

Suporte à Mobilidade em Redes Mesh Sem Fio: estratégias comuns versus SDN

Italo Brito¹, Gustavo B. Figueiredo^{1*}

¹Departamento de Ciência da Computação – Universidade Federal da Bahia (UFBA)
Av. Adhemar de Barros, s/n – Campus Ondina – Salvador, BA – Brasil

{italo, gustavo}@dcc.ufba.br

Abstract. *Wireless mesh networks have been presented as a robust, scalable and low cost solution to provide connectivity in long distance areas. However, given its nature, routing strategies must support seamless mobility of nodes while enabling operation with a good performance and fast self-recovery from links fault. The routing approaches that meet these requirements are based on protocols usually used in ad-hoc networks (e.g. OLSR and B.A.T.M.A.N.) and, more recently, in the SDN paradigm (e.g. OpenWiMesh), each one of them having its own pros and cons. Therefore, it is important to evaluate the benefits and impacts of each approach, taking into account models and metrics inherent to the mobility. In this paper, we propose a simplified implementation of mobility support in OpenWiMesh and evaluate its performance compared to other protocols, using different mobility and data traffic models. The chosen metrics are based on packet loss, occupation of the links with control traffic and delay. Simulated results show that the SDN approach perform better than the classic protocols, beyond flexibility and programmability of SDN itself.*

Resumo. *As redes mesh têm sido apontadas como soluções robustas, escaláveis e de baixo custo para promover conectividade em áreas de longo alcance. Devido à natureza dessas redes, a estratégia de roteamento deve oferecer suporte à mobilidade transparente dos nós, além de operar com desempenho satisfatório e se auto-recuperar rapidamente de falhas nos enlaces que a compõe. As estratégias de roteamento que atendem a esses requisitos baseiam-se em protocolos comumente utilizados em redes ad-hoc (e.g. OLSR e B.A.T.M.A.N.) e, mais recentemente, no paradigma SDN (e.g. OpenWiMesh), cada um com seus prós e contras. Dessa forma, é importante comparar e avaliar os benefícios e impactos de cada estratégia levando em consideração modelos e métricas inerentes à mobilidade. Este trabalho propõe uma implementação simplificada de mobilidade no framework OpenWiMesh e avalia seu desempenho em relação à outras estratégias de roteamento, levando em consideração diferentes modelos de mobilidade e de geração de tráfego. As métricas utilizadas incluem a taxa de perda de pacotes, ocupação dos enlaces com tráfego de controle e atraso no encaminhamento de mensagens. Os resultados mostram que a abordagem SDN pode oferecer melhor desempenho que os protocolos comumente usados em redes ad-hoc, além da maior flexibilidade e programabilidade intrínsecas do paradigma.*

*TR-PGCOMP-003/2015. Technical Report. Computer Science Graduate Program. Federal University of Bahia.

1. Introdução

Uma rede em malha sem fio (WMN, do inglês *Wireless Mesh Network*) é uma rede sem fio de múltiplos saltos em que cada nó é capaz de se comunicar com todos os demais nós da rede, seja diretamente (com seus vizinhos) ou através do encaminhamento por nós intermediários [Akyildiz and Wang 2009]. As WMNs têm sido reiteradamente apontadas como a solução tecnológica de melhor custo benefício para a construção de uma plataforma de comunicação altamente escalável e capaz de promover conectividade a um custo relativamente baixo. Sua utilização traz benefícios potenciais como alta capacidade de comunicação, elevada disponibilidade e tolerância à falhas, além de propiciar uma rápida implantação de serviços de rede.

O roteamento nas WMNs é uma área de pesquisa bastante ativa, com trabalhos investigando diferentes estratégias de escolha e manutenção de caminhos na rede *mesh*. Em particular, um requisito importante dessas estratégias é o suporte à mobilidade dos nós, de forma que um nó móvel, ao movimentar-se, deve ter sua rota de acesso alterada (refletindo sua nova localização) sem interrupção de conectividade. Algumas dessas estratégias baseiam-se em protocolos comuns de redes *ad-hoc*, tais como OLSR [Jacquet et al. 2001] e B.A.T.M.A.N. [Johnson et al. 2008], e outras, mais recentemente, baseiam-se no paradigma de Redes Definidas por Software (SDN, do inglês *Software Defined Networks*), como é o caso do *framework* OpenWiMesh [Brito et al. 2014].

As estratégias comuns de roteamento *ad-hoc* estão bem consolidadas na literatura de redes *mesh*, baseando-se essencialmente em arquiteturas distribuídas, protocolos robustos e possibilidade de extensão, seja por modificação de parâmetros de configuração ou do desenvolvimento de plugins. Por outro lado, pesquisas recentes mostram que esse método de extensão é limitado [Dely et al. 2011], que é necessário maior programabilidade da rede e virtualização de recursos, permitindo assim alterações inclusive na lógica de funcionamento das WMNs (e.g. roteamento baseado em fluxos, engenharia de tráfego a partir de políticas administrativas, etc.). Essas questões são então endereçadas pelo paradigma SDN, e sua aplicação em redes *mesh* vem sendo discutida recentemente na literatura. A literatura, por outro lado, carece de estudos comparativos quanto aos benefícios e impactos de cada estratégia, principalmente levando em consideração modelos e métricas inerentes à mobilidade.

Nesse sentido, torna-se importante mensurar e comparar como os protocolos de roteamento, comuns e SDN, lidam com a mobilidade na rede. Tal avaliação deve levar em consideração a perspectiva de prover conectividade ininterrupta aos nós móveis, sem onerar a rede com alta taxa de ocupação por tráfego de controle ou alta latência na entrega de mensagens. É importante, ainda, que diferentes modelagens de padrão de movimentação e de geração de tráfego sejam consideradas, a fim de que os resultados sejam mais fiéis a ambientes reais. Este artigo apresenta um estudo comparativo de estratégias de roteamento em redes *mesh*, clássicas (OLSR e B.A.T.M.A.N.) e baseadas em SDN (OpenWiMesh), considerando requisitos de mobilidade e desempenho da rede.

O restante do trabalho está dividido da seguinte forma. A Seção 2 apresenta uma visão geral sobre roteamento em WMN. A Seção 4 discute modelos de mobilidade e de geração de tráfego, além de apresentar o ambiente usado para execução dos experimentos. Em seguida, na Seção 5, os resultados dos experimentos são apresentados e discutidos. A Seção 3 traz um breve resumo de trabalhos correlatos. Para que, enfim, na Seção 6

apresente-se as conclusões e trabalhos futuros.

2. Estratégias de Roteamento em Redes Mesh Sem Fio

O roteamento e encaminhamento em redes *mesh* sem fio é similar, de certa forma, ao roteamento em redes cabeadas. No entanto, alguns aspectos inerentes ao canal sem fio como, por exemplo, a atenuação, interferência, contenção, aliadas a características como mobilidade e capacidade de transmissão limitada dos nós, levam a uma alta dinamicidade da rede, que deve ser tratada pela estratégia de roteamento adotada. Tal estratégia deve ser, portanto, otimizada a fim de diminuir o tráfego de controle e reagir rapidamente a mudanças nos enlaces sem fio.

As estratégias de roteamento em redes *mesh* são classificadas geralmente em três categorias: protocolos pró-ativos, reativos e híbridos. Nos protocolos pró-ativos, computa-se as rotas para todos os nós durante a fase de inicialização do protocolo. Já nos protocolos reativos, as rotas são calculadas na medida em que são requisitadas pelas aplicações. Os protocolos reativos possuem a vantagem de não ocupar desnecessariamente a rede com tráfego de controle, ao passo que os protocolos pró-ativos geralmente apresentam menor atraso no estabelecimento de novos fluxos na rede, uma vez que os caminhos já estão definidos. Os protocolos híbridos tentam combinar as vantagens dos protocolos reativos e pró-ativos e adaptar seu funcionamento à diferentes cenários. Mais recentemente, uma quarta categoria passou a ser usada nesse contexto: as redes *mesh* definidas por software [Dely et al. 2011]. Essa é uma área de pesquisa promissora pois combina as características de robustez e escalabilidade das redes *mesh* com a programabilidade e controle centralizado do SDN.

Um ponto que merece especial atenção na escolha da estratégia de roteamento é o suporte à mobilidade transparente entre os nós, onde aplicações de tempo real não tem sua conectividade afetada em face a movimentação de um nó. Nesse sentido, dois requisitos são de importância destacável: mudança de ponto de acesso (processo também conhecido como *handoff*) sem interrupção de conectividade [Amir et al. 2006], e propagação da atualização de rotas depois do *handoff*. Este trabalho aborda principalmente os aspectos de atualização de rotas após o *handoff*.

Nesta seção serão apresentadas duas estratégias comumente usadas para roteamento em WMN e também uma estratégia baseada em SDN, as quais serão alvo de avaliação de desempenho deste trabalho. As estratégias clássicas usadas foram o OLSR e B.A.T.M.A.N., pois são as mais referenciadas pela literatura [Mishra et al. 2013, Abolhasan et al. 2009, Akyildiz and Wang 2009]. Já para a estratégia SDN, adotamos o OpenWiMesh [Brito et al. 2014], que já possui algoritmos de engenharia de tráfego na rede *mesh* e foi estendido para suportar, de forma simples, mobilidade dos nós.

2.1. OLSR

O protocolo OLSR (do inglês *Optimized Link State Routing*) [Jacquet et al. 2001] é um protocolo de roteamento pró-ativo, baseado em algoritmo otimizado de estado do enlace e amplamente utilizado em redes *mesh*. A principal vantagem do OLSR em relação aos algoritmos de estado de enlace clássicos é a existência de “retransmissores multiponto” (MPRs, do inglês *multipoint relay*). Os MPRs são nós especialmente selecionados para fazer a transmissão das informações de topologia, evitando que o processo de inundação

seja realizado a partir de todos os nós. Esta técnica minimiza a contenção e o número de mensagens de controle necessárias para estabelecimento da tabela de roteamento.

O protocolo faz uso de mensagens HELLO periódicas para descoberta de vizinhos e sinalização da seleção de MPRs. Além disso, utiliza também mensagens TC (controle de topologia) para troca de informações de topologia entre nós. Todos os nós, assim, conhecem a topologia e calculam localmente as melhores rotas para os demais nós.

O OLSR trata a mobilidade a partir do recálculo e atualização das rotas de saída e entrada de um nó. Para as rotas de saída, deve-se definir qual vizinho será o novo gateway para cada um dos possíveis destinos na rede, enquanto que para as rotas de entrada é necessário atualizar a tabela de roteamento dos demais nós da rede com o novo caminho através do qual o nó móvel é alcançável. A atualização das rotas de saída é baseada nas mensagens de HELLO, que são enviadas a cada t_{hello} segundos. Para viabilizar mobilidade transparente, é preciso que o valor de t_{hello} seja baixo (e.g. 500ms [Amir et al. 2006]), porém mesmo nessa configuração podem ser necessários múltiplos ciclos de mensagens HELLO para estabelecer enlaces bidirecionais. A atualização das rotas de entrada depende da propagação das informações de topologia, que ocorre através das mensagens de TC a cada t_{tc} segundos, e do recálculo da tabela de rotas em todos os nós.

Em ambos os casos, t_{hello} e t_{tc} poderiam ser reduzidos a grandezas muito baixas para permitir mobilidade transparente, porém dessa forma o tráfego de controle traria alto impacto à rede devido a contenção para controle de colisão.

2.2. B.A.T.M.A.N.

O protocolo B.A.T.M.A.N. (do inglês *Better Approach to Mobile Ad-Hoc Networking*) [Johnson et al. 2008] também faz uso de uma estratégia pró-ativa, todavia a técnica de distribuição da topologia e seleção de rotas é melhor alinhada com a limitação de recursos de hardware geralmente disponíveis nos nós da rede *mesh* sem fio. No B.A.T.M.A.N., os nós não conhecem toda a topologia da rede. Ao invés disso, eles conhecem apenas o vizinho que possui a melhor rota para determinado destino. Adicionalmente, o algoritmo de seleção de rotas baseia-se no fato de que os melhores enlaces entregarão as mensagens de forma mais rápida e confiável. Assim, a seleção de melhores rotas se dá de forma natural, sem necessidade de um cálculo baseado na topologia em cada nó.

De forma geral, cada nó envia, por difusão, *mensagens de originadores* (OGMs, do inglês *originators messages*), informando aos vizinhos sobre a sua existência. Ao receber uma mensagem OGM, o nó retransmite essa mensagem para seus vizinhos. Naturalmente, espera-se que as mensagens encaminhadas por enlaces de maior qualidade cheguem primeiro, por isso a primeira mensagem recebida torna-se o melhor caminho para o originador daquela mensagem e as demais são descartadas. Apenas a primeira OGM recebida para determinado destino (melhor caminho) é retransmitida. Existe, portanto, uma inundação da rede com OGMs. Ademais, uma checagem bidirecional de cada enlace é feita a fim de verificar se o enlace detectado pode ser usado em ambas as direções.

As mensagens OGM são usadas pelo B.A.T.M.A.N. tanto para detectar mudanças quanto qualidade dos enlaces. Quando um nó se move na rede, seus vizinhos levam no mínimo dois ciclos de OGM para atualizar as tabelas de rota, enquanto que o restante da rede depende da difusão dessas OGMs para propagar essa atualização

[Johnson et al. 2008]. Isso implica em tempos razoáveis para atualização das rotas de saída, mas em desempenho ruim para rotas de entrada.

Uma implementação bastante utilizada do B.A.T.M.A.N. é o BATMAN-Adv (B.A.T.M.A.N. *Advanced*) [OpenMesh Project 2008], que opera em camada 2, possui mecanismos de controle de loop e diversas outras funcionalidades.

2.3. OpenWiMesh

O OpenWiMesh é um *framework* para roteamento e engenharia de tráfego em redes *mesh* sem fio, através de uma abordagem baseada em Redes Definidas por Software (SDN) [Brito et al. 2014]. O OpenWiMesh implementa o roteamento na rede *mesh* a partir de uma abordagem centralizada, alinhada com o conceito de SDN, definindo caminhos sob demanda a partir de um grafo que modela a rede. Valendo-se da programabilidade da rede proporcionada pelo SDN, o OpenWiMesh permite a definição de diversas estratégias para escolha dos melhores caminhos. Em particular, três estratégias são pré-distribuídas em conjunto com o OpenWiMesh: HC (do inglês *Hop-Count*), que defini a melhor rota baseado na menor quantidade de saltos; HLRB (do inglês *Highest Link Residual Bandwidth*), onde o melhor caminho é escolhido como aquele cuja banda residual é maior; e HLRB-SHC (do inglês *Highest Link Residual Bandwidth in Same Hop Count*), que combina as duas estratégias anteriores a fim de escolher caminhos menos congestionados porém sem aumentar significativamente a quantidade de saltos. Neste trabalho foi utilizado algoritmo baseado em contador de saltos.

O grafo da rede é mantido de forma centralizada em um nó com maior capacidade computacional definido como controlador, responsável pelo cálculo de rotas e gerencia da rede *mesh*. Dessa maneira os demais nós da rede *mesh* não são onerados com funções de cálculo de rotas ou armazenamento da topologia. Pelo contrário, cada nó de *backbone* possui apenas um componente local, chamado de *graphClient*, cuja função é coletar informações do meio sem fio (lista de vizinhos, potência de sinal de cada vizinho, SINR, etc) e enviá-las ao controlador. O controlador então atualiza o grafo da rede, que será usado para definição das rotas futuras.

O funcionamento do OpenWiMesh pode ser dividido em duas fases: uma fase de *setup* da rede e uma fase de criação e manutenção de rotas para aplicações. Na fase de *setup* os nós tem sua conexão estabelecida com o controlador, através do protocolo Openflow [McKeown et al. 2008] em modo *in-band*. Já na fase de criação e manutenção de rotas de aplicações, os nós *mesh* enviam, sobre demanda, requisições de encaminhamento ao controlador (*Packet-In*), que faz uma busca no grafo da rede a partir da estratégia de roteamento definida.

Na versão original do OpenWiMesh não havia tratamento para nós móveis, sendo necessário que o próprio nó desse início à reconexão após o *timeout* do sistema, gerando por vezes mais de 30 segundos de desconexão. Neste trabalho foi desenvolvida uma extensão do *framework* para adicionar gerenciamento de mobilidade dos nós. A extensão proposta, conforme Figura 1, atua na fase de criação e manutenção de rotas de aplicações, onde o controlador monitora a lista de informações do meio sem fio enviada pelo *graphClient*, e verifica se houveram desassociações cujo enlace estava em uso por alguma aplicação. Nesse caso é iniciada uma ação de re-roteamento na rede, atualizando a tabela Openflow dos nós que utilizavam aquela aplicação (rota de entrada) e

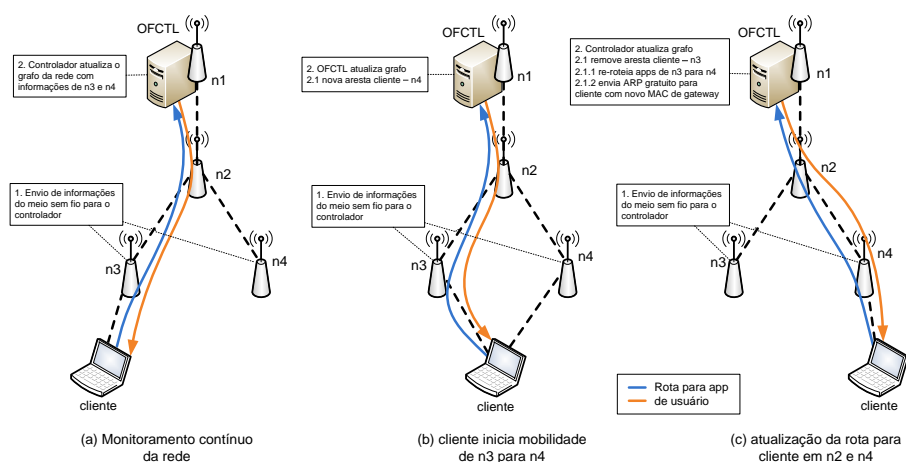


Figura 1. Gerenciamento de mobilidade no OpenWiMesh

enviando uma resposta ARP gratuita ao nó móvel (rota de saída). O envio de mensagem ARP gratuita é importante para que o nó móvel atualize sua tabela ARP substituindo a(s) ocorrência(s) do MAC do nó com o qual ele estava associado anteriormente para o MAC do novo nó de associação. Dessa maneira o *handoff* do nó móvel é gerenciado pelo *framework* OpenWiMesh e não pelo nó que está se movendo, viabilizando a implementação de diversas estratégias de detecção e gerenciando de *handoff* pelo controlador da rede.

Os nós móveis podem ser clientes e até mesmo nós de *backbone*. Caso o nó móvel seja um nó de *backbone*, apenas atualizar o MAC do controlador é suficiente, pois as demais entradas da tabela ARP direcionam para o MAC daquele próprio nó (as regras Openflow reescrevem o MAC para o próximo salto). Porém em caso de tratar-se de nó cliente, serão enviadas tantas respostas ARP gratuitas quanto forem necessárias para aplicações que utilizavam aquele enlace. A gerencia de mobilidade, dessa maneira, propaga as atualizações de rota de forma rápida e eficiente.

O código-fonte do OpenWiMesh, bem como a extensão desenvolvida neste trabalho, estão disponíveis no site do projeto [OpenWiMesh 2013].

3. Trabalhos relacionados

Alguns trabalhos estão disponíveis na literatura avaliando protocolos de roteamento em redes *mesh*, porém nenhum deles compara abordagens clássicas com abordagens SDN. O trabalho [Mishra et al. 2013] compara os efeitos de diferentes modelos de mobilidade e geração de tráfego com base em protocolos de redes *ad-hoc* AODV, OLSR e ZRP, levando em consideração métricas como atraso fim-a-fim, jitter e vazão. Embora faça uso de dois modelos de mobilidade, os impactos da mobilidade na conectividade dos nós não são avaliados. Naquele trabalho faz-se uso de simulação, através da ferramenta QualNet.

Já o trabalho [Abolhasan et al. 2009] investiga o desempenho de protocolos de roteamento pró-ativos em ambiente real, com testbed de rede *ad-hoc* construído em área fechada. Os protocolos comparados são OLSR, B.A.T.M.A.N. e BABEL. As métricas daquela avaliação foram vazão, taxa de entrega de pacotes e latência para recuperação de falhas em enlaces. Este trabalho não leva em consideração a característica de mobilidade das redes *ad-hoc*. Os resultados daquele trabalho estão alinhados com aqueles aqui apre-

sentados: o B.A.T.M.A.N. possui menor tempo de recuperação de falhas em enlaces que o OLSR (que aqui é representado pela menor taxa de perda de pacotes).

O trabalho [Dely et al. 2011] foi o primeiro a integrar Openflow em WMNs, propondo uma arquitetura que viabilize tal integração e provendo a funcionalidade de roteamento baseado em fluxos. Como estudo de caso foi implementado uma solução de mobilidade simplificada entre nós *mesh* e avaliado em uma rede real de testbed (KAU-Mesh). Uma das métricas avaliadas foi a sobrecarga do tráfego de controle, comparando OLSR e a proposta Openflow, a medida que novos fluxos são iniciados. O resultado daquela métrica mostrou que o Openflow apresenta crescimento linear com a quantidade de novos fluxos por segundo, enquanto o OLSR permanece constante (característica esperada dado sua configuração de rotas pró-ativa). Outra métrica avaliada naquele trabalho foi o tempo de desconexão (*handover*) do nó *mesh* móvel. Alcançou-se, naquela ocasião, um tempo de interrupção de 210ms na média, com variação entre 50ms e 270ms. Não estão disponíveis informações sobre o modelo de mobilidade utilizado, enquanto que o padrão de tráfego considera um intervalo entre pacotes de 10ms (100pps) e transporte UDP, via ferramenta *iperf*.

4. Cenário de Avaliação

4.1. Modelos de Mobilidade

O modelo de mobilidade define o padrão de movimentação dos nós na rede. Existem basicamente dois tipos de modelos de mobilidade: mobilidade de entidade ou individual e mobilidade em grupo [Camp et al. 2002]. Para o tipo de mobilidade individual existem diversos modelos disponíveis, tais como *Random Waypoint*, *Random walk*, *Gauss-Markov*, etc. Já na mobilidade em grupo pode-se citar os modelos *Exponential Correlated Random Mobility*, *Nomadic Community Mobility* e, mais utilizado, *Reference Point Group Mobility Model*. Neste trabalho serão utilizados os modelos *Random Waypoint (RWP)* e *Reference Point Group Mobility Model (RPGM)*, ilustrados na Figura 2 e detalhados a seguir.

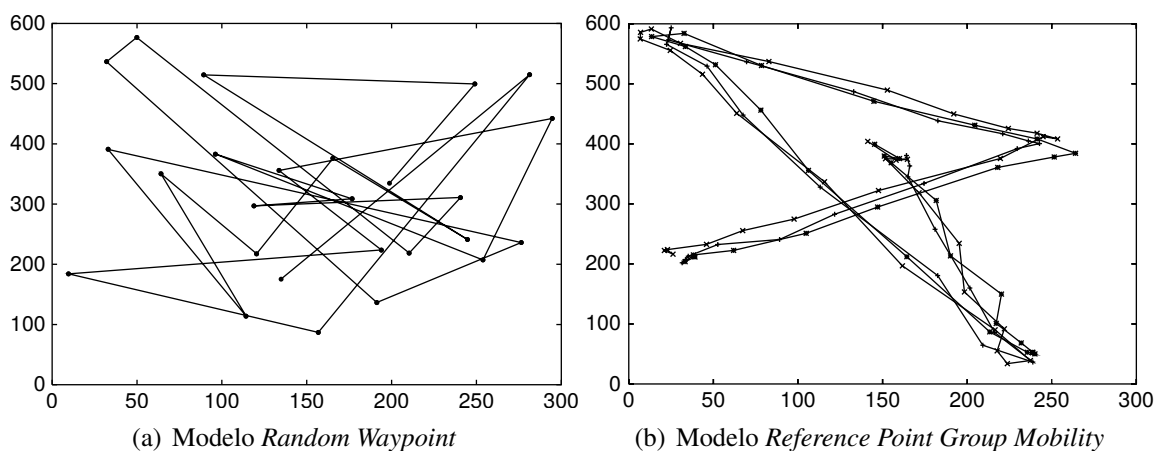


Figura 2. Padrão de movimentação por modelo de mobilidade [Camp et al. 2002]

O modelo RWP é um modelo de mobilidade simples, onde o padrão de movimentação de um nó é modelado como a escolha de um destino aleatório

dentro da área de execução e de uma velocidade uniformemente distribuída entre [minspeed, maxspeed]. Cada nó pode ainda pausar em determinado local por certo período de tempo. O comportamento do modelo RWP pode ser visto na Figura 2a. Este é um modelo de mobilidade amplamente utilizado em estudos de redes *ad-hoc* [Abolhasan et al. 2009, Mishra et al. 2013]. Por vezes esse modelo é simplificado, eliminando-se, por exemplo, os períodos de pausa, abordagem adotada neste trabalho. Alguns aspectos importantes no uso deste modelo são seus parâmetros de configuração (velocidades mínima e máxima, período de pausas, área), a disposição inicial dos nós na área de execução e período de execução do experimento [Camp et al. 2002]. Acerca da disposição inicial dos nós, neste trabalho os nós foram distribuídos uniformemente na área de execução. Em se tratando do período de execução, recomenda-se ignorar os primeiros segundos devido a alta variação no grau de conectividade dos nós. Neste trabalho foram ignorados os primeiros 900 segundos do modelo de mobilidade.

Já o RPGM representa a movimentação de um grupo de nós em determinada direção, bem como uma movimentação individual dentro do grupo. O RPGM é apropriado para modelar situações de desastres, por exemplo, onde grupos de pessoas se movimentam juntas em determinada direção (saídas de emergência) ou ainda em cenários de cooperação mútua [Camp et al. 2002]. Os movimentos do grupo são guiados por um ponto lógico central do grupo, chamado de vetor de movimento GM. Este vetor de movimento caracteriza o movimento de todo o grupo, incluindo seu destino e velocidade. Dentro do grupo, os nós se movem aleatoriamente em relação a seu ponto referencial. A Figura 2b ilustra a movimentação de três nós com o modelo RPGM. Novamente, a posição inicial dos nós, composição dos grupos e velocidade, são configurações importantes para o modelo. Neste trabalho adotou-se uma distribuição uniforme dos nós na área de execução (assim como no modelo RWP) e grupos de tamanho médio de três nós.

4.2. Modelos de Tráfego

Para avaliar as estratégias de maneira mais realística e sobre condições extremas, foram utilizados modelos estocásticos para representar o padrão de geração de tráfego na rede durante o período de execução dos testes. Nesse trabalho fez-se uso de modelos de geração de tráfego baseados em taxa de bits constante (CBR, do inglês *Constant Bit Rate*) e taxa de bits variável (VBR, do inglês *Variable Bit Rate*). Em particular, duas variáveis aleatórias foram utilizadas para caracterizar o modelo de tráfego: o tamanho do pacote (PS, do inglês *Packet Size*) e o intervalo entre o envio de pacotes (IDT, do inglês *Inter Departure Time*).

4.3. Ambiente de Avaliação

Neste trabalho optou-se pela experimentação através de emulação devido a dificuldade de montagem de ambiente real, isolamento de efeitos de interferência externos e replicabilidade. A ferramenta de emulação utilizada foi o CORE [Ahrenholz 2010], que faz uso de recursos de virtualização do kernel do Linux (*network namespaces*) em conjunto com um modelo de rede sem fio 802.11 simplificado.

Os experimentos foram conduzidos tomando-se como fatores a topologia da rede *mesh*, modelos de geração de tráfego e modelos de mobilidade. A respeito das topologias, foram criados cenários com 10, 20 e 30 nós, onde o diâmetro da rede em cada um deles foi 2, 4 e 6 saltos, respectivamente. Houve também uma preocupação de manter o

Tabela 1. Parâmetros de emulação

Parâmetro de emulação	Valor
Topologias	T1: 10 nós, diâmetro 4, dimensão 240x360m T2: 20 nós, diâmetro 6, dimensão 400x360m T3: 30 nós, diâmetro 8, dimensão 560x360m
Disposição dos nós na topologia	Uniformemente distribuídos
Modelo de mobilidade	RWP e RPGM (ambos sem pausa)
Modelo de Tráfego	CBR e VBR
Tempo de execução	180s (+ tempo de convergência dos protocolos)
Modelo de perda de propagação	<i>Free Space Path Loss</i>
Modelo de antena	Omni-directional
Sensibilidade de recepção	-90dBm
Potência de transmissão	20dBm
Ganho da Antena	TX 1 dBi e RX 1 dBi
Camada física	IEEE 802.11ag (canal 11, frequência 2.4 GHz)

mesmo grau de conectividade dos nós, definindo, para isso, áreas de execução com tamanho proporcional à quantidade de nós. Em todas as topologias foram usados 50% de nós de *backbone* estacionários e 50% de nós clientes móveis. Os nós de *backbone* foram posicionados para que houvesse cobertura total da área de execução. O cálculo do diâmetro e grau considera apenas os nós de *backbone*.

Os modelos de mobilidade adotados foram RWP e RPGM. O cenário de mobilidade teve duração de 120 segundos, iniciados após a convergência do protocolo de roteamento e um período monótono (sem tráfego e sem mobilidade) de 60s. Os nós foram configurados para movimentar-se a uma velocidade entre 2m/s e 6m/s, sem período de pausa (cenário com ampla mobilidade). A ferramenta utilizada para gerar os scripts de mobilidade foi o *BonnMotion* [Aschenbruck et al. 2010].

No que tange ao modelo de geração de tráfego, para o CBR tanto o PS quanto o IDT foram mantidos constantes, fixados em 50 pps e 1400 bytes respectivamente, ao passo que no VBR adotou-se uma distribuição uniforme para o PS, variando entre 64 e 1400 bytes, e uma distribuições de Poisson para IDT com média de 50 pps. Em ambos os modelos foi gerado um fluxo de dados UDP a partir de um nó central da topologia (simulando um nó de *backbone* com acesso à Internet) para todos os outros nós móveis. A ferramenta utilizada como gerador de tráfego foi o D-ITG (*Distributed Internet Traffic Generator*) [Avallone et al. 2004]. A Tabela 4.3 apresenta um resumo da configuração do ambiente de execução.

As topologias, modelos de mobilidade, scripts dos experimentos e outros códigos estão disponíveis no repositório do projeto OpenWiMesh (<https://nuvem.pop-ba.rnp.br/gitlab/ufba/openwimesh>).

5. Análise dos Resultados

O estudo comparativo entre as estratégias de roteamento em WMN deu-se pela avaliação do OLSR, BATMAN-Adv e OpenWiMesh com relação às métricas i) **perda de pacotes**, tomando como fatores a topologia (quantidade de nós, diâmetro da rede), modelo de

tráfego e modelo de mobilidade; ii) **tempo de resposta** (latência), baseado no tempo de ida e volta (RTT) tanto para o pior caso (*slowpath*) quanto para o caso médio (*fastpath*); e iii) **ocupação dos enlaces com tráfego de controle**.

Todos os experimentos foram repetidos 30 vezes. Cada execução durou cerca de 240 segundos. As três estratégias aqui avaliadas foram configuradas de forma equivalente, com intervalo entre atualizações de 1 segundo. O intervalo de confiança foi calculado considerando-se um nível de confiança de 95%.

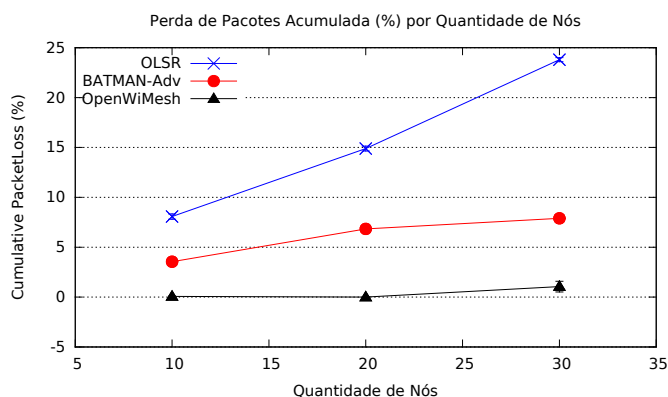


Figura 3. Perda de Pacotes variando-se a Topologia

A primeira métrica avaliada foi a perda de pacotes durante a movimentação dos nós pela rede. Vale reforçar que todas as topologias foram configuradas para que não houvesse áreas sem cobertura. A Figura 3 apresenta o comportamento dos protocolos com a variação das topologias, mantendo fixos o modelo de mobilidade RWP e de tráfego CBR. Os protocolos BATMAN-Adv e OLSR apresentaram elevado grau de perda de pacotes, chegando a aproximadamente 8 e 25% de perda, respectivamente, para topologias com 30 nós. Já o OpenWiMesh manteve um nível de perda de pacotes baixo, próximo a 0% nas topologias com 10 e 20 nós e atingindo um intervalo de confiança de $[0.51, 1.59]$ na topologia com 30 nós, com nível de significância de 0.05. Estes resultados refletem a dificuldade de propagação da atualização de rotas para os demais nós da rede, principalmente no caso do OLSR que depende da sincronização das informações de topologia por todos os nós. O B.A.T.M.A.N., por não necessitar que todo nó conheça a topologia completa, tem desempenho melhor que o OLSR, porém ainda assim depende da propagação das OGMs para atualizar as rotas de entrada. Por outro lado, o OpenWiMesh consegue responder de forma mais rápida à mobilidade, devido a sua gerência centralizada e informações do grafo da rede.

Avaliou-se também o impacto dos modelos de mobilidade na entrega de pacotes, conforme Figura 4. Pode-se observar que o OLSR apresenta significativa perda de pacotes tanto com RWP quanto com RPGM, divergindo em apenas cerca de 1%. O BATMAN-Adv também apresenta perda de pacotes significativa e também com baixa divergência entre os modelos de mobilidade. Chama atenção, todavia, a diferença visual dos resultados do OpenWiMesh. No entanto, observando os valores do intervalo de confiança, essa diferença é de apenas 0.04%, com ambos os modelos gerando uma perda máxima de 0.3%. Para essa métrica foi utilizada a topologia com 20 nós e modelo de tráfego CBR.

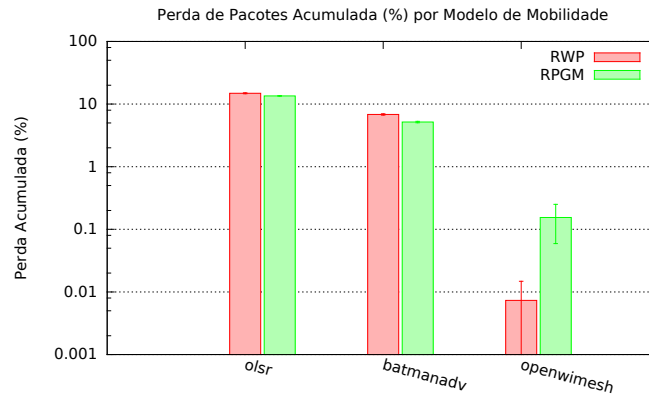


Figura 4. Perda de Pacotes variando-se o Modelo de Mobilidade

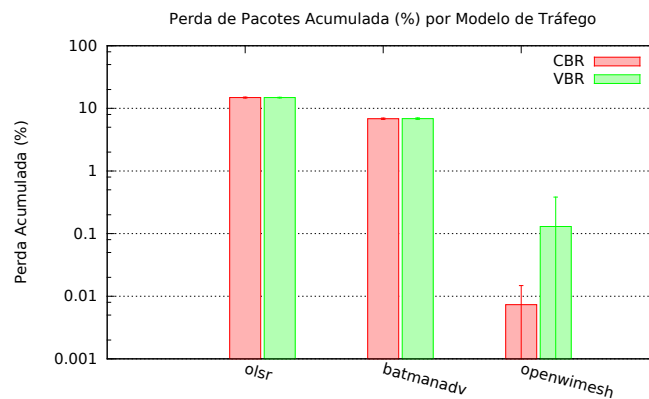


Figura 5. Perda de Pacotes variando-se o Modelo de Tráfego

A variação do modelo de tráfego também não impôs significativa diferença em relação ao cenário anterior, conforme pode ser visto na Figura 5.

Outra métrica avaliada foi o tempo de resposta na transmissão de dados, sendo utilizada a topologia com 20 nós e empregando o modelo de mobilidade RWP. O tempo de resposta levou em consideração o tempo de ida e volta do pacote, analisando o tempo do primeiro pacote, também conhecido como *slowpath*, e a média dos demais pacotes, chamado de *fastpath*. O primeiro pacote é considerado *slowpath* pois ele pode necessitar de alguma configuração do protocolo de roteamento, antes que o fluxo esteja apto a ser encaminhado. Esse é um comportamento normal de protocolos sobre demanda, como o OpenWiMesh, porém mesmo o BATMAN-Adv pode ter um *slowpath* significativo devido a resolução ARP no sistema operacional.

A Figura 6 mostra o resultado da avaliação do tempo de resposta. O resultado do *fastpath* apresenta um equilíbrio entre as estratégias, com tempos de resposta de aproximadamente 0.01 ms, com intervalo de confiança de 0.01 ms e nível de confiança de 95%. Esse resultado, por outro lado, precisa ser investigado levando-se em consideração outros fatores, como caminhos congestionados e considerando múltiplos fluxos simultâneos na rede. Já o resultado do *slowpath* apresenta uma vantagem do OLSR que consegue encaminhar o primeiro pacote de um fluxo com atraso menor que 1 ms. O resultado

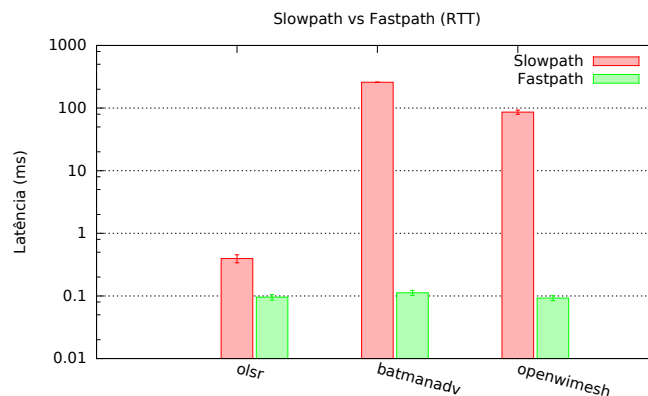


Figura 6. Tempo de Resposta - *Slowpath vs Fastpath*

do BATMAN-Adv, embora sobremaneira elevado, com 259 ms (intervalo de confiança 0.9ms), representa o comportamento convencional da maioria dos protocolos de roteamento, que depende de uma resolução ARP. Nesse sentido, o OpenWiMesh apresenta um resultado significativamente melhor, com 86ms (intervalo de confiança de 6ms). Além do tempo menor, um resultado indireto do OpenWiMesh sobre os demais é que a requisição ARP não é enviada em difusão para os outros nós da rede. Pelo contrário, ela é capturada logo na origem e respondida pelo controlador.

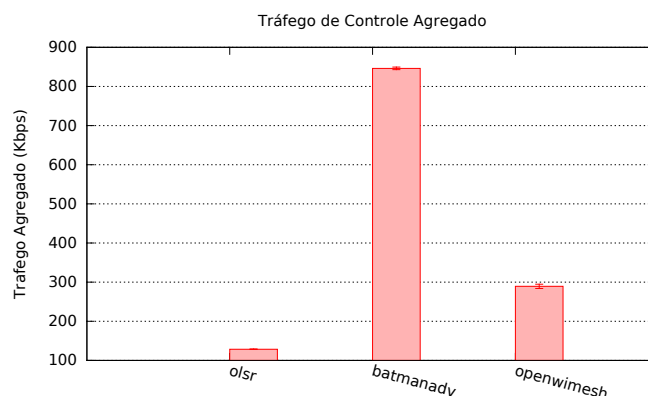


Figura 7. Tráfego de Controle Agregado

Neste experimento o objetivo é avaliar qual a medida de tráfego de controle que é gerado pelos protocolos de roteamento. Idealmente a rede não deve ser significativamente ocupada pelo tráfego de controle. O protocolo de roteamento deve encontrar um equilíbrio entre detectar rapidamente movimentos ou falhas nos enlaces e nós, porém sem sobrecarregar a rede. Os resultados dessa métrica estão dispostos na Figura 7, que apresenta o tráfego agregado de todos os nós da topologia (20 nós) antes que o gerador de tráfego injete pacotes na rede. O OLSR apresenta baixa taxa de ocupação da rede com controle e o OpenWiMesh apresenta uma média de tráfego bem aceitável, com valor agregado de aproximadamente 290 Kbps (intervalo de confiança de 5 Kbps). O BATMAN-Adv, por outro lado, apresentou uma significativa taxa de ocupação da rede com tráfego de controle. Em um ambiente real isso pode gerar contenção, podendo impactar negativamente nas aplicações da rede.

Este trabalho não considera requisições de rotas (ativação de novas aplicações) na computação da sobrecarga do tráfego de controle.

6. Conclusões e Trabalhos Futuros

Este trabalho apresentou um estudo comparativo de estratégias de roteamento em redes *mesh*, considerando abordagens comumente usadas como OLSR e BATMAN e uma abordagem SDN (OpenWiMesh). O comparativo levou em consideração requisitos como mobilidade e desempenho da rede. Os resultados mostram que o OpenWiMesh apresenta menor perda de pacotes com a mobilidade de nós, com taxas muito próximas a 0% de perda. Tal sucesso na entrega de pacotes não implica em maior oneração para a rede, uma vez que o tráfego de controle apresentou limites aceitáveis, comparado com os outros protocolos e com a capacidade de transmissão dos nós, e a latência no encaminhamento de pacotes esteve estatisticamente igual aos demais protocolos considerando o *fastpath*. Já a latência do *slowpath*, apesar de alta, está abaixo do tempo de estratégias baseadas em requisições ARP, como o BATMAN-Adv. Além disso, a resolução ARP do OpenWiMesh é mais eficiente que as abordagens convencionais, uma vez que evita que a requisição seja difundida por toda a rede (*broadcast*).

Como proposta futuras, este trabalho pode ser estendido para contemplar também a avaliação em ambientes reais, considerando aspectos como contenção e interferência do canal sem fio, além de incluir novas métricas como a sobrecarga do tráfego de controle face a requisições simultâneas de novos fluxos e a distribuição de carga na rede. Convém, ainda, investir na otimização dos protocolos utilizados para cenários de alta mobilidade e experimentar técnicas de gerenciamento de *handoff*.

Referências

- [Abolhasan et al. 2009] Abolhasan, M., Hagelstein, B., and Wang, J.-P. (2009). Real-world performance of current proactive multi-hop mesh protocols. In *15th Asia-Pacific Conference on Communications (APCC)*, pages 44–47. IEEE.
- [Ahrenholz 2010] Ahrenholz, J. (2010). Comparison of core network emulation platforms. In *MILITARY COMMUNICATIONS CONFERENCE, 2010-MILCOM 2010*, pages 166–171. IEEE.
- [Akyildiz and Wang 2009] Akyildiz, I. and Wang, X. (2009). *Wireless mesh networks*, volume 3. John Wiley & Sons.
- [Amir et al. 2006] Amir, Y., Danilov, C., Hilsdale, M., Musiloiu-Elefteri, R., and Rivera, N. (2006). Fast handoff for seamless wireless mesh networks. In *Proceedings of the 4th international conference on Mobile systems, applications and services*, pages 83–95. ACM.
- [Aschenbruck et al. 2010] Aschenbruck, N., Ernst, R., Gerhards-Padilla, E., and Schwamborn, M. (2010). Bonnmotion: a mobility scenario generation and analysis tool. In *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*, page 51. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).
- [Avallone et al. 2004] Avallone, S., Guadagno, S., Emma, D., Pescapè, A., and Ventre, G. (2004). D-itg distributed internet traffic generator. In *Quantitative Evaluation of Sys-*

- tems, 2004. QEST 2004. Proceedings. First International Conference on the*, pages 316–317. IEEE.
- [Brito et al. 2014] Brito, I., Gramacho, S., Ferreira, I., Nazaré, M., Sampaio, L., and Figueiredo, G. (2014). Openwimesh: um framework para redes mesh sem fio definidas por software. In *32th Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*, pages 413–426.
- [Camp et al. 2002] Camp, T., Boleng, J., and Davies, V. (2002). A survey of mobility models for ad hoc network research. *Wireless communications and mobile computing*, 2(5):483–502.
- [Dely et al. 2011] Dely, P., Kessler, A., and Bayer, N. (2011). Openflow for wireless mesh networks. In *Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on*, pages 1–6. IEEE.
- [Jacquet et al. 2001] Jacquet, P., Muhlethaler, P., Clausen, T., Laouiti, A., Qayyum, A., and Viennot, L. (2001). Optimized link state routing protocol for ad hoc networks. In *IEEE International Multi Topic Conference (IEEE INMIC). Technology for the 21st Century. Proceedings*.
- [Johnson et al. 2008] Johnson, D., Ntlatlapa, N., and Aichele, C. (2008). Simple pragmatic approach to mesh routing using batman. In *2nd IFIP International Symposium on Wireless Communications and Information Technology in Developing Countries (CSIR)*, pages 6–7.
- [McKeown et al. 2008] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). Openflow: enabling innovation in campus networks. In *ACM SIGCOMM Computer Communication Review*.
- [Mishra et al. 2013] Mishra, S., Xavier, A. D., and Jaisawal, A. (2013). Effect of mobility and different data traffic in wireless ad-hoc network through qualnet. In *International Journal of Engineering and Advanced Technology (IJEAT)*, volume 2, pages 364 – 368.
- [OpenMesh Project 2008] OpenMesh Project (2008). B.a.t.m.a.n. advanced wiki. URL: <http://www.open-mesh.org/projects/batman-adv/wiki>.
- [OpenWiMesh 2013] OpenWiMesh (2013). Site do projeto OpenWiMesh. URL: <http://grade.dcc.ufba.br/OpenWiMesh>.