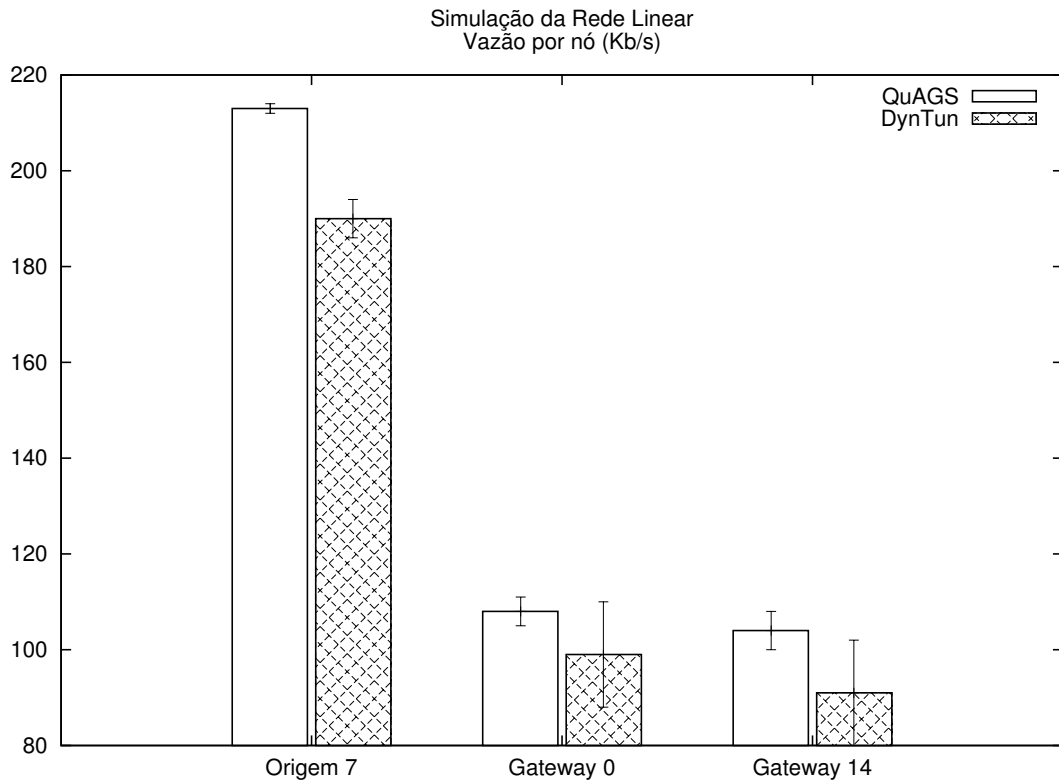


Figura 5.6: Simulação da rede linear - vazão média por nó.

é verificar o desempenho do QuAGS em uma rede esparsa, onde os caminhos para os dois *gateways* podem ter qualidade semelhante. Os parâmetros do modelo de propagação *Shadowing* utilizados nas outras topologias são mantidos para esta topologia.

A Figura 5.6 apresenta a vazão média das 100 conexões TCP abertas pelo nó origem e em cada *gateway*, com o DynTun e com o QuAGS, nas dez simulações.

Na média, o QuAGS teve uma vazão média de 213,5 kbps, enquanto o DynTun teve uma vazão média de 190,9 kbps. Nesse resultado confirma-se a tendência do QuAGS ter resultado superior em nós que estão geograficamente localizados no centro da rede. Com a vazão da rede maior se comparado aos resultados das figuras 5.2 e 5.4, o ganho do QuAGS é visível, nesse caso de 12%, aproximando-se do ganho obtido na avaliação experimental,

Mecanismo	Escolha <i>Gateway 0</i>	Escolha <i>Gateway 2</i>
QuAGS	49%	51%
DynTun	54%	46%

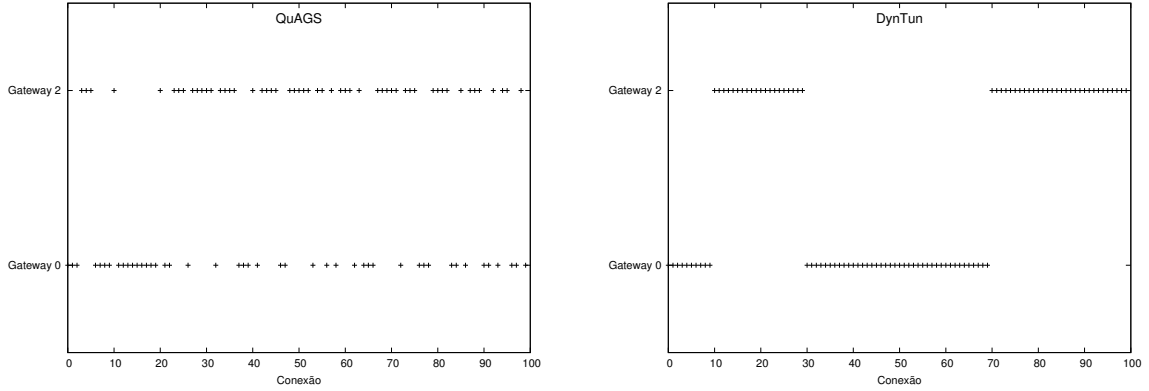
Tabela 5.6: Simulação da rede linear - porcentagem de uso dos *gateways*.Figura 5.7: Simulação da rede linear - histórico de uso dos *gateways*.

Figura 5.3. Apesar de serem topologias diferentes, as características de localização do nó origem e dos *gateways* é semelhante.

Um resultado interessante é o da Tabela 5.6, que apresenta a porcentagem média de uso dos *gateways* nos testes realizados. Em algumas das sementes de aleatoriedade, o DynTun realiza o balanceamento de carga da rede em 50-50%, enquanto o QuAGS apresenta algo em torno de 55% para o *gateway* mais utilizado. Esse resultado demonstra que o uso, em número absoluto, totalmente igualitário da rede não é sinônimo de balanceamento de carga realizado de forma ideal. O mais importante é como a carga é distribuída ao longo do tempo. A Figura 5.7 apresenta a seleção de *gateways* ao longo do tempo de simulação pelo DynTun e pelo QuAGS na semente 1, onde o uso dos *gateways* no DynTun e no QuAGS são, respectivamente, 50-50% e 56-44% (as porcentagens são apresentadas na forma “*gateway 0* - *gateway 2*”).

Apesar do DynTun, no final, realizar o balanceamento igualitário, a escolha do *gateway* é feita de forma rígida, ao ponto da escolha do *gateway* modificar-se apenas quando ocorre a saturação do caminho até o *gateway* utilizado anteriormente. Já com o QuAGS, a carga é balanceada de forma suave, sem saturar um *gateway* da rede. A Figura 5.8 mostra a comparação, em termos de vazão, entre QuAGS e DynTun no intervalo temporal de 30 s antes das trocas (saturação) de *gateway* no DynTun, em todas as sementes testadas. Pode-se perceber que na região crítica, próxima a saturação do *gateway*, o QuAGS tem um

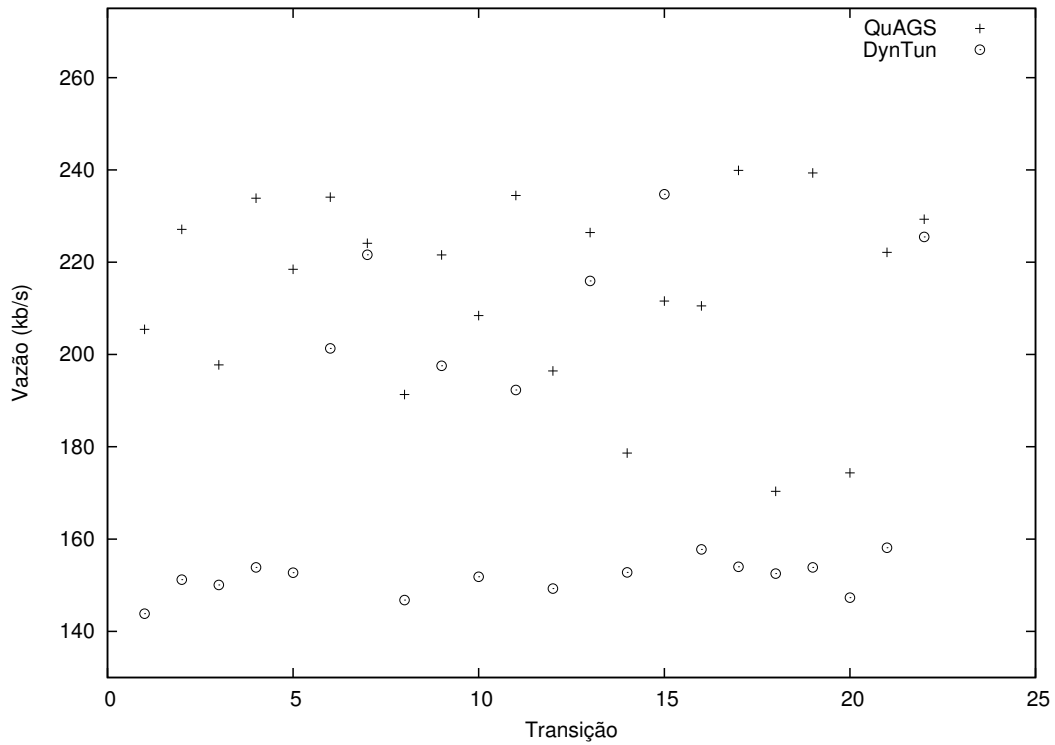


Figura 5.8: Testes da Rede Linear - Vazão da rede no intervalo de 30s antes de cada transição no DynTun

ganho significativo de vazão, se comparado ao DynTun. Essa baixa na vazão do DynTun se deve ao fato da alta queda da métrica nos enlaces, antes da realização de trocas de escolha de *gateway* para conexões consecutivas. Assim, é apresentada a importância de evitar a saturação de um *gateway* com o QuAGS.

5.2.3 Topologias em Grade

Outro tipo de topologia utilizada na avaliação da proposta foi a topologia em grade. Esse tipo de topologia foi escolhido por ser uma rede densa, com diversas possíveis rotas entre os nós da rede. O principal objetivo da avaliação é verificar o desempenho do QuAGS em uma rede com um número maior de nós, em relação às outras topologias apresentadas. Como os parâmetros do modelo de propagação *Shadowing* utilizados nas outras topologias foram calibrados a partir de uma rede real, os mesmos foram mantidos para esta topologia. Duas topologias em grade diferentes foram idealizadas para a avaliação.

A primeira topologia em grade, ilustrada na Figura 5.9, é formada por 49 nós dispostos em uma matriz 7x7. Cada lado da matriz tem 120 m. Assim, cada nó se distancia 20 m dos seus vizinhos próximos de linha ou coluna. Para essa topologia, três cenários foram

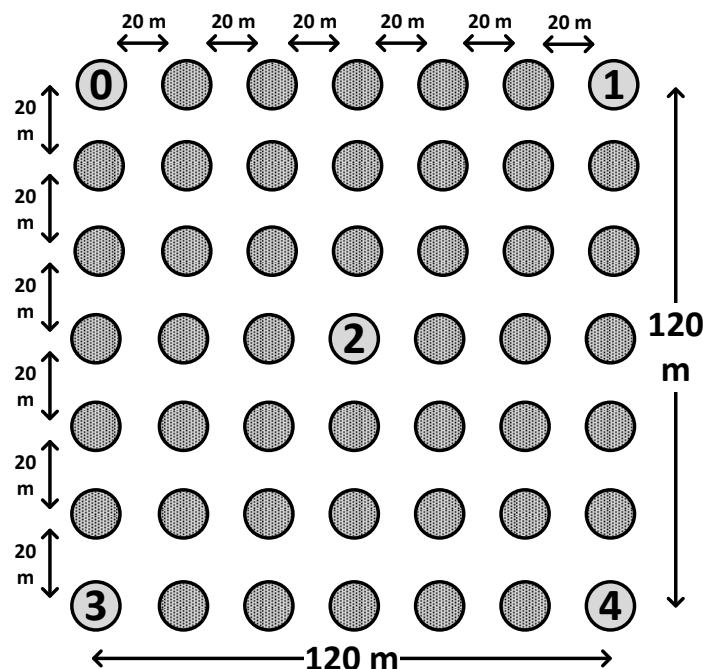


Figura 5.9: Topologia para simulações: topologia em grade 7x7.

idealizados para realizar a avaliação:

- O primeiro contém apenas um nó fonte, que é o Nó 2. Ele foi escolhido por ser o nó central da rede. Nesse cenário, quatro *gateways* foram escolhidos, os nós 0, 1, 3 e 4. Eles foram escolhidos por serem os nós das extremidades da rede;
- O segundo, assim como o primeiro, tem apenas um nó fonte. Mas para esse cenário o nó fonte escolhido foi o Nó 0. O objetivo desse cenário é fazer que o tráfego saia de um dos vértices da rede para os outros vértices. Para isso, três nós foram designados como *gateways*, que são os nós 1, 3 e 4. Foram escolhidos por serem os outros vértices da rede.
- O terceiro cenário tem dois nós fonte e dois *gateways*. O objetivo desse cenário é analisar como o mecanismo se comporta em uma rede densa com múltiplas fontes e múltiplos *gateways*, estando ambas as fontes a uma distância geográfica igual de ambos os *gateways*. Para esse fim, os nós 0 e 4 são escolhidos como fontes, enquanto os nós 1 e 3 são configurados como *gateways* da rede.

A segunda topologia em grade, ilustrada na Figura 5.10, é formada por 64 nós dispostos em uma matriz 8x8. Cada lado da matriz tem 140 m. Assim, cada nó se distancia 20 m

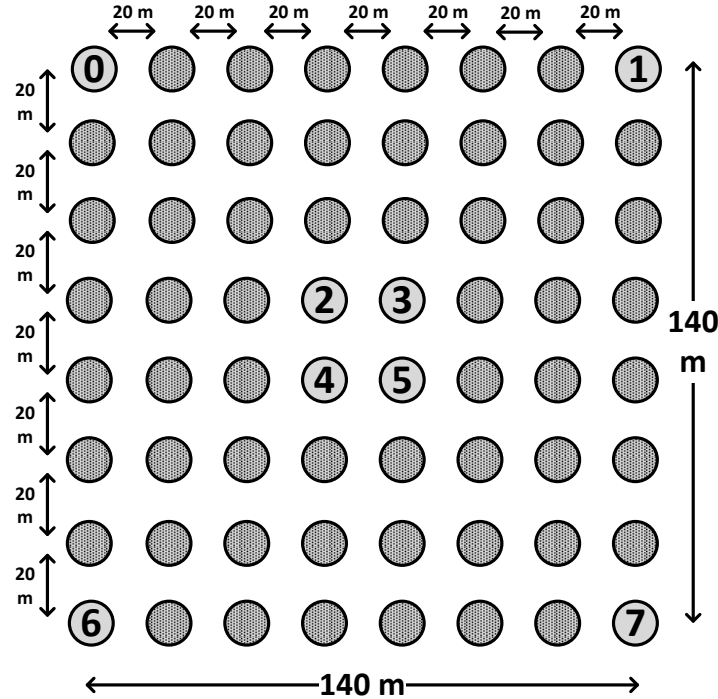


Figura 5.10: Topologia para simulações: topologia em grade 8x8.

dos seus vizinhos próximos de linha ou coluna. O uso dessa topologia tem como objetivo realizar testes com vários nós origem e vários *gateways* em uma rede densa. Para esta topologia, dois cenários foram idealizados para realizar a avaliação, ambos com quatro nós fonte e quatro *gateways*:

- Em um dos cenários, os quatro nós centrais da rede (2, 3, 4 e 5) fazem o papel de nós fontes, enquanto os quatro nós vértices da rede (0, 1, 6 e 7) são os *gateways*. Esse cenário foi modelado de forma a propiciar maior disputa pelo meio no núcleo, próximo às fontes dos fluxos destinados aos vértices da matriz 8x8. Assim, os nós de origem estão dentro do raio de transmissão uns dos outros;
- No outro cenário, os quatro nós vértices da rede (0, 1, 6 e 7) fazem o papel de nós fontes, enquanto os quatro nós centrais da rede (2, 3, 4 e 5) são os *gateways*. A modelagem desse cenário tem como intenção fazer que o fluxo tenha o sentido inverso do cenário anterior, com maior disputa pelo meio no núcleo da rede, próximo aos *gateways*.

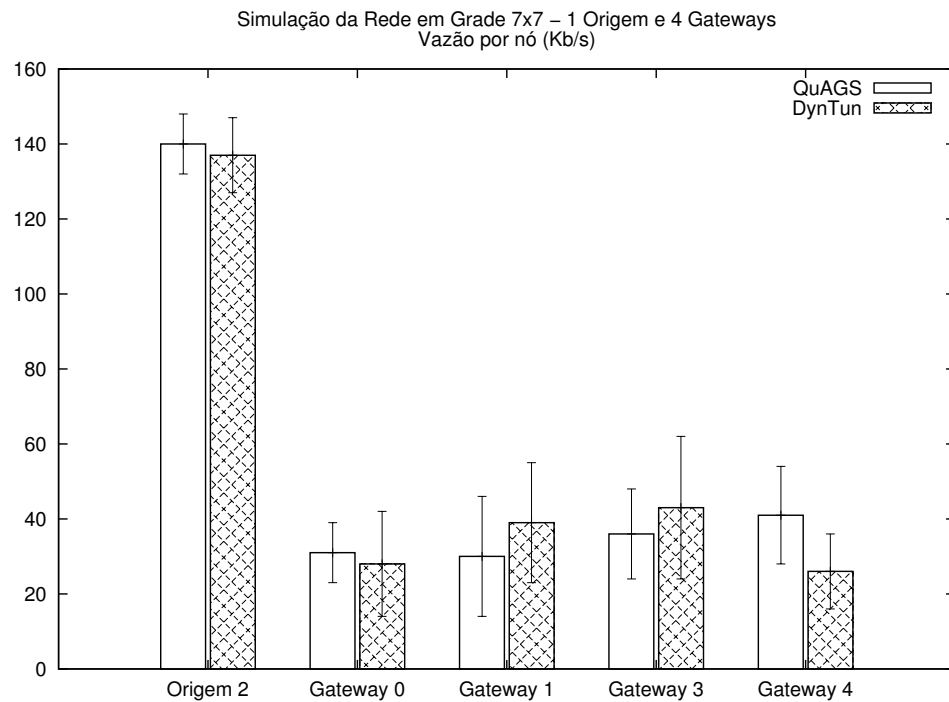


Figura 5.11: Simulação da topologia em grade 7x7, cenário com uma origem e quatro *gateways* - vazão média por nó.

<i>Gateway</i>	QuAGS	DynTun
0	23,9%	21%
1	23,2%	29%
3	30,4%	27%
4	22,5%	23%

Tabela 5.7: Simulação da topologia em grade 7x7, cenário com uma origem e quatro *gateways* - porcentagem de uso dos *gateways*.

5.2.3.1 Resultados - Topologia em Grade 7x7

Os resultados a seguir referem-se ao cenário com um nó origem e quatro *gateways* na rede. A Figura 5.11 apresenta a vazão média das 100 conexões TCP, com o DynTun e com o QuAGS, nas dez simulações. A Tabela 5.7 apresenta a porcentagem média de uso dos *gateways* nos testes realizados pelo nó origem.

Na média, o QuAGS teve uma vazão de 140,3 kbps, enquanto o DynTun teve uma vazão de 137,9 kbps. Esse cenário é mais um caso no qual a característica da topologia e do cenário da rede limita a sua vazão. Pode-se ver pelos dados da Tabela 5.7 que, na média, o uso dos *gateways* em ambos os mecanismos é semelhante. O fato da distância entre o nó origem e os quatro *gateways* da rede ser geograficamente igual se assemelha

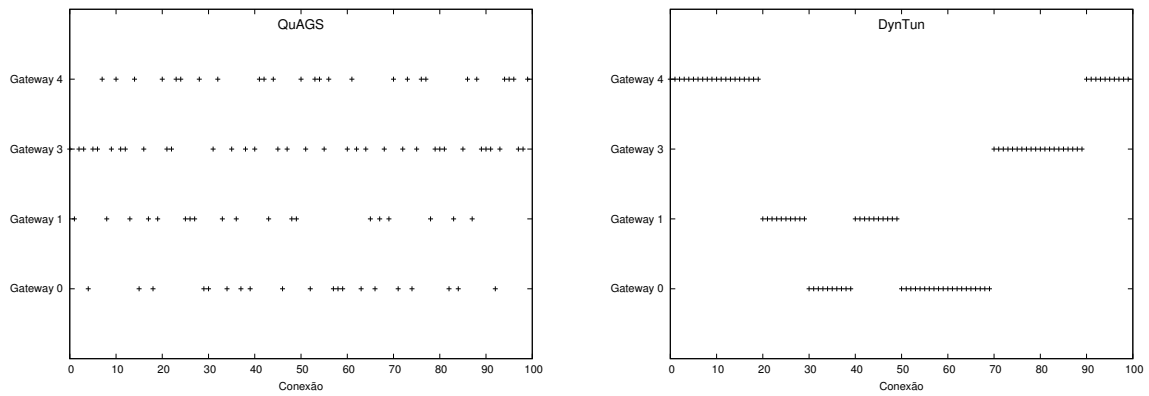


Figura 5.12: Simulação da topologia em grade 7x7, cenário com uma origem e quatro *gateways* - histórico de uso dos *gateways*.

<i>Gateway</i>	QuAGS	DynTun
1	38%	51%
3	40,6%	49%
4	21,4%	0%

Tabela 5.8: Simulação da topologia em grade 7x7, cenário com uma origem e três *gateways* - porcentagem de uso dos *gateways*.

às características da rede linear. Logo, assim como na rede linear, a porcentagem total de uso dos *gateways* não é um fator preponderante na eficiência do mecanismo e sim a distribuição da escolha de *gateways* ao longo do tempo. A Figura 5.12 apresenta a seleção de *gateways* ao longo do tempo de simulação pelo DynTun e pelo QuAGS na semente 8, onde o uso dos *gateways* no DynTun e no QuAGS são aproximadamente semelhantes. De igual forma a rede linear, o QuAGS realiza o balanceamento de carga de forma suave, buscando evitar a saturação do caminho de um *gateway*, o que ocorre no uso do DynTun, onde o *gateway* escolhido para uma nova conexão muda apenas quando o caminho para anterior encontra-se com a métrica de roteamento reduzida.

Os resultados do segundo cenário da topologia em grade 7x7, com um nó origem e três *gateways*, estão apresentados a seguir. A Figura 5.13 apresenta a vazão média das 100 conexões TCP, com o DynTun e com o QuAGS, nas dez simulações. A Tabela 5.8 apresenta a porcentagem média de uso dos *gateways* nos testes realizados pelo nó origem.

Na média, o QuAGS teve uma vazão de 191,2 kbps, enquanto o DynTun teve uma vazão de 189 kbps. Esse cenário apresenta uma mistura das características das diferentes topologias e seus cenários já apresentados nesse trabalho. Como geograficamente os *gateways* 1 e 3 estão a mesma distância do nó origem 0, eles são escolhidos com mais

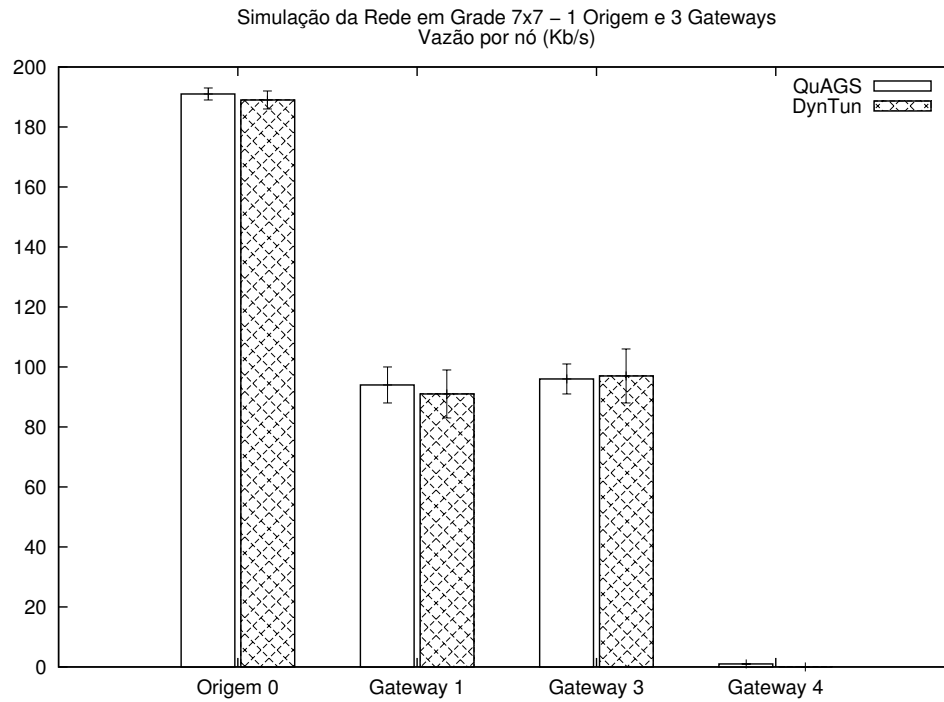


Figura 5.13: Simulação da topologia em grade 7x7, cenário com uma origem e três *gateways* - vazão média por nó.

frequência do que o *gateway* 4. O DynTun, por exemplo, não seleciona o *gateway* 4 em nenhuma das sementes testadas. Isso ocorre porque quando o DynTun seleciona um dos *gateways* mais próximos, ao ocorrer a saturação do caminho até ele, o outro *gateway* mais próximo é escolhido. Quando o caminho para esse segundo *gateway* é saturado, o caminho do primeiro *gateway* selecionado se recupera. Assim, o terceiro *gateway* nunca é utilizado, o que apresenta uma potencial subutilização de recursos da rede. O QuAGS apresenta o comportamento esperado, utilizando todos os *gateways* da rede. Pela tabela pode-se perceber que o uso de métrica como probabilidade de escolha para seleção do *gateway* faz com que o *gateway* localizado a um maior número de saltos da origem seja utilizado para aliviar a carga dos dois *gateways* mais próximos, de modo que ele é o *gateway* menos utilizado dos três. Porém, a vazão nesse *gateway* é mais de 100 vezes inferior ao dos outros dois *gateways*. Assim, antes de decidir acerca de qual mecanismo utilizar na rede, é necessário analisar o compromisso entre a vazão agregada da rede, a porcentagem de uso dos *gateways* e a vazão individual das conexões que sairão pelo *gateway* de pior qualidade.

A seguir estão apresentados os resultados do terceiro cenário da topologia em grade 7x7, que consiste de dois nós origens e dois *gateways*, todos localizados nos vértices da rede. A Figura 5.14 apresenta a vazão média das 100 conexões TCP nos nós origens e nos

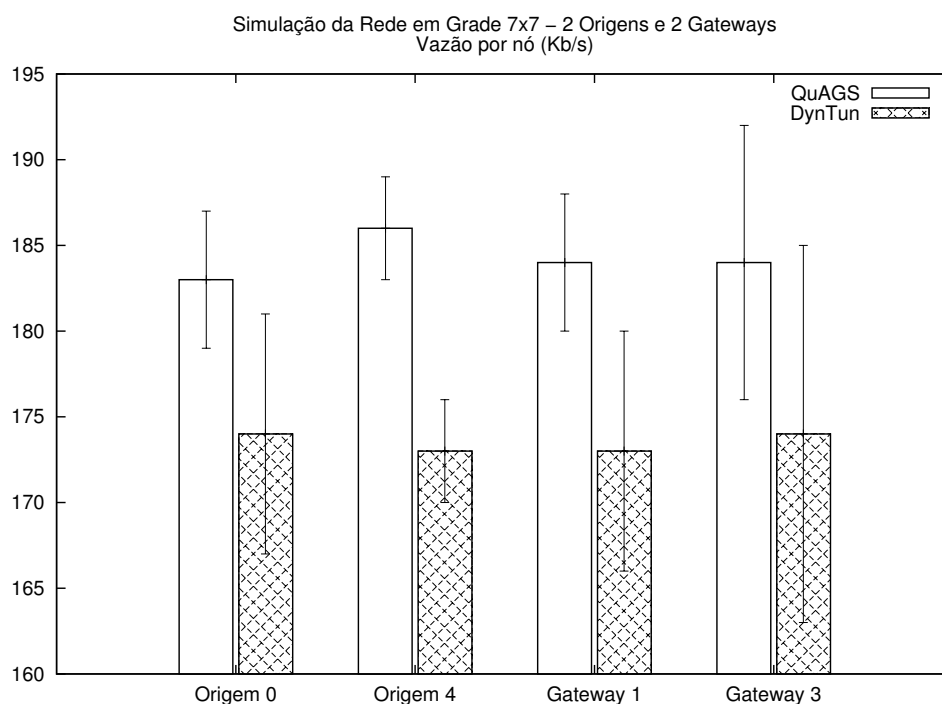


Figura 5.14: Simulação da topologia em grade 7x7, cenário com duas origens e dois *gateways* - vazão média por nó.

Mecanismo (nó origem)	Escolha <i>Gateway</i> 1	Escolha <i>Gateway</i> 3
QuAGS (0)	48,4%	51,6%
DynTun (0)	50%	50%
QuAGS (4)	51,1%	48,9%
DynTun (4)	47%	53%

Tabela 5.9: Simulação da topologia em grade 7x7, cenário com duas origens e dois *gateways* - percentagem de uso dos *gateways*.

gateways, com o DynTun e com o QuAGS. A Tabela 5.9 apresenta a percentagem média de uso dos *gateways* nos testes realizados pelos nós origens.

O Nó 0 teve uma vazão média de 183 kbps com o QuAGS e 174,7 kbps com o DynTun. O Nó 4 obteve resultados de vazão de 186,5 kbps e 173,2 kbps com o QuAGS e DynTun, respectivamente. O resultado da Tabela 5.9 mostra que, na média, ambos mecanismos utilizam os *gateways* de forma igualitária. Assim como nos cenários da topologia linear e da grade 7x7 com 1 origem e 4 *gateways*, os nós origem estão geograficamente equidistantes aos *gateways* da rede. Da mesma forma, a percentagem total de uso dos *gateways* não é um fator preponderante na eficiência do mecanismo, e sim a distribuição da escolha de *gateways* ao longo do tempo. Assim como nos cenários citados, o balanceamento de carga é realizado de forma suave pelo QuAGS. A média de trocas de escolha de *gateway* para

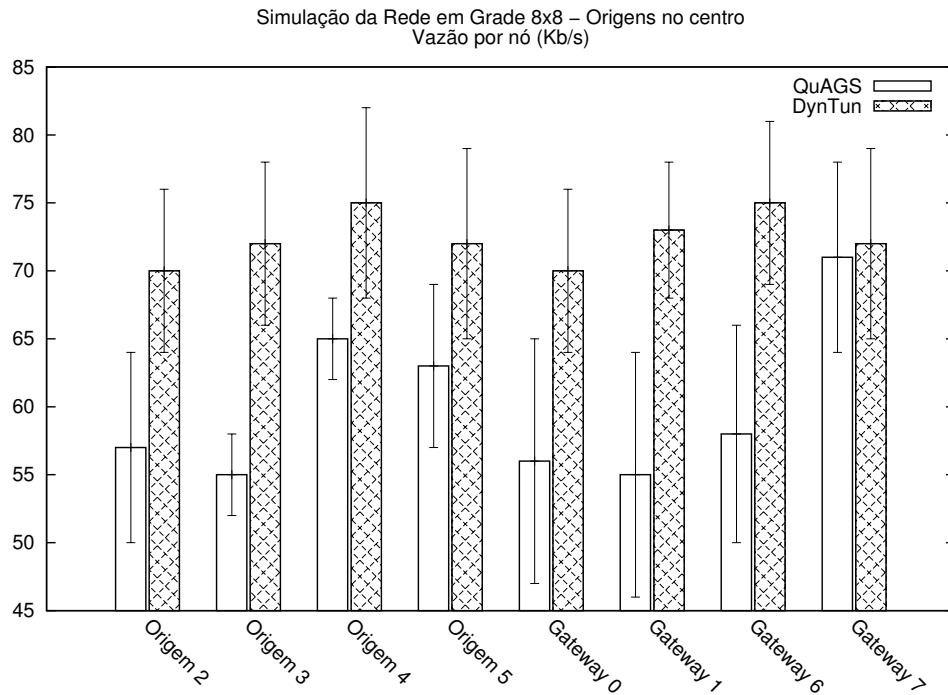


Figura 5.15: Simulação da topologia em grade 8x8, cenário com origens no centro - vazão média por nó.

conexões consecutivas pelos mecanismo aponta a suavidade no balanceamento de carga com o QuAGS: nas dez sementes de simulação, o QuAGS apresenta uma média de 50,5 trocas realizadas pelo nó origem 0 e 51,2 trocas realizadas pelo nó origem 4, enquanto o DynTun apresenta uma média de 4,2 trocas realizadas pelo nó origem 0 e 2,7 trocas realizadas pelo nó origem 4.

5.2.3.2 Resultados - Topologia em Grade 8x8

Os resultados de vazão média das 100 conexões por nó de origem no cenário com as origens no centro da rede está apresentado na Figura 5.15.

A vazão média do Nó 2, mais perto do *Gateway 0*, é de 57,8 kbps com o QuAGS e 70,3 kbps com o DynTun. O Nó 3, mais perto do *Gateway 1*, atinge uma vazão média de 55,2 kbps com o QuAGS e 72,8 kbps com o DynTun. Por sua vez, o Nó 4, que está mais próximo do *Gateway 6*, tem uma vazão média de 65,2 e 75,8 kbps para QuAGS e DynTun respectivamente. E por fim, o Nó 5, que se localiza mais próximo do *Gateway 7*, tem resultados de vazão média de 63,3 kbps para o QuAGS e 72,6 kbps para o DynTun.

Os resultados de vazão média das 100 conexões por nó de origem no cenário com as

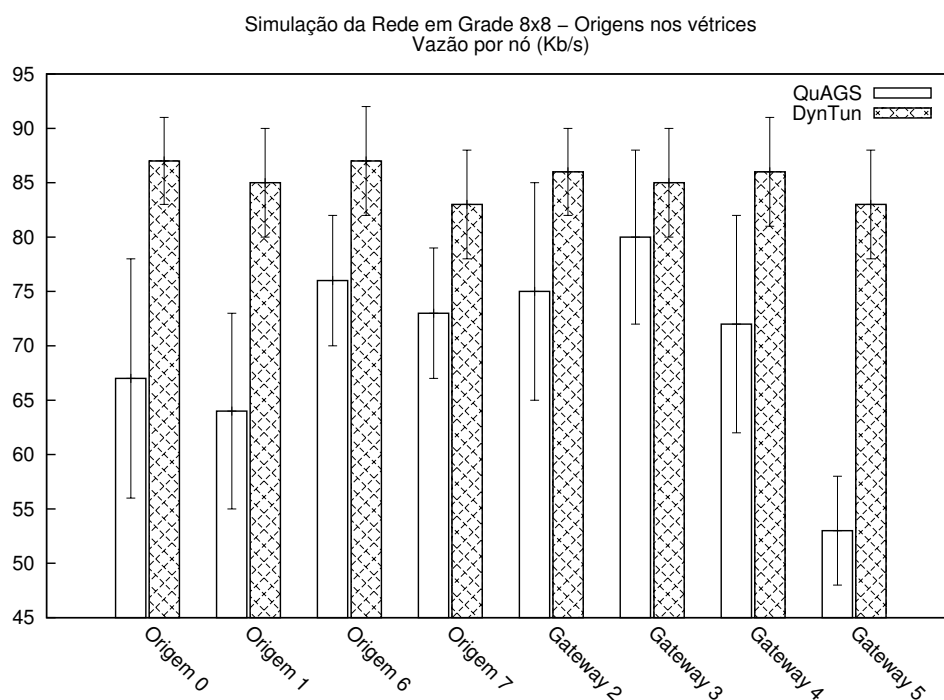


Figura 5.16: Simulação da topologia em grade 8x8, cenário com origens nos vértices - vazão média por nó origem.

origens nas extremidades da rede está apresentado na Figura 5.16.

O Nó 0, que se localiza mais próximo do *Gateway 2*, tem vazão média de 67,7 kbps com o QuAGS e 87 kbps com o DynTun. O Nó origem 1, mais perto do *Gateway 3*, atinge uma vazão média de 65 kbps com o QuAGS e 85,4 kbps com o DynTun. Os resultados de vazão média do Nó 6, que se localiza mais próximo do *Gateway 4*, são de 76,2 kbps com QuAGS e 87,1 kbps com DynTun. Finalmente, o Nó 7, mais próximo do *Gateway 5*, tem resultados de vazão média de 73,4 kbps para o QuAGS e 83,6 kbps para o DynTun.

Para ambos os cenários, o QuAGS fez uma distribuição balanceada entre os *gateways* da rede, enquanto o DynTun direcionou todas as conexões de um nó origem para o seu *gateway* mais próximo. Nesses cenários, a limitação do QuAGS é apresentada. Tal qual foi visto a queda de vazão nos nós próximos aos *gateways* nas simulações da rede interna com múltiplos *gateways* (Figura 5.4), o mesmo ocorre em ambos os cenários. Pela característica do QuAGS de realizar o balanceamento de carga na rede entre os múltiplos *gateways*, ele seleciona todos os *gateways* em algum momento para um determinado nó origem, quando a melhor opção seria utilizar sempre o *gateway* mais próximo dele. Assim, pode-se concluir que o QuAGS é um mecanismo mais indicado para nós que sejam mais centrais na rede.

Capítulo 6

Conclusão

Este trabalho apresentou o QuAGS, um mecanismo de seleção de *gateways* para prover balanceamento de carga em redes em malha sem fio com múltiplos *gateways*. O QuAGS é uma solução de fácil implementação, descentralizada e transparente ao usuário. Com base na qualidade dos caminhos entre um cliente e os *gateways*, informado ao mecanismo pelo protocolo de roteamento, determina-se a probabilidade de seleção de um determinado *gateway* por uma nova conexão.

O assunto de redes em malha sem fio tem uma grande gama de problemas em aberto. Características de uma rede desse tipo, suas vantagens e desvantagens são apresentadas na parte inicial desse trabalho. Entre esses problemas estão o balanceamento de carga e utilização de recursos nesse tipo de rede. O QuAGS tem como objetivo principal aproveitar os recursos da rede, de modo a evitar a sobrecarga de *gateways* que normalmente são superutilizados, encaminhando algumas de suas conexões para *gateways* normalmente subutilizados. Porém, é necessário que essa ação não diminua a vazão da rede, que pode acontecer caso os *gateways* normalmente subutilizados (por serem alcançados por caminhos de menor qualidade) sejam utilizados de forma demasiada.

Também foi apresentado brevemente o DynTun, um mecanismo que tem como objetivo evitar a quebra de conexões TCP em redes em malha sem fio com múltiplos *gateways* que utilizam a técnica NAT. Esse problema também é brevemente apresentado. Um dos objetivos do QuAGS é manter essa característica desejada do mecanismo DynTun. Os problemas e o potencial do DynTun servem como ponto de partida para a criação do QuAGS.

O QuAGS é implementado em uma rede experimental, em roteadores comerciais. Poucas adições de módulos são necessárias, de forma que a implementação do mecanismo é simples. Apesar de ser implementado como uma extensão do protocolo de roteamento

OLSR, o mecanismo é de fácil implementação em qualquer protocolo de roteamento, desde que a qualidade dos enlaces seja visível para o QuAGS. Além da rede experimental, o QuAGS também é implementado e avaliado através de simulações no simulador de rede NS-2. As características da implementação para o simulador são ligeiramente diferentes da implementação real, porém o necessário é implementado para chegar ao objetivo principal da simulação, que é avaliar o modo de escolha de *gateways*.

Em todos os testes realizados, o desempenho do QuAGS é comparado ao desempenho do DynTun. Mostrou-se nos experimentos práticos que o QuAGS obtém um balanceamento de carga eficiente entre os *gateways* da rede, sem que a vazão seja comprometida.

Nas simulações as características do QuAGS e do DynTun são evidenciadas. O QuAGS se apresenta superior quando utilizado em nós que não se encontram perto (pelo número de saltos) de nenhum *gateway* da rede e em redes onde a disputa pelo meio é baixa. Em nós que estão mais perto de um dos *gateways* do que dos outros, a probabilidade do QuAGS de enviar tráfego para *gateways* que estejam a um maior número de saltos existe, ao contrário do DynTun, onde na sua maioria, é inexistente. O envio demorado de tráfego para esses *gateways* com um maior número de saltos gera um impacto negativo na vazão, principalmente nos cenários com múltiplas origens de tráfego. Logo, recomenda-se utilizar o QuAGS em nós que não estejam próximos de nenhum *gateway*. O cenário da rede linear mostra o bom funcionamento do QuAGS e o da rede em grade 8x8 evidencia sua limitação. Devido às características apresentadas nesse trabalho, recomenda-se o QuAGS para redes onde a economia de energia seja um fator crítico. Uma rede de sensores com múltiplos nós sorvedouros é um exemplo.

Concluiu-se também que o QuAGS suaviza o efeito de alternância de conexões entre os múltiplos *gateways*. Apesar de em alguns casos o DynTun, em número absoluto, alcançar o balanceamento de carga entre os *gateways* da rede, foi apresentado que ele realiza a troca de escolha do melhor *gateway* quando o caminho para ele se satura, de forma a não realizar o balanceamento da forma que o QuAGS realiza. O QuAGS faz um balanceamento de carga de forma suave ao longo do tempo, evitando a saturação do caminho.

6.1 Trabalhos Futuros

Alguns trabalhos futuros são propostos para uma melhor avaliação do QuAGS, assim como um possível aumento no ganho de vazão do mecanismo.

Um possível trabalho futuro inclui a realização de testes em mais topologias de redes

experimentais. O objetivo é avaliar se o ganho de vazão do QuAGS se assemelha aos casos nas simulações, nos quais a vazão da rede é maior. Assim, deve ser verificado se consequentemente o ganho do QuAGS sobre o DynTun também é maior. Apesar de ser possível também a criação de mais topologias e cenários nas simulações, acredita-se que um grande número de combinações diferentes foi realizado e a tendência de escolha do QuAGS e do DynTun são conhecidas.

A comparação com outros mecanismos de seleção de *gateways*, dentre eles os apresentados na Seção 3.3.1, é outra atividade possível de ser realizada como trabalho futuro. Apesar de esses mecanismos terem funcionamento diferente do QuAGS e do DynTun, pode-se reimplementá-los com marcação e túneis lógicos, de forma a comparar não os mecanismos em si, mas a estratégia de seleção de *gateway* de cada um. Outra estratégia de escolha de *gateway* que seria interessante de analisar é um *baseline*, onde cada conexão é escolhida ciclicamente por um *gateway* diferente, de forma a realizar forçadamente uma distribuição de carga igualitária. O impacto dessa forma de escolha na vazão da rede, comparado ao DynTun (que tende a escolher apenas os *gateways* mais próximos) e ao QuAGS é um resultado a ser avaliado.

O impacto da variação do valor do ε , apresentado na Equação 4.2 não é abordado nesse trabalho. Assim, um possível trabalho futuro é fazer essa avaliação. Outro possível trabalho seguindo essa linha é propor outra função para atualização das métricas na rede como, por exemplo, o uso do histórico das métricas, atribuindo pesos para a métrica recente e as métricas do histórico.

O estudo de outras funções para calcular a probabilidade de seleção dos *gateways* é outra proposta de trabalho futuro, comparando essas funções com a proposta nesse trabalho. O interessante de ter mais de uma função no mesmo mecanismo seria poder adaptá-lo à característica da rede onde ele é implantado.

Referências

- [1] A. Akella, B. Maggs, S. Seshan, A. Shaikh e R. Sitaraman. A measurement-based analysis of multihoming. Em *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 353–364. ACM, 2003.
- [2] I.F. Akyildiz, X. Wang e W. Wang. Wireless mesh networks: a survey. *Computer networks*, 47(4):445–487, 2005.
- [3] E. Ancillotti, R. Bruno e M. Conti. Load-balanced routing and gateway selection in wireless mesh networks: Design, implementation and experimentation. Em *IEEE WoWMoM*, pp. 1–7. IEEE, 2010.
- [4] M.E.M. Campista, L.H.M.K. Costa e O.C.M.B. Duarte. Privileging long-life packets in multihop wireless networks. Em *9th IFIP MWCN*, pp. 136–140. IEEE, 2007.
- [5] M.E.M. Campista, L.H.M.K. Costa e O.C.M.B. Duarte. WPR: a proactive routing protocol tailored to wireless mesh networks. Em *IEEE GLOBECOM*, pp. 1–5, 2008.
- [6] M.E.M. Campista, P.M. Esposito, I.M. Moraes, LHM Costa, OCM Duarte, D.G. Passos, C.V.N. Albuquerque, D.C.M. Saade e M.G. Rubinstein. Routing metrics and protocols for wireless mesh networks. *IEEE Network*, 22(1):6–12, 2008.
- [7] R. Carrano, L.C.S. Magalhaes, D.C. Saade e C.V.N. Albuquerque. IEEE 802.11s Multihop MAC: a tutorial. *Communications Surveys & Tutorials, IEEE*, (99):1–16, 2010.
- [8] C. da Silva, D. Passos, J. Duarte, I. Moraes e C.V.N. de Albuquerque. DynTun: A tool for providing multihoming support in wireless mesh networks. Em *International Information and Telecommunication Technologies Conference (I2TS)*, 2010.
- [9] C. da Silva, D. Passos, I. Moraes e C.V.N. de Albuquerque. QuAGS: Um Mecanismo de Seleção de Gateways Eficiente Baseado na Qualidade dos Enlaces. Em *Simpósio Brasileiro de Telecomunicações (SBrT)*, 2011.
- [10] K. Egevang e P. Francis. The IP network address translator (NAT). RFC 1631, 1994.
- [11] D. Farinacci, T. Li, S. Hanks, D. Meyer e P. Traina. Generic routing encapsulation (GRE). RFC 2784, 2000.
- [12] J.J. Galvez, P.M. Ruiz e A.F.G. Skarmeta. A feedback-based adaptive online algorithm for Multi-Gateway load-balancing in wireless mesh networks. Em *IEEE WoWMoM*, pp. 1–9. IEEE, 2010.
- [13] <http://isi.edu/nsnam/ns/>. The network simulator ns-2. Acessado em 22-11-2011.

- [14] <https://openwrt.org/>. Openwrt. Acessado em 22-11-2011.
- [15] <http://www.cisco.com/en/US/netsol/ns621/index.html>. Cisco outdoor wireless network solution. Acessado em 22-11-2011.
- [16] <http://www.iptables.org/>. Iptables and netfilter. Acessado em 22-11-2011.
- [17] <http://www.linuxfoundation.org/collaborate/workgroups/networking/iproute2>. Iproute2. Acessado em 22-11-2011.
- [18] <http://www.midiacom.uff.br/remote>. Projeto remote. Acessado em 22-11-2011.
- [19] IEEE. 1999 Edition IEEE Standards for Information Technology Telecommunications and Information Exchange Between System Local and Metropolitan Area Network Specific Requirements Part 11: Wireless LAM Medium Access Control (MAC) and Physical Layer (PHY) Specification. <http://standards.ieee.org/getieee802/download/802.11-1999.pdf>, 1999.
- [20] IEEE. 2005 Edition IEEE Standard for Information Technology - Telecommunications and Information Exchange Systems Between Systems - Local and Metropolitan Area Networks-Specific Requirements - Part 15.1a: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Wireless Personal Area Networks (WPAN). <http://standards.ieee.org/getieee802/download/802.15.1-2005.pdf>, 2005.
- [21] IEEE. 2009 Edition IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Broadband Wireless Access Systems. <http://standards.ieee.org/getieee802/download/802.16-2009.pdf>, 2009.
- [22] M. Iqbal, X. Wang, D. Wertheim e X. Zhou. Load balanced multiple gateway support in wireless mesh networks for broadband services. Em *18th Annual Wireless and Optical Communications Conference, 2009*, pp. 1–5. IEEE, 2009.
- [23] M. Ito, T. Shikama e A. Watanabe. Proposal and evaluation of multiple gateways distribution method for wireless mesh network. Em *Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication*, pp. 18–25. ACM, 2009.
- [24] A. Kamerman e N. Erkocevic. Microwave oven interference on wireless lans operating in the 2.4 ghz ism band. Em *The 8th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 1997. 'Waves of the Year 2000'. PIMRC'97.*, volume 3, pp. 1221–1227. IEEE, 1997.
- [25] Kurose, J. and Ross, K. *Computer Networking: A Top-Down Approach*. Pearson Addison-Wesley, 5th edition, 2010.
- [26] S. Lakshmanan, K. Sundaresan e R. Sivakumar. On multi-gateway association in wireless mesh networks. Em *2nd IEEE Workshop on Wireless Mesh Networks*, pp. 64–73, 2006.
- [27] D. Nandiraju, L. Santhanam, N. Nandiraju e D.P. Agrawal. Achieving load balancing in wireless mesh networks through multiple gateways. Em *IEEE MASS*, pp. 807–812, 2006.

- [28] D. Passos, C.V.N. de Albuquerque, M.E.M. Campista, L.H.M.K. Costa e O.C.M.B. Duarte. Minimum loss multiplicative routing metrics for wireless mesh networks. *Journal of Internet Services and Applications*, 1(3):1–14, 2011.
- [29] J. Postel. Transmission control protocol. RFC 793, 1981.
- [30] Theodore S. Rappaport. *Wireless Communications: Principles and Practice*. IEEE Press, 1996.
- [31] M.G. Rubinstein e J.F. Rezende. Qualidade de serviço em redes 802.11. *XX Simpósio Brasileiro de Redes de Computadores (SBRC2002)*, 2002.
- [32] J. Tsai e T. Moors. Minimum interference multipath routing using multiple gateways in wireless mesh networks. Em *5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems, 2008 (MASS 2008)*., pp. 519–520. IEEE, 2008.