

# DynTun: Túneis Dinâmicos e a Escalabilidade de Redes em Malha\*

Jairo L. Duarte , Diego Passos e Célio Vinicius Neves de Albuquerque

Instituto de Computação – Universidade Federal Fluminense (UFF)

{jduarte, dpassos, celio}@ic.uff.br

**Abstract.** *This paper proposes, implements and evaluates a solution to the problem of multi-homing in wireless mesh networks. This problem comes from the interaction between dynamic routing, common in mesh networks, and the use of NAT, usual in access networks. The high variability in routing choices may result in a constant swap of the default gateway. However, the use of NAT may lead to issues, causing disruptions in user connections. The proposal presented in this paper, called DynTun, is based on dynamic management of tunnels, packet tagging and routing policy, and solves the problem in an efficient way, without the need of adding new elements to the network. The DynTun evaluation shows that it is possible to harmonize, likely with scalability improvement, the use of multiple gateways and the NAT on the same access network.*

**Resumo.** *Este artigo propõe, implementa e avalia uma solução para o problema de multi-homing em redes em malha sem fio. Este problema surge da interação entre o roteamento dinâmico, característico das redes em malha, e a utilização de NAT, comum em redes de acesso. A alta variabilidade na escolha das rotas pode resultar em trocas constantes do gateway preferencial. Entretanto, o NAT traz problemas neste cenário, causando quebras nas conexões dos usuários. Este trabalho propõe o DynTun, uma solução eficiente baseada na criação dinâmica de túneis, marcação lógica de pacotes e políticas de roteamento sem a necessidade da adição de novos elementos na rede. Os testes do DynTun demonstram que é possível harmonizar, com potencial de aumentar a escalabilidade, o uso de múltiplos gateways com NAT em uma rede de acesso.*

## 1. Introdução

Redes em malha [15], ou *mesh*, são redes auto-configuráveis que utilizam comunicação sem fio de múltiplos saltos, formando um *backbone* que interconecta pontos de acesso tipicamente estacionários [1]. Nos últimos anos, estas redes vêm ganhando cada vez mais atenção por parte da comunidade científica. Diversos projetos de pesquisa, geralmente com foco na inclusão digital, têm utilizado uma infra-estrutura em malha para a implantação de redes de acesso [27]. Esta popularidade das chamadas redes *mesh* também é notória em termos comerciais. Grandes empresas do ramo, como Cisco e Nortel, já contam com soluções *mesh* em suas linhas de produtos [25, 22].

Redes em malha tipicamente estendem o alcance de redes *Wi-Fi* tradicionais. Entretanto, elas podem apresentar um gargalo. Embora teoricamente seja possível abranger

---

\*Este trabalho foi realizado com o apoio do CNPq, CAPES, Faperj, RNP e TBE/Aneel.

uma área de cobertura arbitrariamente grande, na prática foi demonstrado [5, 19] que a vazão obtida na comunicação entre dois nós cai exponencialmente com o aumento do número de saltos. Desta forma, a partir de uma determinada distância, a comunicação entre dois nós se torna restrita. No caso de uma rede de acesso, usuários localizados a um grande número de saltos do *gateway* seriam prejudicados, obtendo um desempenho consideravelmente inferior ao dos demais.

Uma solução bastante imediata é a adoção de múltiplos nós *gateways* para a Internet, ou seja, implementar *multi-homing*. Se tais *gateways* forem bem posicionados, atendendo aos demais roteadores da rede de maneira uniforme, o gargalo deixará de existir, permitindo um crescimento escalável da rede. Define-se *multi-homing* [12] como a utilização de mais de um ponto de conexão com a Internet em uma rede de acesso.

No entanto, o simples aumento do número de conexões com a Internet não garante a melhora do desempenho. Sem os devidos cuidados, este modelo pode trazer problemas de conectividade para alguns nós, fazendo com que as conexões dos usuários sejam quebradas constantemente. Tais problemas são decorrentes da utilização da técnica tradicional de NAT (*Network Address Translation*) [9] em conjunto com *multi-homing*.

A rede Remesh [15, 7] é uma bem sucedida implementação de rede em malha que enfrentou o supracitado gargalo, sendo assim a motivação para este trabalho. Esta rede é fruto do projeto Remesh, cujo objetivo é incluir digitalmente a comunidade acadêmica da Universidade Federal Fluminense (UFF), através do oferecimento de acesso banda larga para as vizinhanças dos *campi* da universidade.

Este trabalho propõe a técnica DynTun, uma solução eficiente para o problema de *multi-homing* em redes em malha que utilizam a técnica NAT. Esta solução é baseada em criação dinâmica de túneis, marcação lógica de pacotes e políticas de roteamento. É apresentada também uma implementação concreta da proposta, que possibilitou a avaliação da solução em uma rede real. Os resultados dos testes de desempenho mostram que a solução DynTun preserva a semântica das conexões dos usuários e tem o potencial de aumentar o desempenho e a escalabilidade da rede, assim como comprovam que o acréscimo de custo computacional no processamento dos pacotes é bastante baixo.

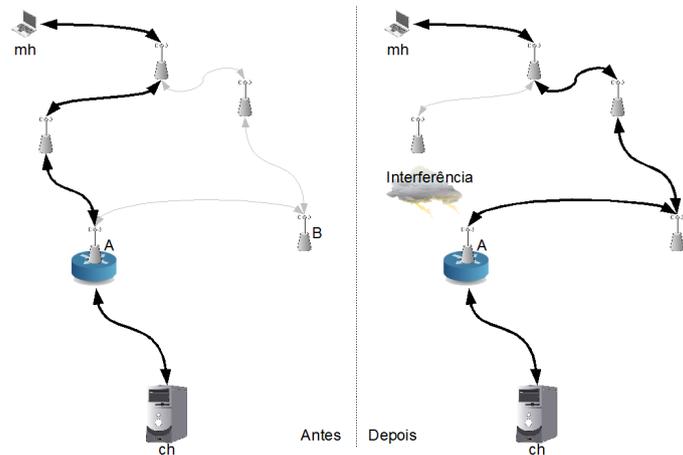
Apesar do DynTun ter sido desenvolvido para a rede Remesh, os critérios e mecanismos de implementação selecionados são comuns a diversas outras redes de acesso. Portanto, a solução é potencialmente aplicável a outras redes.

O texto é organizado da seguinte forma: na Seção 2 o problema de violação da semântica das conexões dos usuários é detalhado. Na Seção 3 são apresentados alguns critérios utilizados para a definição de uma solução para suporte a *multi-homing*. Na Seção 4 são analisadas algumas propostas existentes para este suporte. Na Seção 5 a solução DynTun é apresentada em detalhes. Na Seção 6 são comentados aspectos de implementação da solução proposta. Finalmente, na Seção 7 são exibidos os resultados da avaliação realizada, seguidos pelas considerações finais e trabalhos futuros na Seção 8.

## 2. Descrição do problema

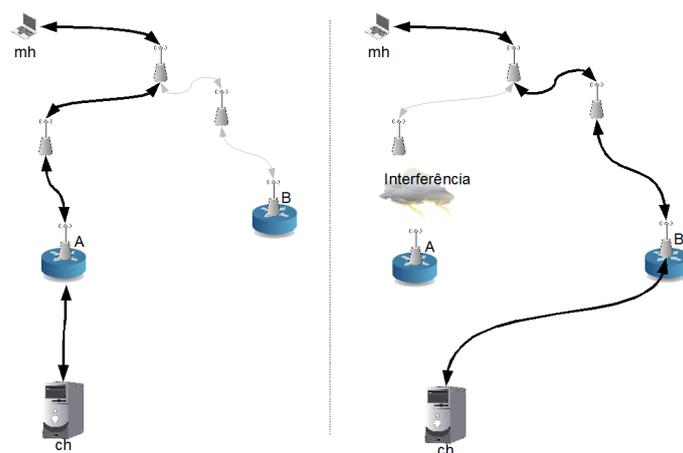
O problema de violação da semântica das conexões dos usuários acontece pela interação entre a escolha dinâmica de rotas e a utilização da técnica de NAT. A Figura 1 mostra um exemplo do comportamento desejável de um protocolo de roteamento no caso de

problemas em um enlace. Nela, o cliente *mh* utiliza inicialmente a rota mais curta ( em negrito) para o nó *gateway*. Em um dado instante de tempo, ocorre uma falha em um dos enlaces da rota utilizada. Neste instante, o protocolo deve detectar esta falha e alterar o caminho escolhido para uma rota alternativa.



**Figura 1. Exemplo de uma mudança de rota causada por problemas de conectividade.**

Neste cenário, o único efeito sentido pelo nó cliente é possivelmente o de perdas temporárias de alguns pacotes e um aumento do atraso fim-a-fim. No entanto, uma vez concluída a alteração de rotas, a conexão voltará ao seu funcionamento normal. A situação ilustrada pela Figura 1, entretanto, é válida para uma rede com apenas um *gateway*. Ao se analisar uma topologia semelhante, porém composta por dois *gateways*, é fácil entender o problema que pode ocorrer. A Figura 2 ilustra um cenário muito parecido com o anterior. Agora, no entanto, o nó *B* também trabalha como *gateway* para a Internet.



**Figura 2. Utilização de diversos gateways e a técnica NAT em conjunto.**

Ao se considerar, novamente, que uma falha ocorrerá em um dos enlaces da rota preferencial (em negrito) desta última topologia, é razoável supor que o protocolo de roteamento irá agir para contornar o problema, utilizando como nova rota o caminho que passa pelo *gateway B*. Desta forma, quando os pacotes de dados chegarem ao nó *B*, eles

serão encaminhados para o servidor *ch* com o endereço de *B*, devido a técnica NAT, e não com o endereço de *A*, como na antiga rota. Ao serem recebidos por *ch*, os pacotes não serão corretamente associados à sua conexão, causando a quebra da comunicação.

Embora neste exemplo tenha sido utilizado o caso em que um enlace apresenta falhas, a troca de *gateways* pelo protocolo de roteamento pode ocorrer pela própria variação da qualidade dos enlaces sem fio. Isto é especialmente verdadeiro em redes que trabalham com protocolos que realizam medidas ativas [4].

Pode-se argumentar que uma solução trivial para este problema é a não utilização do NAT. Entretanto, existem diversas vantagens [21] técnicas no emprego do NAT. Assim sendo, a busca ou elaboração de outras soluções que tornem possível utilizar NAT e *multi-homing* se fazem necessárias.

### 3. Critérios

Uma boa solução para o problema de violação da semântica das conexões, apresentado na Seção 2, deve atender a determinadas restrições. Tais restrições visam garantir a manutenção das características das redes em malha sem fio, em especial a rede real Remesh.

Desta forma, uma solução para o problema de *multi-homing* deve: facilitar os processos de desenvolvimento, implantação e modificação, a fim de permitir que futuros ajustes sejam facilmente realizáveis; ser transparente e não depender da cooperação do cliente ou do provedor de Internet; ter um baixo custo de implementação e manutenção; operar de forma autônoma e dinâmica; funcionar de forma eficiente no contexto de redes em malha; e, por fim, ser compatível com NAT.

Todas estas características são levadas em consideração na solução proposta neste trabalho, como será apresentado na Seção 5.

### 4. Trabalhos Relacionados

Em redes que possuem endereços IP públicos, independentes dos seus provedores de Internet (ISP), o balanceamento de carga por *multi-homing* pode ser obtido pela técnica de “BGP peering” [3, 25, 23]. A referida técnica, no entanto, requer que a rede possua uma identificação de sistema autônomo (*AS number*), construa acordos de *peering* com os seus provedores e gerencie tabelas de rotas BGP. Tais requerimentos são perfeitamente adequados para grandes corporações capazes de ter equipes e equipamentos especializados. Contudo, redes que utilizam endereços associados aos seus provedores de Internet (ISP) podem obter o balanceamento de carga pelo uso da técnica NAT [20, 16, 17]. Estas soluções utilizam mecanismos proprietários embarcados em equipamentos que não permitem que sejam implementadas modificações que dessem suporte a redes *mesh*. Além disso, seria necessária uma modificação da topologia da rede com a finalidade de criar uma hierarquia, onde os roteadores de borda da rede *mesh* estivessem ligados a um equipamento centralizador, que atuaria como único ponto de ligação à Internet e onde seria implementado o NAT.

Não restam dúvidas de que são relevantes as questões de custos levantadas por Goldenberg et al [11], relacionadas ao valor monetário do uso de múltiplas conexões. Quando múltiplos ISPs são utilizados, o custo financeiro do uso de cada *gateway* pode

ser diferente. Entretanto, isso não ocorre na rede Remesh que tem como preocupação atual a questão do desempenho. Deste modo, otimizações relacionadas ao custo ou ao balanceamento de carga no roteadores serão reservadas para trabalhos futuros, uma vez que todas as conexões utilizam um mesmo provedor e, portanto, possuem custo idêntico.

Um estudo realizado por Akella *et al* [2] quantifica o potencial benefício de desempenho trazido pelo uso de *multi-homing*, através da análise de históricos reais de tráfego da Internet. A conclusão apresentada é que o *multi-homing* pode melhorar em até 25% o desempenho em redes com dois provedores de acesso, sendo que boa parte dessa melhora pode ser obtida com até quatro provedores. Contudo, este estudo foi realizado em uma rede com infra-estrutura de comunicação por cabo, que apresenta importantes diferenças no desempenho, se comparada às redes sem fio. Desta forma, os resultados apresentados nesse trabalho podem não ser totalmente aplicáveis às redes em malha.

Nas soluções apresentadas em [26, 14], túneis são criados para um agente permanente (um *Home Agent*), o que requer o uso de servidores adicionais externos à rede com endereços IP fixos. Assim, maiores gastos são necessários, o que contradiz as necessidades de baixo custo das redes em malha. Contudo, ambas as propostas apresentam qualidades relevantes, como a possibilidade do uso de diversos provedores de acesso e o suporte à redundância de forma simplificada.

No trabalho apresentado por Shin *et al* [24], a decisão de selecionar o melhor *gateway* é relegada ao dispositivo cliente. Assim, esta opção não é transparente ao usuário, pois este deve gerenciar o *multi-homing*. Outra questão está na forma conservadora adotada para evitar o problema de se utilizar mais de um *gateway* durante uma mesma sessão. Os autores propõem que o mesmo *gateway* seja utilizado para todas as sessões, enquanto houver uma em atividade. Logo, outro *gateway* somente pode ser selecionado no momento em que todas as sessões são fechadas.

Uma proposta mais próxima dos critérios apresentados na Seção 3 é apresentada em [8]. Os autores consideram uma rede com características similares as das redes em malha, com uso de NAT nos roteadores e de túneis até esses roteadores. Contudo, a solução não é transparente aos dispositivos clientes, além de não haver seleção e gerenciamento dos túneis de forma dinâmica.

## 5. A Solução DynTun

Como explicado na Seção 2, uma vez que uma conexão TCP tenha sido aberta, os pacotes do cliente não podem chegar ao servidor com outro endereço IP de origem. Desta forma, tendo-se em vista os critérios apresentados na Seção 3, fica evidente que existem apenas duas classes de soluções desejáveis para o problema apresentado.

A primeira é composta por soluções em que todos os pacotes dos nós usuários são encaminhados para fora da rede com um mesmo endereço IP. Uma possível implementação deste método é através da centralização do NAT. Ou seja, o NAT não seria realizado pelos *gateways* da rede em malha, mas sim por um roteador central ao qual estes *gateways* estariam conectados através de uma infra-estrutura cabeada, por exemplo. Esta solução, entretanto, traz implicações de escalabilidade e tolerância a falhas. Uma outra alternativa seria fazer uma alteração na técnica de NAT, de forma que os pacotes de uma conexão sempre saiam da rede com o mesmo endereço IP. Em outras palavras,

os *gateways* seriam obrigados a “forjar” um endereço IP. Esta solução, no entanto, pode entrar em conflito com políticas de segurança de determinados ISPs, dado que muitos empregam a técnica *Ingress filtering* que verifica se o IP de origem dos pacotes pertence ao conjunto de sub-redes do ISP, com a finalidade de evitar ataques de *DoS (Denial of Service)*.

A segunda classe de soluções engloba aquelas em que cada conexão utiliza apenas um *gateway* durante toda sua duração. A esta classe pertence, por exemplo, a solução apresentada em [8], como comentado na Seção 4. Este tipo de solução não sofre dos problemas enfrentados pelas soluções da primeira classe, adequando-se assim a vários dos critérios especificados na Seção 3. Desta forma, optou-se neste trabalho por seguir os conceitos da segunda classe.

A proposta, denominada DynTun (*Dynamic Tunnels*), consiste de um método no qual ao se iniciar uma nova conexão, seja escolhido um *gateway* pelo roteador ao qual o cliente está conectado, de forma que todos os pacotes referentes àquela conexão sejam encaminhados através deste mesmo *gateway*. Para isso, cada roteador da rede que atende diretamente aos usuários deve criar dinamicamente um túnel até cada *gateway*. Quando um roteador recebe um pacote de dados oriundo de um dos seus usuários, ele verifica se é um pacote de abertura de conexão. Em caso afirmativo, o melhor *gateway* no momento é escolhido e associado à conexão. Por outro lado, se o pacote recebido é de uma conexão já existente, o roteador simplesmente verifica o *gateway* associado àquela conexão e encaminha a mensagem pelo túnel respectivo. É interessante notar que conexões distintas de um mesmo usuário podem ser roteadas por *gateways* diferentes. Assim, o DynTun apresenta o potencial de explorar roteamento por múltiplos caminhos.

A informação dos *gateways* disponíveis pode ser obtida através do protocolo de roteamento. Se o protocolo utilizado for baseado em estado de enlaces, a qualquer momento será possível obter essa informação, pois toda a topologia da rede é conhecida. Se, por outro lado, o protocolo adotado se baseia em vetor de distâncias, é necessária a implementação de um sistema de anúncio de nós *gateways* [24], para que os roteadores da rede saibam qual o melhor *gateway* em cada instante de tempo. A escolha do melhor *gateway* em cada momento é feita através dos critérios do próprio protocolo de roteamento. No caso da rede Remesh, utilizada como base deste trabalho, o protocolo utilizado é o OLSR-ML (*Optimized Link State Routing - Minimum Loss*) [18], baseado em estado de enlaces. Assim, a implementação do sistema de anúncio de *gateways* não foi necessária.

É fácil perceber pela descrição apresentada que os roteadores precisam manter estado das conexões ativas. A princípio, isto pode parecer um problema de escalabilidade. Na Seção 7 serão avaliados os impactos do aumento no processamento. Outra característica imediata é a transparência para os usuários. Toda a solução é implementada apenas nos roteadores da rede, não necessitando de modificações nos usuários.

Vale destacar que cada conexão tem a possibilidade de escolher o melhor *gateway* no momento de sua abertura. Para conexões de curta duração, existe uma grande probabilidade de que a escolha inicial se mantenha como a melhor durante toda a comunicação. Já para conexões mais longas, é razoável supor que eventualmente o melhor *gateway* passe a ser outro. Neste caso, o desempenho obtido pode não ser ótimo ao longo de toda a comunicação.

Pode-se verificar ainda um efeito colateral interessante. Como citado na Seção 2, a utilização de métricas de roteamento ativas por parte dos protocolos pode acentuar o problema de mudança de *gateways* por causar constantes alterações nas rotas. Por outro lado, com a utilização da solução DynTun, o que ocorre é um balanceamento natural de carga na rede. A medida que conexões são roteadas através de um mesmo *gateway*, os enlaces que compõem a rota até ele receberão uma classificação menor por parte do protocolo de roteamento. Assim, o protocolo poderá eventualmente escolher um *gateway* alternativo pelo qual as novas conexões serão abertas. Desta forma, um mesmo cliente poderá ter conexões estabelecidas através de vários *gateways* distintos.

## 6. Implementação

Além de propor a solução DynTun, neste trabalho também foi realizada uma implementação da proposta para fins de avaliação. Esta implementação foi realizada tomando-se por base as redes em malha implantadas pelo projeto Remesh [15, 7]. O projeto conta com duas implementações de redes em malha, sendo uma externa, construída ao redor de um dos *campi* da Universidade Federal Fluminense (UFF), e outra interna a um dos prédios da universidade. A rede externa, ilustrada na Figura 3, apresenta uma área de cobertura de cerca de 30  $km^2$  nas vizinhanças do *campus* localizado na cidade de Niterói, enquanto a rede interna (Figura 4) interconecta 7 salas (como laboratórios e bibliotecas) em dois andares de um dos prédios da Escola de Engenharia.

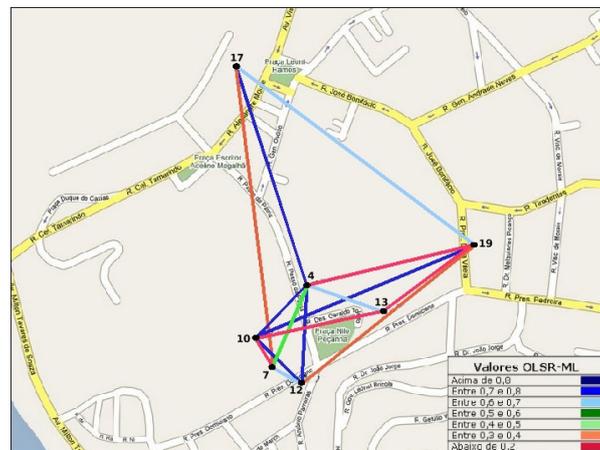


Figura 3. Visualização da topologia externa utilizada nos testes.

Em ambas, os nós da rede (exceto os usuários) são compostos por roteadores *Linksys WRT54G*. Este tipo de *hardware* apresenta grandes restrições de processamento [6] e memória, sendo assim um bom ambiente de testes para a avaliação do custo computacional da solução. Optou-se neste trabalho por desenvolver o DynTun como um módulo do protocolo de roteamento. Esta decisão simplificou o processo de obtenção da lista de *gateways* da rede, necessária à solução.

Ao ser iniciado, o DynTun cria um túnel GRE (*Generic Routing Encapsulation*) [10] do tipo NBMA (*Non-Broadcast Multi-Access*). Este tipo de túnel tem como característica a ausência de um endereço de destino na sua especificação. Além do túnel, são criadas algumas regras de marcação de pacotes utilizando o módulo *conntrack* da ferramenta *iptables* [13], freqüentemente encontrado em roteadores baseados em GNU/Linux.

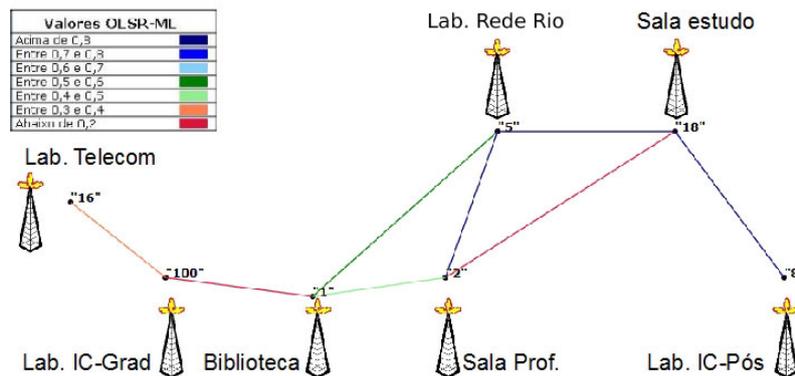


Figura 4. Visualização da topologia interna utilizada nos testes.

Cada conexão dos usuários recebe uma marcação lógica interna ao *kernel* do roteador (ou seja, o conteúdo do pacote não é alterado) que será verificada para determinar o *gateway* utilizado. A partir deste ponto, o DynTun entra em um ciclo que consiste em:

- buscar a lista de todos os *gateways* disponíveis na rede;
- criar uma tabela de rotas para cada *gateway*, contendo apenas uma entrada, a do *gateway* em questão;
- criar uma regra de roteamento associando uma determinada marca à tabela de roteamento de cada *gateway*;
- identificar o melhor *gateway* naquele momento e;
- finalmente, alterar a regra de marcação das conexões para utilizar a marca referente ao *gateway* preferencial do momento. Todos os pacotes de uma conexão terão a mesma marca utilizada no primeiro.

A busca na lista completa dos *gateways* é realizada periodicamente, tanto para que novos *gateways* sejam detectados automaticamente, quanto para que recursos do sistema (regras de roteamento) sejam liberados, caso um *gateway* deixe de existir. Esta lista de *gateways* é extraída durante o usual processamento sob a topologia, realizado periodicamente pelo OLSR.

## 7. Avaliação

Para avaliar o desempenho da solução DynTun, foram realizados testes comparativos em duas redes reais. As topologias utilizadas foram as das redes do projeto Remesh. Ambas apresentam dois nós com a capacidade de atuarem como *gateways*. As Figuras 3 e 4 mostram ilustrações das topologias. Os nós 7 e 17 representam os *gateways* da topologia externa, enquanto os nós 8 e 5 são os *gateways* da rede interna. Nas seções seguintes serão apresentados os testes realizados, seguidos dos seus respectivos resultados.

### 7.1. Semântica das conexões

O primeiro teste realizado teve como objetivo verificar a frequência com que acontecem as quebras de conexões ocasionadas pelos problemas abordados na Seção 2. Para tanto, o nó 18 da topologia interna foi utilizado como origem de vários fluxos de dados TCP em direção a um servidor externo à rede. A opção pelo nó 18 se deve a sua proximidade aos dois *gateways* disponíveis na rede. Dada a semelhança da qualidade dos enlaces dele com cada *gateway*, é esperado que ocorram trocas frequentes.

**Tabela 1. Distribuição dos tempos de duração das conexões quebradas.**

Intervalo (s)	Ocorrências
0 – 50	9
50 – 100	3
100 – 150	3
150 – 200	2
200 – 250	1
250 – 300	2
300 – 350	2
350 – 400	0
400 – 450	2
450 – 500	1
500 – 550	2

O experimento consistiu da abertura de 100 conexões TCP, iniciadas com intervalos de 1 minuto. A duração das conexões foi fixada em 900 segundos, totalizando 114 minutos de teste. A escolha destes parâmetros foi para simular o uso de conexões de aplicações P2P, que possuem relativa duração, quantidade de dados transferidos e paralelismo maiores do que outros tipos de aplicações, e portanto, é o tipo mais susceptível à sofrer quebras.

Ao longo do período testado, foram observados os tempos de duração de cada conexão e eventuais fechamentos causados pela troca de *gateway*. Pôde-se observar que 73 conexões foram concluídas com sucesso, enquanto as outras 27 foram quebradas. O tempo médio de duração das conexões quebradas foi de 178,9 segundos, com um desvio padrão de 170 segundos. A Tabela 1 mostra a distribuição dos tempos de quebra das conexões.

Este resultado mostra a necessidade da utilização de algum esquema de gerenciamento de *multi-homing* neste tipo de rede que preserve a semântica das conexões, como o DynTun. Mais de um quarto das conexões foram quebradas ao longo do teste, um número certamente não desprezível. Destas, 33,3% (9 conexões) tiveram duração de menos de 50 segundos.

## 7.2. Aumento da Capacidade

Esta seção tem por objetivo avaliar o aumento na capacidade de escoamento do tráfego da rede com a solução DynTun. Em outras palavras, deseja-se comparar a vazão obtida pelos nós usuários em conexões com servidores externos com e sem a utilização da solução.

Na topologia interna, utilizando os nós 2 e 18, o seguinte experimento foi realizado. Inicialmente, mediu-se a vazão de conexões TCP simultâneas de cada um dos dois nós até um servidor localizado fora da rede, utilizando-se apenas o nó 8 como *gateway*. Este teste foi então repetido utilizando-se como único *gateway* o nó 5. Estes dois testes tem o objetivo de esclarecer o ganho de desempenho obtido por realocar o *gateway* para uma melhor posição física, a fim de que o último teste com dois *gateways* seja possível reconhecer a real contribuição da solução DynTun.

A Tabela 2 mostra os resultados médios obtidos. Quando o único *gateway* dis-

**Tabela 2. Vazão obtida na rede interna em cada um dos três cenários (em Mbps).**

	<i>Gateway 5 apenas</i>	<i>Gateway 8 apenas</i>	DynTun
Nó 2	3,88	0,22	4,51
Nó 18	3,12	4,66	2,97
Agregado	7,00	4,88	7,48

ponível é o nó 8, o nó 2 necessita de três saltos para chegar à rede externa. Isso faz com que sua vazão diminua bastante, especialmente no cenário avaliado, onde existe a concorrência do nó 18, que é um dos três saltos necessários ao nó 2 para alcançar o *gateway*. Este teste demonstrou um uso injusto da rede, pois o nó 18 claramente priorizou a própria conexão em total detrimento da conexão do nó 2.

Por outro lado, quando o nó 5 é o único *gateway*, os usuários (nós 2 e 18) estão igualmente próximos ao *gateway*, e portanto ambos obtiveram bons resultados, mesmo concorrendo pelo acesso.

Quando os dois *gateways* são utilizados com a solução DynTun, cada fluxo TCP pôde ser iniciado por uma saída diferente. Isto permitiu um aumento de cerca de 7% na vazão agregada obtida. É notável perceber também que houve um aumento de mais de vinte vezes na vazão obtida pelo nó 2, em relação a situação em que apenas o nó 8 foi utilizado como *gateway*.

Em um segundo experimento realizado na rede externa, um fluxo TCP de 20 minutos foi iniciado e, 10 minutos depois, enquanto a primeira conexão ainda estava ativa, um segundo fluxo foi disparado. Ambos tiveram a mesma origem (o nó 4 da topologia) e o mesmo destino (um servidor fora da rede). Entretanto, quando este experimento foi executado na rede com a solução DynTun (ou seja, com dois *gateways*), a primeira conexão foi aberta através do nó 7, enquanto a segunda foi forçada a utilizar o nó 17 como *gateway*. A Figura 5 mostra os resultados obtidos.

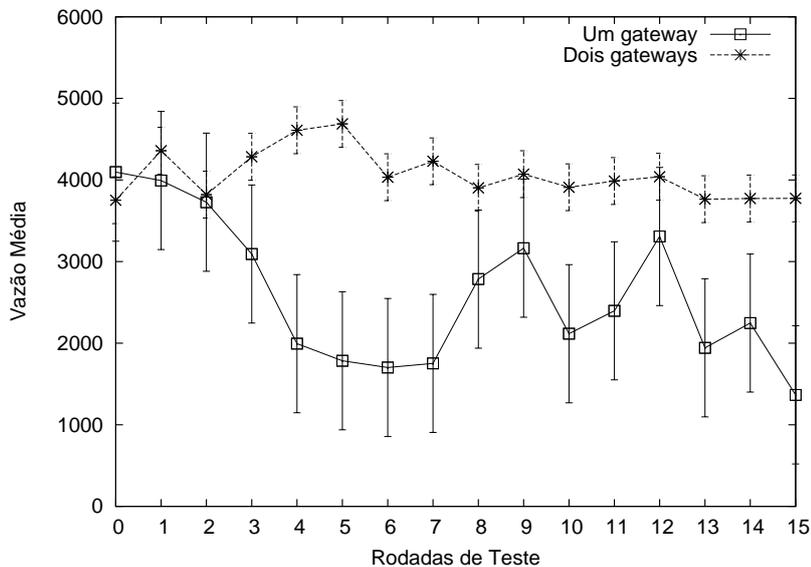
Este experimento foi repetido 16 vezes, totalizando 8 horas de teste para cada cenário (com um *gateway* ou dois). Desta forma, o gráfico da Figura 5 mostra os resultados agregados dos fluxos 1 e 2 em cada cenário. Em ambos os casos, foi considerado o desvio padrão dos experimentos.

Durante este teste, houve uma tendência clara de aumento na capacidade da rede com a utilização de dois *gateways*. Pode-se notar também que a variação verificada no cenário com dois *gateways* é consideravelmente menor em relação ao experimento com um *gateway* apenas. A Tabela 3 mostra os valores exatos de máximo, mínimo, média e desvio padrão dos pontos plotados na Figura 5.

**Tabela 3. Máximo, mínimo, média e desvio padrão dos cenários (todos em Kbps).**

Cenário	Máximo	Mínimo	Média	Desvio Padrão
Um gateway	4096	1366	2591,25	846,30
Dois gateways (DynTun)	4687	3752	4062,31	287,05

Os valores máximos foram relativamente próximos nos dois cenários, com uma diferença de menos de 600 Kbps. Entretanto, as diferenças entre os valores de mínimo



**Figura 5. Vazão agregada de saída da rede utilizando um e dois gateways.**

(2386 Kbps) e média (1471,06 Kbps) foram bastante acentuados. Como efeito, o desvio padrão da amostra do cenário com dois gateways foi quase três vezes menor do que no cenário com apenas um gateway. Estes valores se apresentam como um indicativo de melhora no escoamento do tráfego da rede tanto em termos de banda oferecida (na média, o teste apontou uma melhora de 56,77%), quanto em relação à estabilidade da mesma.

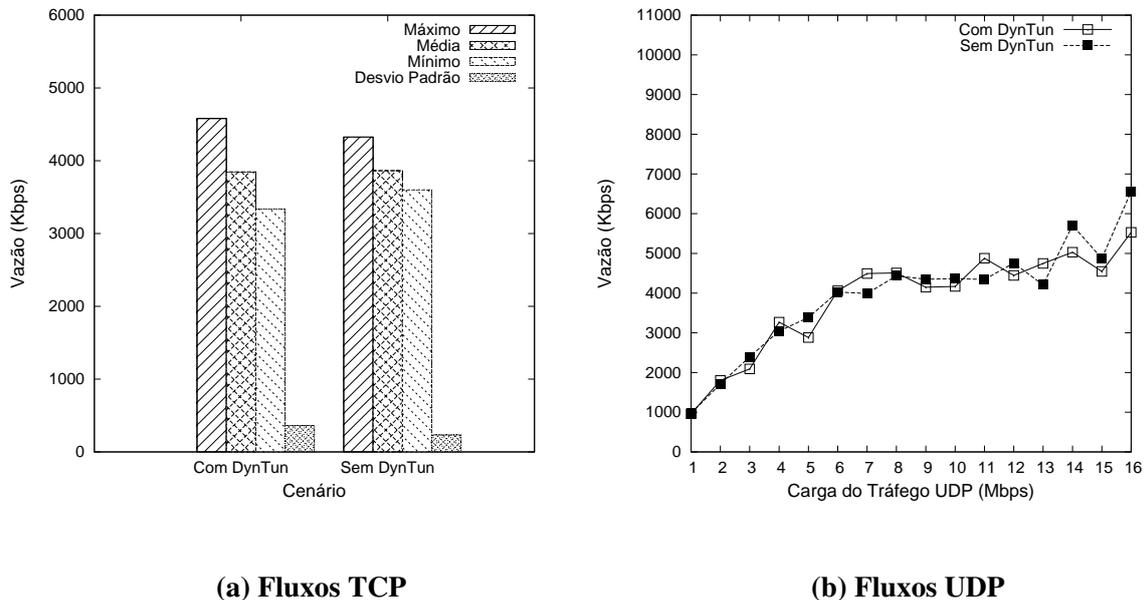
### 7.3. Impacto do Processamento

Na Seção 5 foi discutido de forma teórica o impacto da solução DynTun no processamento dos pacotes. O fato dos roteadores precisarem manter o estado das conexões de seus usuários poderia, a princípio, comprometer o desempenho obtido.

Para avaliar de maneira prática o real impacto da solução, foram realizados dois experimentos semelhantes. No primeiro, um fluxo de dados TCP foi enviado a partir do nó 4 da topologia até um servidor localizado fora da rede. Para este teste, o nó 17 foi desativado, fazendo com que a rede contasse com apenas um gateway (o nó 7). Neste cenário, o experimento foi realizado com e sem a utilização do *plugin* DynTun. Os resultados obtidos podem ser vistos no gráfico da Figura 6a. A média é referente a 12 repetições de 5 minutos para cada série.

Claramente a utilização do DynTun não afetou o desempenho neste cenário. Em ambas as situações, as médias foram bastante próximas com uma ligeira desvantagem para o cenário em que a solução foi utilizada. Ao se considerar o desvio padrão das medidas, pode-se dizer que o impacto foi desprezível.

No segundo experimento, ao invés de utilizar fluxos TCP, optou-se por utilizar fluxos UDP, variando a taxa de transmissão de 1Mbps a 16Mbps. Neste caso, novamente os experimentos tiveram 5 minutos de duração, porém foram realizados apenas uma vez.



**Figura 6. Demonstração do baixo impacto da solução sobre diferentes fluxos de dados.**

A Figura 6b resume os resultados.

Novamente os dois cenários obtiveram resultados bastante próximos para todas as taxas avaliadas. Isso demonstra que, embora a solução proposta provoque um aumento no processamento realizado pelos roteadores, na topologia avaliada este impacto não foi significativo diante do desempenho do canal de comunicação. Vale ressaltar ainda que as capacidades do *hardware* [6] dos roteadores utilizados neste experimento são bastante limitadas. O *clock* do processador é de 216 Mhz e a memória RAM disponível é de apenas 16MB, valores bastante baixos para os padrões atuais.

## 8. Conclusões e Trabalhos Futuros

Para permitir um crescimento incremental, redes em malha podem utilizar *multi-homing*, de forma a evitar uma queda de desempenho causada pelo aumento no número de saltos entre os usuários e o *gateway* da rede. Como demonstrado no primeiro teste na Seção 7, a simples combinação das técnicas de *multi-homing* e NAT pode resultar no efeito negativo, que consiste na quebra de uma grande quantidade das conexões. Tal efeito certamente prejudica a qualidade da rede percebida pelo usuário. Portanto, é necessário um mecanismo que evite este efeito negativo.

Neste artigo, foi proposta a solução DynTun, uma implementação real e funcional que resolve o problema da interação entre as técnicas de *multi-homing* e NAT, preservando a semântica das conexões dos usuários. Apesar do DynTun ter como foco evitar as quebras nas conexões, os testes da Seção 7.2 demonstraram um real aumento da capacidade da rede, tanto no quesito banda agregada de seus usuários quanto no significativo aumento, em uma ordem de magnitude, de desempenho dos usuários que anteriormente se encontravam a mais saltos dos *gateways*.

As Seções 5 e 6 possuem vários itens que levantam a suspeita de que o DynTun poderia impactar negativamente no desempenho da rede, devido ao processamento adicional necessário ao gerenciamento dos túneis. Contudo, o terceiro e último teste demonstrou o baixo impacto que a implementação do DynTun teve no desempenho da rede.

A implementação do DynTun, apesar de ter sido feita na rede Remesh, é também aplicável a outras redes, pois os critérios adotados são tão restritivos que facilitam sua adaptação às características de outras redes. Os dois itens que podem precisar de adaptação seriam a descoberta de *gateways* e a medição da qualidade das rotas.

Alguns trabalhos futuros incluem considerar a capacidade e carga de utilização de cada *gateway*, oferecer suporte a *QoS*, evitar que conexões de uma mesma aplicação sejam tuneladas para *gateways* distintos, por cause de restrições que algumas aplicações possuem, e adicionar criptografia nos túneis com o objetivo de proteger os dados dos usuários.

## Referências

- [1] Antônio Jorge Gomes Abelém, Célio Vinicius Neves de Albuquerque, Débora Christina Muchaluat-Saade, Elisângela Santana Aguiar, Jairo Lino Duarte, José Eduardo Mendonça da Fonseca e Luiz Claudio Schara Magalhães. *Redes mesh: Mobilidade, qualidade de serviço e comunicação em grupo. Minicurso do SBRC 2007, capítulo 2*, maio de 2007.
- [2] Aditya Akella, Bruce Maggs, Srinivasan Seshan, Anees Shaikh e Ramesh Sitaraman. A measurement-based analysis of multihoming. Em *SIGCOMM '03*, pp. 353–364, New York, NY, USA, 2003. ACM.
- [3] T. Bates e Y. Rekhter. Scalable support for multi-homed multi-provider connectivity. RFC Experimental 2260, Internet Engineering Task Force, janeiro de 1998.
- [4] Miguel E. M. Campista, Diego Gimenez Passos, Pedro M. Esposito, Igor M. Moraes, Célio Vinicius Neves de Albuquerque, Débora Muchaluat-Saade, Marcelo Gonçalves Rubinstein, Luís Henrique Maciel Kosmowski Costa e Otto Carlos Muniz Bandeira Duarte. Routing metrics and protocols for wireless mesh networks. *Network, IEEE*, 22(1):6–12, 2008.
- [5] D. De Couto, D. Aguayo, J. Bicket e R. Morris. A high-throughput path metric for multi-hop wireless routing. Em *MobiCom '03*, pp. 134–146, 2003.
- [6] Especificação do Hardware BCM4712. <http://www.broadcom.com/products/Wireless-LAN/802.11-Wireless-LAN-Solutions/BCM4712>, 2008. Acessado em 22-03-2008.
- [7] Jairo Duarte, Diego Passos, Rafael Valle, Etienne Oliveira, Débora C. Muchaluat-Saade e Célio Albuquerque. Management issues on wireless mesh networks. Em *5th Latin American Network Operations and Management Symposium (LANOMS 2007)*, 2007.
- [8] Engelstad P. E., Tonnesen A., Hafslund A. e Egeland G. Internet connectivity for multi-homed proactive ad hoc networks. *2004 IEEE International Conference on Communications*, 7:4050–4056, 2004.
- [9] K. Egevang e P. Francis. The IP network address translator (NAT). RFC Experimental 1631, Internet Engineering Task Force, maio de 1994.

- [10] D. Farinacci, T. Li, S. Hanks, D. Meyer e P. Traina. Generic routing encapsulation (GRE). RFC Experimental 2784, Internet Engineering Task Force, março de 2000.
- [11] David K. Goldenberg, Lili Qiuy, Haiyong Xie, Yang Richard Yang e Yin Zhang. Optimizing cost and performance for multihoming. Em *SIGCOMM '04*, pp. 79–92, New York, NY, USA, 2004. ACM.
- [12] Fanglu Guo, Jiawu Chen, Wei Li e Tzi cker Chiueh. Experiences in building a multihoming load balancing system. *INFOCOM 2004*, 2:1241– 1251, 2004.
- [13] Iptables e NetFilter. <http://www.iptables.org/>, 2007. Acessado em 03-02-2007.
- [14] T. J. Kniveton, J. Malinen, V. Devarapalli e C.Perkins. Mobile router tunneling protocol. Internet draft, Internet Engineering Task Force, 2002.
- [15] Débora C. Muchaluat-Saade, Célio Albuquerque, Luiz C. Schara Magalhães, Diego Passos, Jairo Duarte e Rafael Valle. Redes em malha: Solução de baixo custo para popularização do acesso à internet no brasil. Em *SBrT 2007*, 2007.
- [16] F5 Networks. <http://www.f5.com/>, 2007. Acessado em 03-02-2007.
- [17] Nortel Networks. <http://www.nortelnetworks.com/>, 2007. Acessado em 03-02-2007.
- [18] Diego Passos e Célio Albuquerque. Proposta, implementação e análise de uma métrica de roteamento multiplicativa para redes em malha sem fio. *Revista Eletrônica de Iniciação Científica (REIC)*.
- [19] Diego Passos, Douglas Teixeira, Debora Muchaluat-Saade, Luiz Magalhães e Celio Vinicius Neves de Albuquerque. Mesh network performance measurements. Em *5th International Information and Telecommunication Technologies Symposium*, 2006.
- [20] Radware. <http://www.radware.com/>, 2007. Acessado em 03-02-2007.
- [21] Paul Francis Ramakrishna. Ipn1: A NAT-extended internet architecture. Em *the 2001 SIGCOMM conference*, pp. 69–80. ACM, 2001.
- [22] S. Roch. Nortel's wireless mesh network solution: Pushing the boundaries of traditional wlan technology. Relatório Técnico Issue 2, Nortel Technical Journal, outubro de 2005.
- [23] RouteScience. <http://www.routescience.com>, 2007. Acessado em 03-10-2007.
- [24] Jaewook Shin, Haeryong Lee;, Jeehyeon Na, A. Park e S. Kim. Gateway discovery and routing in ad hoc networks with NAT-based internet connectivity. Em *Vehicular Technology Conference*, pp. 2883–2886, 2004.
- [25] Cisco Wireless Mesh Networking Solution. <http://www.cisco.com/go/wirelessmesh/>, março de 2007. Acessado em 03-03-2006.
- [26] L. Suciú, J.M. Bonnin, K. Guillouard e T. Ernst. Multiple Network Interfaces Management for Mobile Routers. *Proceedings of the International Conference on Intelligent Transportation Systems Telecommunications (ITST), Brest, June, 2005*.
- [27] N. Tsarmpopoulos, I. Kalavros e S. Lalis. A low-cost and simple-to-deploy peer-to-peer wireless network based on open source linux routers. Em *Proceedings of TRIDENT-COM'05*, pp. 92–97, 2005.