Anderson Rosa

Aumento da capacidade de uma rede mesh IEEE 802.11 com uso de múltiplos canais

São José – SC

Agosto / 2014

Anderson Rosa

Aumento da capacidade de uma rede mesh IEEE 802.11 com uso de múltiplos canais

Monografia apresentada à Coordenação do Curso Superior de Tecnologia em Sistemas de Telecomunicações do Instituto Federal de Santa Catarina para a obtenção do diploma de Tecnólogo em Sistemas de Telecomunicações.

Orientador:
Prof. Marcelo Maia Sobral

CURSO SUPERIOR DE TECNOLOGIA EM SISTEMAS DE TELECOMUNICAÇÕES
INSTITUTO FEDERAL DE SANTA CATARINA

São José – SC

Agosto / 2014

Monografia sob o título *"Aumento da capacidade de uma rede mesh IEEE 802.11 com uso de múltiplos canais"*, defendida por Anderson Rosa e aprovada em 27 de agosto de 2014, em São José, Santa Catarina, pela banca examinadora assim constituída:

Prof. Marcelo Maia Sobral, Dr.
Orientador

Prof. Odilson Tadeu Valle, M. Eng.
IFSC

Prof. Jorge Henrique Busatto Casagrande, M. Eng.
IFSC

As coisas, por si sós, não são interessantes,
mas tornam-se interessantes apenas se nos interessamos por elas.
S. Ceccato

Agradecimentos

Gostaria de agradecer a minha mãe Roseli pelo apoio durante a realização deste curso e também por seus conselhos. Aos meus amigos que tornaram essa jornada mais divertida. Ao professor orientador Marcelo Maia Sobral por sua excelente didática e dedicação durante todo o curso. Agradeço a todas as outras pessoas que não foram citadas aqui. Muito obrigado a todos.

Resumo

A rede sem-fio IEEE 802.11 pode trabalhar no modo mesh, em que estações se organizam em uma topologia com múltiplos saltos, formando um domínio de broadcast. Nessa rede, uma espécie de roteamento em nível de enlace possibilita que quadros de uma estação de origem cheguem até uma estação de destino, com encaminhamentos feitos por estações intermediárias. Porém as estações mesh de um mesmo domínio broadcast utilizam o mesmo canal de transmissão. A utilização do mesmo canal por diversas estações acaba gerando uma limitação no aproveitamento do meio sem-fio, fazendo com que a vazão média obtida pelas estações seja menor. Este trabalho proprõe uma arquitetura para o uso integrado de múltiplos canais por estações mesh. Essa integração se faz com o uso de uma interface de rede sem-fio para cada canal, e um controlador Openflow para o encaminhamento transparente de quadros entre as interfaces. Um protótipo baseado no sistema operacional Linux foi desenvolvido para demonstrar a viabilidade da arquitetura.

Abstract

The wireless network IEEE 802.11 can work in mesh mode, where stations are organized in a multihop topology, forming a broadcast domain. In this network, a kind of link level routing tables enables an originating station to come to a destination station, with referrals made by intermediate stations. However the mesh stations broadcast the same domain use the same transmission channel. The use of the same channel by several end stations generating a limitation in the use of wireless means, causing the average flow rate obtained by the stations is smaller. This work proprõe an architecture for the integrated use of multiple channels mesh stations. This integration makes use of an interface wireless network for each channel, and an OpenFlow controller for transparent routing of frames between interfaces. A prototype based on the Linux operating system was developed to demonstrate the feasibility of the architecture.

Sumário

Lista de Figuras

Lista de Tabelas

1 Introdução

Com a rápida adoção das redes infraestruturadas WiFi (padrão 802.11), veio a necessidade de oferecer acesso sem fio em locais onde a ligação de vários APs a um switch não era possível. Muitas vezes a instalação de cabos Ethernet no centro de grandes ambientes como indústrias, é considerado fisicamente e financeiramente inviável. Uma rede infraestruturada sem fio é composta de APs (Access point=Ponto de acesso) e clientes, os quais necessariamente devem utilizar esses APs para trafegarem em uma rede.

Além das redes sem fio infraestruturadas existem as redes sem fio não infraestruturadas baseadas em soluções que estendem os padrões existentes tais como redes mesh. Portanto a rede mesh é um modo de operação diferente do que é apresentado pelas redes infraestruturadas do padrão 802.11. Uma rede Mesh é composta de vários nós/roteadores Mesh, que passam a se comportar como uma única e grande rede como o exemplo da figura 1.1, possibilitando que o cliente se conecte em qualquer um destes nós. Os nós têm a função de repetidores e cada nó está conectado a um ou mais dos outros nós. Desta maneira é possível transmitir mensagens de um nó a outro por diferentes caminhos (SAADE et al., 2008).

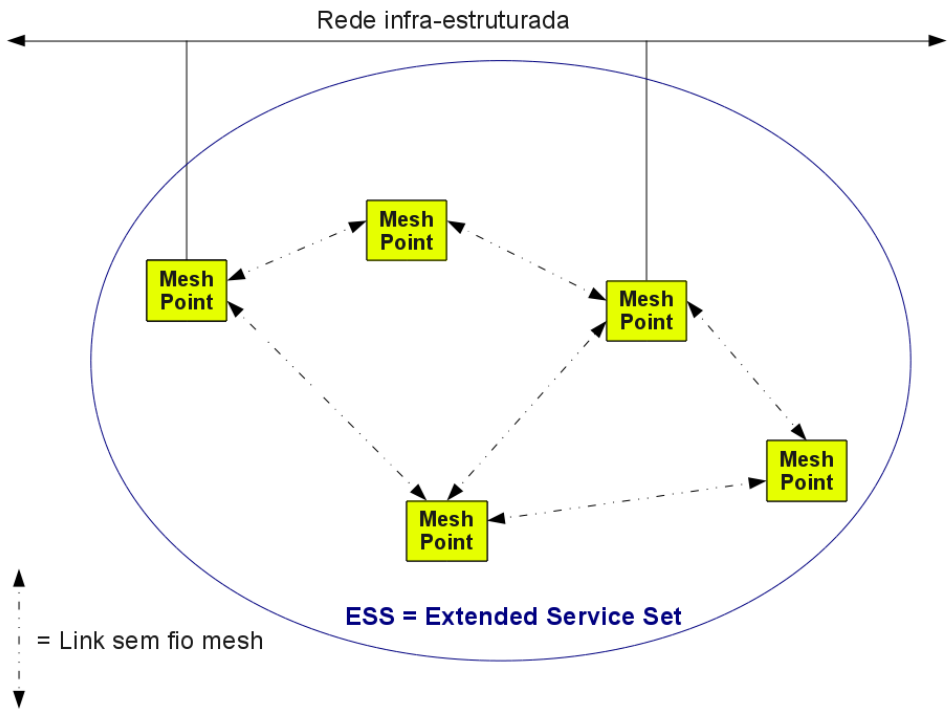

Figura 1.1: Representação de uma rede mesh conectada à rede infra-estruturada.

Porém com o aumento das redes Mesh, também aumentam a quantidade de nós/roteadores, o que provoca a perda de desempenho por número de saltos/retransmissões. Não existe na prática uma limitação para o número de saltos que um pacote pode dar numa rede Mesh, mas existe a degradação da vazão que vai aumentando conforme aumenta o número de saltos. Em equipamentos com apenas uma antena de rádio 802.11g de um fabricante dos EUA, a vazão que pode chegar a 7 Mbps no primeiro salto, não passa de 1 Mbps à partir do quinto salto (WIKIPEDIA, 2014).

Uma proposta para aumentar a vazão em múltiplos saltos seria a utilização de duas ou mais interfaces de rádio em cada roteador Mesh. Essa solução necessitaria de um mecanismo de controle de encaminhamento para fazer balanceamento de quadros entre as interfaces de rádio para obter um ganho na vazão obtida na rede. Porém não existe ainda um padrão finalizado que regulamenta a utilização de múltiplos canais em roteadores Mesh.

1.1 Objetivo Geral

Propor uma arquitetura de forma integrada para redes mesh que utilize múltiplos canais de comunicação.

1.2 Justificativa

As redes Mesh se apresentam como uma opção às rede infraestruturada, quando há a necessidade de oferecer acesso sem fio em locais onde a ligação de vários APs a um switch através de cabos não é possível. Porém essas redes apresentam limitações de vazão quando são organizadas de tal maneira que é necessário o encaminhamento de pacotes por estacões intermediárias, ocorrendo uma significativa diminuição na vazão e o aumento na perda de quadros transmitidos.

2 *Fundamentação Teórica*

2.1 Redes sem fio Mesh

A tecnologia Mesh, teve origem no Defense Advanded Research Projects Agency (DARPA) que é um centro de tecnologia militar dos Estados Unidos. O objetivo da criação da tecnologia era desenvolver uma rede que permitisse uma comunicação fim a fim, sem a necessidade de comunicação com um nó central. Esta tecnologia esteve presente na guerra entre os Estados Unidos e Iraque, onde foi utilizada para interligar soldados, helicópteros, tanques e outro veículos militares (TEIXEIRA, 2004).

2.1.1 Arquitetura da rede Mesh

As redes seguem o padrão IEEE 802.11, padrão seguido pelas redes locais sem fio (WLANs - Wireless Local Area Network). Essas redes estão bastante difundidas devido a razões como facilidade de instalação e suporte a mobilidade. Mesmo assim, apesar dos padrões IEEE 802.11a, b e g serem extremamente populares e encontrados na maioria dos notebooks, PDAs e outros equipamentos sem fio, ainda existe vários desafios em diversas áreas de pesquisa relacionadas que demandam novas soluções. (SAADE et al., 2008).

Porém as redes sem fio possuem dificuldade em atender requisitos como a qualidade de serviço. As principais razões para isso são as limitações da radiotransmissão e a alta taxa de erros devido à interferências; taxa de transmissão ainda muito baixa se comparada com as redes cabeadas; dificuldade de prover segurança, pois os canais sem fio são mais suscetíveis a interceptores. Mas apesar das desvantagens, as vantagens proporcionadas pelas redes sem fio ainda são grandes e a utilização das redes sem fio continua crescendo.

Um dos tópicos de interesse é a comunicação através de múltiplos saltos em redes sem fio baseada em soluções que estendam os padrões existentes. Esta área é bastante promissora, pois redes sem fio infraestruturadas, apesar de oferecerem uma série de vantagens, só podem ser expandidas se os nós forem capazes de encaminhar tráfego originado em outros nós de maneira ad-hoc e autoconfigurável. Comunicação através de múltiplos saltos pode, por exemplo, estender o alcance dos pontos de acesso sem fio sem a necessidade de infraestrutura adicional. (SAADE et al., 2008)

O padrão IEEE 802.11s foi desenvolvido para comunicação através de múltiplos saltos via camada de enlace. Este padrão faz com que a rede sem fio de múltiplos saltos pareça uma única rede local para a camada de rede. O padrão especifica funções de encaminhamento através de múltiplos saltos na camada MAC, utilizando um mecanismo de seleção de caminhos obrigatório chamado HWMP (*Hybrid Wireless Mesh Protocol*). O padrão IEEE 802.11s amplia a definição do sistema de distribuição sem fio original do padrão IEEE 802.11, permitindo que a rede sem fio tenha uma área de cobertura que pode crescer à medida que novos nós se integram a rede adicionando um novo salto.

A rede Mesh é composta por várias WSTAs (Wireless Stations) autônomas que estabelecem enlaces sem fio com as WSTAs vizinhas para a troca de mensagens. Essas estações juntas formam um BSS (Basic Service Set) da rede Mesh chamado de MBSS (Mesh Basic Service Set). As mensagens podem atravessar múltiplos saltos através de WSTAs intermediárias até chegar a WSTA destino. Portanto, uma mensagem pode ser encaminhada para uma WSTA que não está ligada diretamente na WSTA de origem. Todas as WSTAs em um MBSS se comunicam diretamente em nível de enlace trabalhando de maneira cooperativa. A capacidade de múltiplos saltos tem o efeito de aumentar o alcance das WSTAs, e assim a conectividade da rede local sem fio.

Cada estação Mesh funciona como um roteador recebendo e transmitindo mensagens através de seus vizinhos, formando um MBSS que forma um domínio de broadcast. De acordo com o proposto pelo padrão IEEE 802.11s, os nós em uma rede em malha 2.1 podem sem classificados da seguinte forma (SAADE et al., 2008):

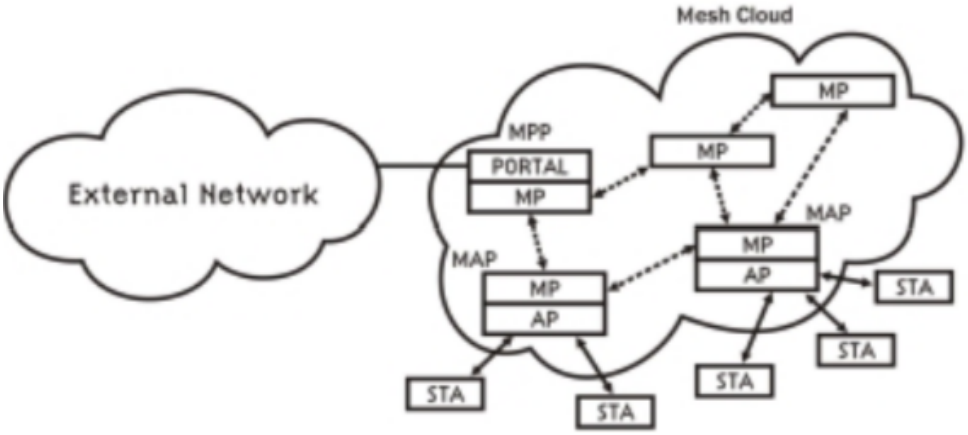

Figura 2.1: Disposição dos componentes de uma rede Mesh IEEE 802.11s (SAADE et al., 2008).

- *Cliente ou estação (STA)* é uma estação que utiliza serviços da rede, mas não participa do encaminhamento de pacotes de outros nós e também dos mecanismos de descoberta de caminhos;

- *Mesh Point ou nó Mesh (MP)* é uma estação que repassa dados e participa das descobertas de rotas, participando da formação e operação da rede Mesh;

- *Mesh Portal Point ou portal Mesh (MPP)* é um MP com a funcionalidade de atuar como um *gateway*, conectando a rede Mesh e uma rede cabeada (internet, por exemplo);
- *Ponto de acesso Mesh ou Mesh Access Point (MAP)* é um MP que incorpora funcionalidades de um ponto de acesso (AP) oferecendo serviços às estações (STA);
- *MBSS (Mesh Basic Service Set)* é formado por um conjunto de estações Mesh. Podendo conter MP, MAP e MPP. Um MBSS pode conter mais de um MPP.

Estas estações Mesh se conectam entre si utilizando o protocolo de roteamento HWMP para descobrir caminhos de comunicação dinamicamente através de seus nós vizinhos. Para a formação dos caminhos cada estação deve utilizar o mesmo canal de transmissão em sua interface. Portanto mesmo utilizando estações com multiplas interfaces, cada interface da estação fará parte de um caminho diferente, em outras palavras, interfaces em canais diferentes pertencem a MBSS distintos. Utilizando os conceitos de SDN/Openflow, apresentados na seção 2.2, iremos explorar a possibilidades do balanceamento de carga entre interfaces diferentes, fazendo com que os pacotes mudem de caminho de forma transparente entre as interfaces da estação.

2.1.2 Sintaxe do padrão IEEE 802.11s

Com o objetivo de criar uma rede sem fio de múltiplos saltos implementada no nível de enlace (camada MAC), o padrão IEEE 802.11s estende o formato dos quadros IEEE 802.11.

Formato dos quadros IEEE 802.11s

O campo de controle do quadro (*frame control field*) está contido nos dois primeiros bytes de um quadro IEEE 802.11. Os dois bits (o terceiro e o quarto) do campo de controle do quadro identificam o tipo de quadro. A Figura 2.2 apresenta os tipos de quadros 802.11 (SAADE et al., 2008).

00 = quadro de gerenciamento	01 = quadro de controle
10 = quadro de dados	11 = reservado

Figura 2.2: Tipos de quadros IEEE 802.11

Além dos quatro tipos, existem ainda os subtipos. Para isso outros quatro bits são reservados para a definição desses subtipos dentro de um tipo (quadro de gerenciamento, controle ou dados). Os quadros mesh introduzidos pelo IEEE 802.11s extendem do padrão IEEE 802.11, portanto precisam ser classificados dentro dos quatro tipos de quadros existentes. Portanto foi definido que (SAADE et al., 2008):

- Os quadros de dados mesh (*Mesh Data Frames*) utilizados nas trocas de informações entre MPs vizinhas, são definidos como quadros de dados (tipo 0x20).
- Os quadros PREQ e PREP, conhecidos como quadros de gerência da rede mesh são definidos como quadros de gerenciamento do tipo 0x0 e subtipo 0xF.

Através da Figura 2.3 é possível observar a estrutura de um quadro IEEE 802.11 estendido pelo cabeçalho mesh. Esse está contido no corpo do quadro IEEE 802.11 que é apresentado na Figura 2.4 e contém quatro campos. *Mesh Flags* é o primeiro campo com 1 byte.

Dos oito bits do campo *Mesh Flags* apenas dois são utilizados. Esses dois primeiros bits determinam a quantidade de endereços MAC presente no campo *Mesh Address Extension*. Portanto pode variar entre 0 e 3, fazendo com que o último campo fique com 0, 6, 12 ou 18 bytes. Cada endereço MAC possui 6 bytes (SAADE et al., 2008).

O segundo campo, Mesh TTL define a quantidade de saltos que um quadro poderá realizar até ser descartado. Quando o quadro atravessa a rede mesh, o TTL é decrementado por cada estação que passa, evitando que quadros fiquem infinitamente na rede criando um *loop* de encaminhamento (SAADE et al., 2008).

O terceiro campo *Mesh Sequence Number* (número de sequência mesh) serve para identificar o quadro, previnindo assim duplicatas na rede evitando retransmissões desnecessárias dentro da rede mesh. E o quarto e último campo *Mesh Address Extension* serve para carregar endereços MAC extras, em alguns casos específicos é necessário a utilização de até seis endereços (SAADE et al., 2008).

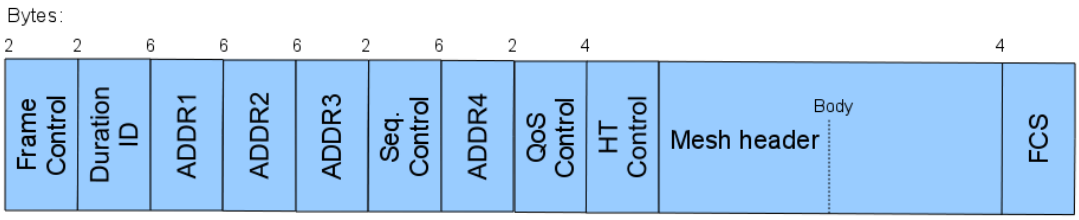

Figura 2.3: Estrutura do quadro IEEE 802.11s

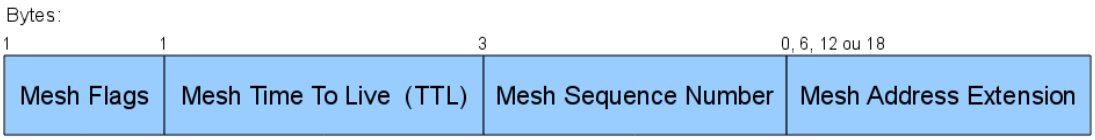

Figura 2.4: Formato do cabeçalho mesh

2.1.3 Endereçamento dos quadros mesh

A rede mesh utiliza o formato de quadro IEEE 802.11s para transmitir seus quadros. O quadro IEEE 802.11s possui quatro endereços que são utilizados na transmissão dos quadros

mesh, 2.3. Os quatro endereços MAC dispostos no quadro IEEE 802.11 são (SAADE et al., 2008):

- ADDR1 representa o endereço MAC do roteador mesh de destino do quadro (next hop);
- ADDR2 representa o endereço MAC do roteador mesh de origem do quadro;
- ADDR3 representa o endereço MAC da estação de destino do quadro (o mesh point final);
- ADDR4 representa o endereço MAC da estação de origem do quadro (mesh point de origem).

2.1.4 Como a rede Mesh é criada

Nas redes sem fio infra-estruturadas um indentificador é utilizado para marcar pontos de acesso que pertencem a mesma rede, chamados de SSID (Service Set IDentifier).Ele serve para diferenciar uma rede sem fio de outra rede sem fio (SAADE et al., 2008).

Assim como a rede sem fio infra-estruturada, a rede mesh também precisa de um identificador de rede para marcar suas estações Mesh e diferenciá-las de outra rede Mesh. Esse identificador é chamado de Mesh ID ou identificador Mesh. Assim como ocorre nas rede sem fio IEEE 802.11, quadros de controles (*beacons*) também são utilizados na rede Mesh para anunciar tal rede. Para que uma estação da rede IEEE 802.11 não se confunda e tente entrar na rede Mesh, estações Mesh (MPs) enviam em broadcast os *beacons* com o SSID setado com um valor especial (SAADE et al., 2008).

Existem dois outros elementos que caracterizam uma rede mesh além do Mesh ID. São eles o protocolo de seleção de caminho citado na seção 2.1.5 e a métrica de seleção de caminho citado na seção 2.1.6. Portanto estes três elementos juntos formam um perfil Mesh. Cada estação Mesh pode suportar diferentes perfis, mas essas estações Mesh em um determinado momento, podem compartilhar o mesmo perfil (SAADE et al., 2008).

O padrão IEEE 802.11s define o HWMP (protocolo de mecanismo de descoberta de caminho) e a métrica *Airtime Link Metric* como sendo os elementos obrigatórios em uma rede Mesh. Se for necessários utilizar protocolos e métricas de descoberta de caminhos alternativos, o padrão define um *framework* que deve ser utilizado para desenvolvimento desses mecanismos alternativos (SAADE et al., 2008).

A rede Mesh é composta por MPs, citado na seção 2.1.1, e também por suas estações vizinhas que compartilham o mesmo perfil. O mecanismo de descoberta de estações Mesh vizinhas é parecido ao que é proposto pelo padrão IEEE 802.11, que faz o escaneamento ativo e passivo (SAADE et al., 2008).

O escaneamento é uma função das redes sem fio infra-estruturadas. É através dela que uma placa de rede sem fio consegue descobrir se um determinado lugar está coberto por alguma rede

Wi-Fi. No escaneamento ativo é utilizado o quadro *probe* e quando acontece a troca entre ponto de acesso são itilizados os quadros *beacon*. Esses quadros são estendidos para incluir campos realacionados a rede Mesh. Portanto a rede Mesh não introduz novos quadros, apenas estende os que já existem no padrão IEEE 802.11 (SAADE et al., 2008).

Além disso é importante comentar sobre o estabelecimento dos *peer links*. Uma estação Mesh cria e mantem *peer links* com seus vizinhos que compartilham o mesmo perfil. Toda vez que uma estação Mesh vizinha é encontrada, através do escaneamento ativo ou passivo, a estação MP usa o protocolo *Mesh Peer Link Managent* para definir um *peer link* (SAADE et al., 2008).

No estabelecimento de um *peer link* entre duas estações mesh, ambos os MPs trocam quadros *Peer Link Open* e *Peer Link Confirm*, como retratado na Figura 2.5. Entretanto, se um MP quer fechar um *peer link*, ele deve mandar um quadro *Peer Link Close* para o MP par (SAADE et al., 2008).

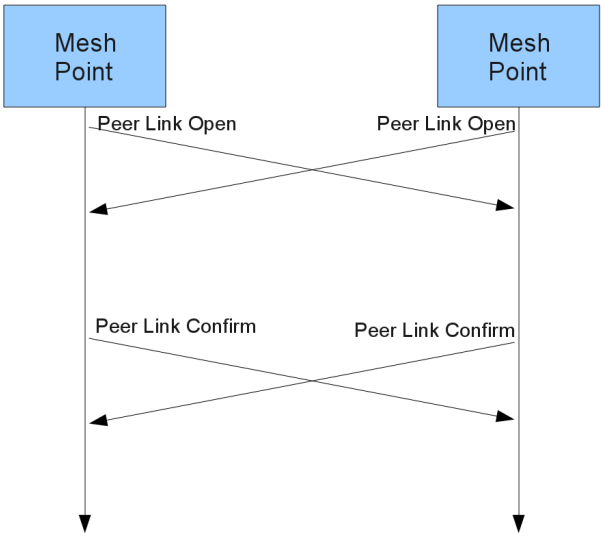

Figura 2.5: Criação de *peer link* entre estações mesh (padrão IEEE 802.11s).

A escolha é feita através de parâmetros trocados entre os MPs através do elemento *Mesh Configuration Element* que são transportados pelos quadros de *beacon*, quadros *Peer Link Open* e *Peer Link Confirm*. Esse elemento de configuração mesh contém diversos campos, porém os mais importantes são o que identifica o protocolo de seleção de caminho (*Path Selection Protocol Identifier*) e o identificador da métrica de seleção de caminho (*Path Selection Metric Identifier*).

2.1.5 Protocolo HWMP

O padrão IEEE 802.11s propõe um protocolo de descobertas de caminhos obrigatório chamado de HWMP (Hybrid Wireless Mesh Protocol). O HWMP é inspirado no protocolo AODV (Ad Hoc On Demand Distance Vector) (PERKINS; ROYER, 2003) e pode ser configurado para trabalhar em dois modos: i) modo reativo sob demanda (on-demand reactive mode)

ou ii) modo pró-ativo baseado em árvore (tree-based proactive mode) (SAADE et al., 2008).

- **Modo sob demanda** por default está sempre disponível e é apropriado para estabelecimento de caminhos entre nós mesh (MPs). A descoberta dos caminhos neste modo é realizada somente quando um pacote precisa ser transmitido pela rede. Portanto, se uma estação precisa transmitir um quadro, inicia-se a descoberta de caminho sob demanda.
- **Modo proativo** cada nó calcula antecipadamente uma topologia em árvore onde a raiz é um determinado nó que se anuncia como tal (MP raiz). Esta abordagem pode ser implementada através de dois mecanismos distintos, pode aumentar a eficiência no encaminhamento de quadros quando existe uma tendência de concentração de tráfego em direção ao nó raiz, que pode por exemplo, estar atuando como portal Mesh (descrito na seção 2.1.1).

Durante o processo de descoberta de um caminho, cada nó participante irá contribuir com seus cálculos de métrica acrescentando ou atualizando dados nos quadros de gerenciamento dedicados à troca de informações de encaminhamento. As funções do HWMP são implementadas independente do modo de operação (pró-ativo ou reativo) pelos seguintes quadros de gerenciamento (SAADE et al., 2008):

- *Path Request* (PREQ) - requisição de caminho - estes quadros são enviados em difusão (*broadcast*) por um MP que deseja encontrar um caminho para outro MP.
- *Path Reply* (PREP) - resposta de caminho - estes quadros são enviados pelo MP de destino, em resposta ao recebimento de uma requisição de caminho (PREQ).
- *Path Error* (PERR) - erro no caminho - estes quadros são usados para notificação de que um caminho não está mais disponível.
- *Root Announcement* (RANN) - anúncio de nó raiz - estes quadros são utilizados pelo nó que se anuncia como nó raiz (MP raiz).

Todos os quadros mencionados são utilizados pelos mecanismos providos pelo HWMP. Para a implementação de múltiplas interfaces nos roteadores mesh, vamos dar preferência ao mecanismo sob demanda por se tratar de um mecanismo mais simples de manusear. Na figura 2.6 é apresentado um exemplo da descoberta de caminhos utilizando o mecanismo sob demanda. Neste exemplo um nó origem (S-MP) deseja encontrar o caminho para o nó destino (D-MP). Para alcançar o nó destino, o nó origem utiliza os nós intermediários (I-MPs) para o encaminhamento dos quadros de gerenciamento.

Para alcançar o nó D-MP, o S-MP envia um quadro PREQ em broadcast na rede com o MAC de D-MP. O nó intermediário I-MP que receber esta requisição, deve checar suas tabelas

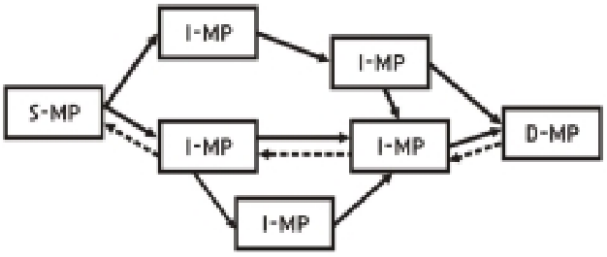

Figura 2.6: Exemplo de descoberta de caminho sob demanda (SAADE et al., 2008).

e verificar se conhece um caminho para D-MP. Se reconhecer um caminho, o I-MP deve encaminhar um quadro de resposta de volta para S-MP. Caso contrário, o nó I-MP deve retransmitir o quadro de broadcast contendo o MAC de D-MP.

Os quadros PREQ e PREP possuem um campo chamado métrica que é atualizado toda vez que uma estação intermediária encaminha o quadro de gerenciamento. O campo métrica é utilizado para determinar o melhor caminho entre os nós origem e destino. Utilizando a métrica, o nó destino D-MP é capaz de selecionar um caminho reverso com a melhor métrica dentre os vários caminhos. Com a seleção do caminho reverso pelo D-MP o ciclo de descoberta de caminhos é encerrado.

Em uma rede mesh densa, uma grande cobertura em um meio sem fio não alcança altas taxas de transmissão, pois um afeta diretamente o outro. Na rede mesh, para atingir o maior número de estações possível, quadros broadcast são transmitidos a baixas taxas, pois estações distantes terão maior chance de receber o quadro que foram transmitidos. Em contra partida, a baixa taxa de transmissão pode fazer com que os quadros viagem de forma mais lenta na rede mesh, principalmente em redes densas.

Com o uso de várias interfaces nos roteadores Mesh podemos gerar mais caminhos na rede Mesh utilizando multiplos canais. Desta forma pretende-se melhorar as opções de caminhos, levando-se em conta a métrica.

2.1.6 Métrica ALM

Atualmente o IEEE 802.11s especifica ALM (Airtime Link Metric) como sendo a métrica para definir a qualidade de enlace sem fio. A métrica ALM representa o tempo de transmissão de um quadro considerando a taxa de transmissão, overhead imposto pela camada física e a taxa de erros do enlace. Apesar de o padrão IEEE 802.11s assegurar compatibilidade entre dispositivos de diferentes fabricantes ditando um mecanismo mandatório (HWMP e *Air Link Metric*), ele também inclui um framework estensível que pode ser usado para dar suporte a aplicações específicas (SAADE et al., 2008). A métrica ALM, de acordo com o padrão, é apresentada na equação 2.1.

$$C_a = [O + \frac{B_t}{r}]\frac{1}{1 - e_f} \tag{2.1}$$

A sigla O é a latência constante de *overhead* da camada física, Bt é o tamanho do quadro de teste (1024 bytes), r é a taxa de transmissão em Mbps com que o MP transmite o quadro de teste e ef é a probabilidade de erro medida no envio do quadro de teste. Durante o processo de descoberta de um caminho (*path discovery*), cada nó participante irá contribuir com seus cálculos de métrica acrescentando ou atualizando dados nos quadros de gerenciamento dedicados à troca de informações de encaminhamento.

Como cada interface do roteador Mesh pertence a uma rede Mesh própria, o caminho de cada interface irá também ter uma métrica diferente. Desta forma, pode-se fazer um balanceamento de carga entre canais distintos de transmissão.

2.1.7 Suporte a rede Mesh no Linux

O open 802.11s é uma implementação *open-source* do padrão IEEE 802.11s para o sistema operacional linux. Este projeto foi incorporado ao *framework* MAC802.11. Os desenvolvedores utilizam o MAC802.11 para escrever *device drivers* para interfaces de rede sem fio do tipo softMAC. Dispositivos softMAC permitem um controle mais preciso do hardware, permitindo a gestão de quadros 802.11 que são criados por software. Em dispositivos softMAC, a análise e geração de quadros 802.11 são feitos por software e não por hardware que tradicionalmente são feitos sob medida para dispositivos dedicados, como switches e roteadores.

O softMAC é um device driver WiFi do tipo mais comum encontrado em Linux. Ele utiliza o framework "MAC80211" para efetuar as funções de associação, autenticação e scanning da rede WiFi. A rede sem fio é implementa no Kernel do Linux através de um framework chamado "mac80211" que é responsável por toda comunicação Wi-Fi no Linux. A figura 2.7 mostra como o "MAC80211" está estruturado dentro do Linux.

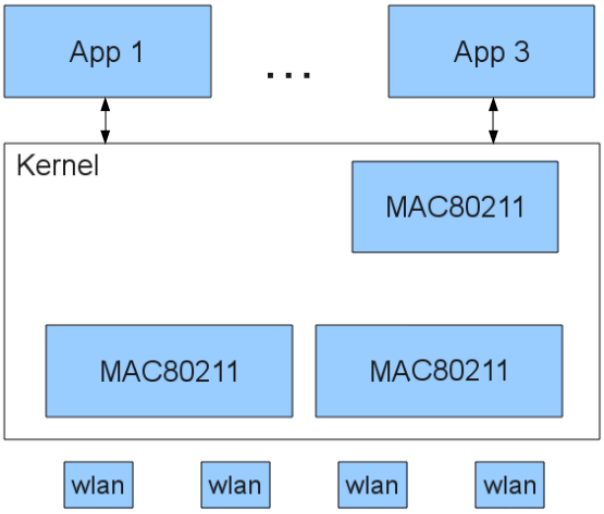

Figura 2.7: Framework MAC80211 no kernel linux.

Através do open 802.11s é possível implementar uma rede mesh através de computadores com o sistema Linux. Assim podem-se utilizar interfaces usb conectadas nesses computadores para implantar redes Mesh sem a necessidade da aquisição de equipamentos, pois o IFSC possui vários computadores disponíveis para projetos. Com várias interfaces conectadas nos computadores pode-se implantar uma rede Mesh para cada canal configurado nas interfaces sem fio.

2.2 SDN

Os equipamentos de rede atuais (roteadores, switches, *access* points, etc) funcionam de maneira autônoma. Esses dispositivos convencionais dependem de protocolos padronizados que definem mecanismos de encaminhamento de pacotes. Portanto estes equipamentos não possibilitam a programação ou alteração do modo de encaminhamento, seja para incorporar protocolos e mecanismos de comutação, ou modificá-los. Isso inibi a experimentação de novos protocolos e técnicas de encaminhamento de pacotes. Essa limitação levou ao desenvolvimente de um novo conceito em redes de computadores chamado de SDN (Software Defined Network).

2.2.1 Conceituação SDN

No conceito SDN, a inteligência da rede é centralizada nos controladores SDN baseados em software que mantêm uma visão geral da rede. O conceito SDN separa conceitualmente a estrutura dos equipamentos de rede em três camadas, a camada de aplicação, a camada de controle e a camada de encaminhamento de dados. A camada de controle possui uma interface de programação que permite aos administradores de rede programar o comportamento da rede. Após a camada de controle ser programada com as regras desejadas para o tratamento de fluxo de dados, a camada de encaminhamento irá comutar os pacotes de acordo com as regras programadas na camada de controle. Através do conceito SDN, a camada de controle, que é responsável pelo encaminhamento de pacotes, pode ser separada do hardware do equipamento de rede.

O conceito dá aos gerentes de rede a flexibilidade para configurar, gerenciar, proteger e otimizar os recursos de rede. São três camadas que compõem o conceito SDN como vemos na figura 2.12. Na camada mais inferior se encontram a camada de hardware onde estão os dispositivos de rede. Estes possuem um módulo SDN para receber instruções de funcionamento da camada intermediária. A camada intermediária ou camada de controle contém o controle da arquitetura. A camada de controle, neste modelo, é abstraída do hardware podendo até mesmo estar em outro equipamento. A camada de controle se comunica com a camada de aplicação através de APIs de comunicação. Finalmente, a camada de aplicação é uma interface de gerenciamento do controlador que permite acesso aos parâmetros de configuração da camada de controle (FOUNDATION, 2012).

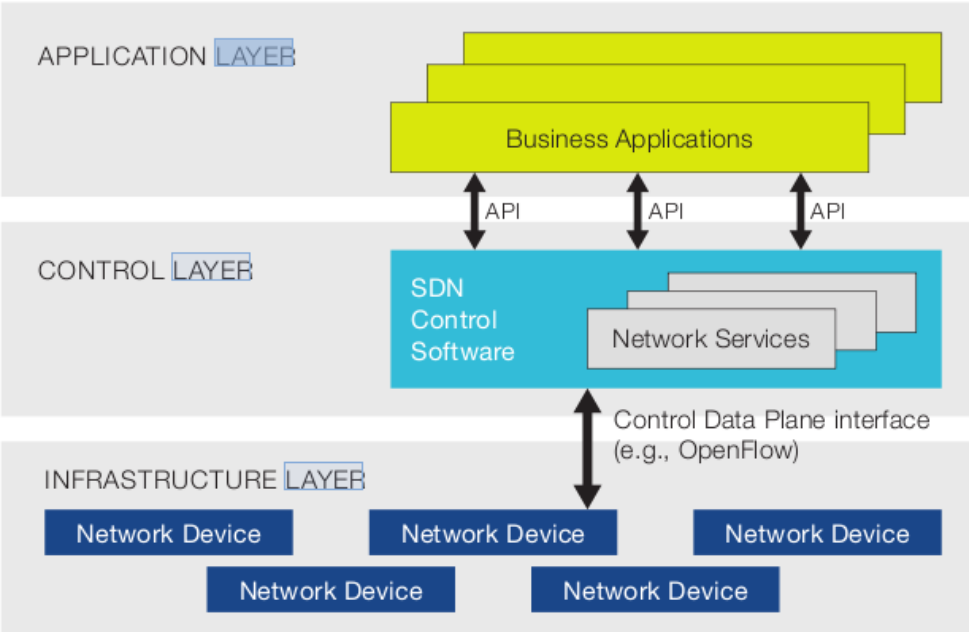

Figura 2.8: Camadas da arquitetura SDN (FOUNDATION, 2012).

2.2.2 Openflow

O *Openflow* é um protocolo para implementação do conceito SDN em redes de computadores. Ele foi desenvolvido pela universidade de Stanford em 2008. Seu principal objetivo era desenvolver experimentos de protocolos de redes utilizando o conceito de SDN (SCOTTI, 2013).

O protocolo define o padrão da interface entre o controlador e a camada de encaminhamento do SDN. Um switch *Openflow* consiste de uma tabela de fluxo (datapath) que contém as regras de encaminhamento de pacotes e um canal *Openflow* (control path) para a comunicação com um controlador externo. A tabela de fluxo consiste de uma ação associada a cada entrada de pacote. O controlador administra o switch através do protocolo *Openflow*. Usando este protocolo, o controlador pode adicionar, atualizar e deletar entradas de fluxo (SCOTTI, 2013).

Através da tabela de encaminhamento de fluxos do datapath é definida uma ação para cada fluxo de pacotes entrante. As regras são carregadas no datapath através do controlador que utiliza o canal *Openflow* para se comunicar com o datapath (SCOTTI, 2013).

O switch *Openflow* é dividido no mínimo em três partes:

- Uma tabela de fluxo que associa uma ação para cada entrada de fluxo;
- Um canal seguro que conecta o switch ao controlador remoto;
- O protocolo *Openflow* que provê um modo aberto e padrão para a comunicação entre o controlador e o switch.

O *Openflow* fornece um protocolo aberto para programar a tabela de fluxo em diferentes switches. Os pesquisadores podem controlar seus próprios fluxos de dados, escolhendo as rotas

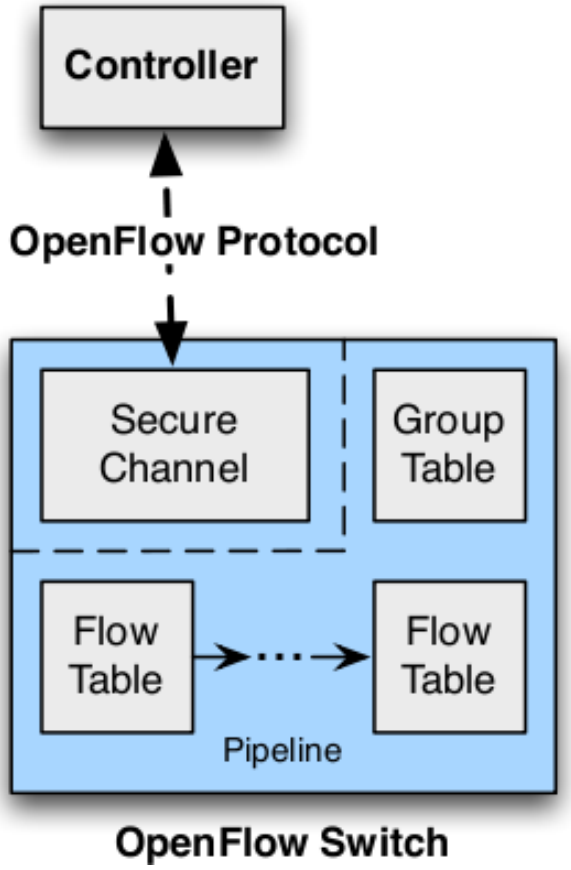

Figura 2.9: Arquitetura do switch Openflow (MCKEOWN et al., 2008).

que seus pacotes devem seguir, alterando o processamento de encaminhamento de pacotes do roteador. Desta forma, os pesquisadores e gerentes de rede podem experimentar novos protocolos de encaminhamento de pacotes (SCOTTI, 2013).

2.2.3 Tabela Openflow

O roteador openflow, pode possuir uma ou mais tabelas de fluxos. Através desta, o controlador pode adicionar, atualizar e deletar entradas de fluxo (SCOTTI, 2013).

Cada entrada na tabela de fluxos contém:

Match fields	Contadores	Instruções

Figura 2.10: Principais componente da entrada de fluxo na tabela de fluxo (MCKEOWN et al., 2008).

- **Match Fields:** para comparar os pacotes. A comparação é feita com a porta de entrada e o cabeçalho do pacote, também podem ser comparados metadados especificado por uma tabela anterior;
- **Couters:** para atualizar os pacotes comparados;
- **Instructions:** para modificar as ações ou o processamento pipeline.

As tabelas de fluxos de um roteador Openflow são numeradas sequencialmente, começando na tabela 0. O fluxo de entrada é sempre comparado primeiro com a tabela 0. Outras tabelas podem ser usadas na comparação, dependendo do resultado da comparação com a tabela 0. Se por acaso o fluxo de entrada não for classificado em nenhuma tabela, será enviado para o controlador para que ele possa inserir uma nova regra de encaminhamento na tabela (SCOTTI, 2013).

O *pipeline* Openflow de cada roteador Openflow podem conter várias tabelas de fluxo, cada tabela de fluxo contendo várias entradas de fluxo. O processamento pipeline Openflow define como os pacotes interagem com a tabela de fluxo. Um roteador openflow com apenas uma tabela de fluxo pode ser configurado, nesse caso o processamento do pipeline será mais reduzido (SCOTTI, 2013).

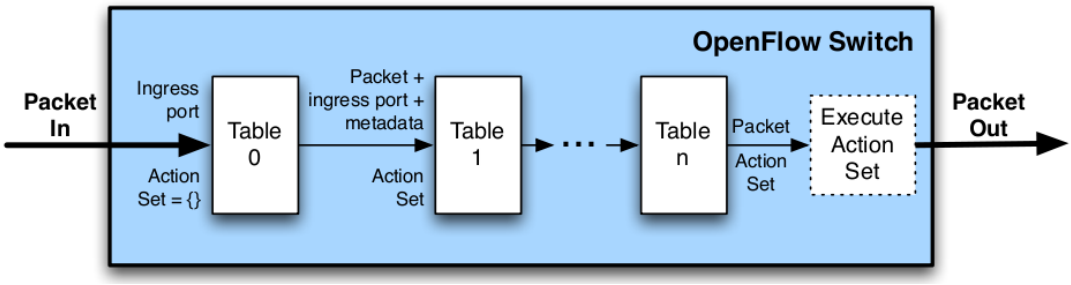

Figura 2.11: Entradas de fluxo sendo processadas por múltiplas tabelas (MCKEOWN et al., 2008).

As instruções de entrada de fluxo podem encaminhar explicitamente o pacote para uma outra tabela de fluxo, usando a instrução Goto, onde o mesmo processo é repetido novamente. Uma entrada de fluxo só pode direcionar um pacote para um número de tabela de fluxo que seja maior do que seu próprio número de tabela atual. Portanto, as entradas de fluxo do último quadro do pipeline não pode incluir uma instrução Goto. Quando o processamento pipeline pára, ou seja, se a entrada de fluxo correspondente ao pacote não inclui uma instrução Goto, o pacote é processado com o seu conjunto de ações associadas e usualmente é encaminhado (SCOTTI, 2013).

Cada tabela possui em suas regras os valores que serão comparados com o campo match field da entrada de fluxo. Além dos cabeçalhos dos pacotes, metadados podem ser usados para passar informações entre tabelas do roteador openflow. Então quando um pacote é processado por uma tabela, esta pode inserir metadados para serem comparados com outras tabelas (SCOTTI, 2013).

Quando o roteador recebe um pacote, o roteador openflow executa as funções mostradas no fluxograma 2.13. O roteador começa a processar o pacote na tabela 0. Se o pacote combinar com alguma entrada da tabela, as instruções da entrada de fluxo podem ser: atualizar o conjunto de ações; atualizar os campos de comparação do pacote; atualizar os metadados. Nesse ponto o contador da entrada correspondente é atualizado. Após executar as instruções, se não tiver nenhum encaminhamento do pacote para outra tabela, serão executadas as ações determinadas

Campos de comparação (Match Field)
Porta de entrada
Metadados
MAC de Origem
MAC de Destino
Tipo de Ethernet
Identificação de VLAN
Prioridade de VLAN
Etiqueta MPLS
Classe de tráfego MPLS
Origem IPv4
Destino IPv4
IPv4 com ARP
Bits ToS IPv4
Porta de Origem do TCP/UDP/SCTP do ICMP
Porta de Destino do TCP/UDP/SCTP do ICMP

Figura 2.12: Campos do pacote usados para comparar com o fluxo de entrada (MCKEOWN et al., 2008).

pela tabela de fluxo. Se não tiver nenhuma entrada correspondente ao pacote, a entrada estará sujeita as determinações de acordo com a configuração da tabela. Essas determinações podem ser o envio do pacote para o controlador, descartar a entrada de fluxo ou encaminhar a entrada para a próxima tabela (SCOTTI, 2013).

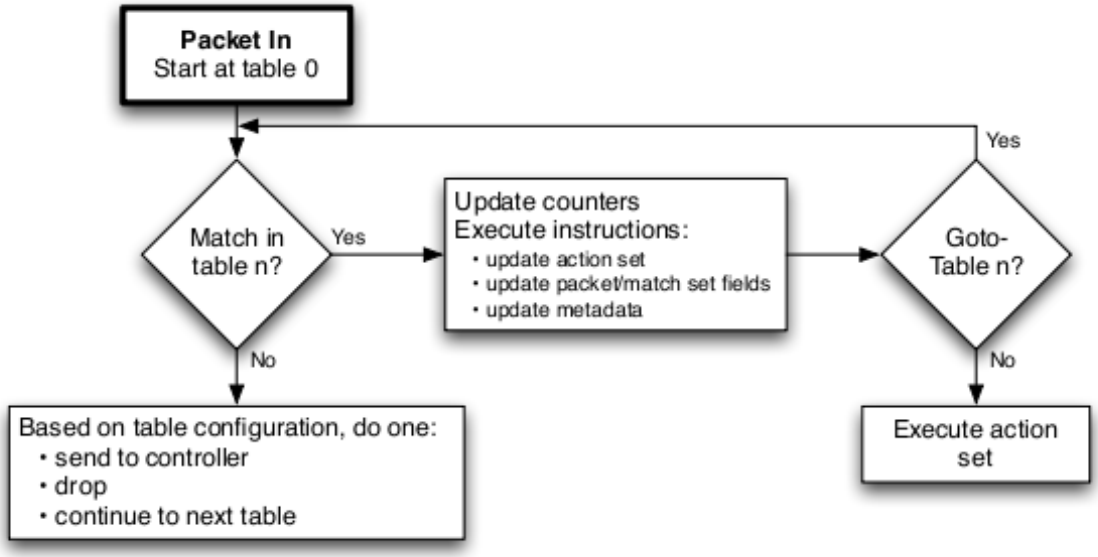

Figura 2.13: Fluxo dos pacotes pelo roteador Openflow (MCKEOWN et al., 2008).

Cada entrada da tabela possui instruções a serem executadas quando o pacote corresponde ao campo match field da entrada. Essas instruções podem alterar parâmetros dos pacotes, nas ações do pacote ou processo pipeline. As instruções suportadas pelo roteador openflow estão listadas abaixo (MCKEOWN et al., 2008):

- **Apply-Actions *actions(s)*:** Aplica as ações imediatamente, sem qualquer alteração no

conjunto de ações. Estas instruções podem ser utilizadas para modificar o pacote entre duas tabelas.

- **Clear-Actions:** Remove todo o conjunto de ações imediatamente.
- **Write-Actions *actions(s)*:** Insere uma ação específica no conjunto de ações existentes. Se a ação inserida já existir, ela será sobrescrita.
- **Write-Metadata *metadata/mask*:** Escreve o valor de metadados no campo metadados.
- **Go-Table next-table-id:** Indica a próxima tabela no processamento pipeline. O id da tabela precisa ser maior que a atual.

O switch openflow pode rejeitar alguma entrada de fluxo caso não suporte a ação associada. Nesse caso o roteador irá retornar uma mensagem de erro, de fluxo não suportado. As ações estão associadas a cada pacote. As ações associadas aos pacotes são definidas ao passar pelas tabelas de fluxos recebendo as instruções Write-actions e Clear-actions. Essas ações serão executadas quando o processo pipeline chegar ao fim, quando a entrada de fluxo não pussuir mais a instrução Go-to (SCOTTI, 2013).

Os switches openflow não são obrigados a suportar todos os tipos de ações, alguns são opcionais. Quando o controlador é ligado a um roteador openflow, este indica qual das ações opcionais que suporta. Algumas ações seguem listadas abaixo (SCOTTI, 2013):

- **Output:** esta ação encaminha o pacote para uma porta de saída. Os roteadores openflow devem suportar encaminhamentos para portas físicas e virtuais.
- **Drop:** quando não existe uma ação específica para descartar o pacote, eles serão descartados quando não houver nenhuma ação para ser executada.
- **Group:** processa o pacote através de um grupo específico.
- **Set-Queue (Opcional):** define o ID de fila do pacote. Quando o pacote é encaminhado para a porta usando a ação output, o ID de fila determina em qual fila de uma determinada porta o pacote será encaminhado.
- **Set-Field (opcional):** Permite modificar os campos dos cabeçalhos do pacote.

2.2.4 Canal Openflow

O canal openflow é a interface que conecta o roteador openflow com o controlador. Utilizando esta interface, o controlador pode configurar e gerenciar trocas de mensagens, receber eventos do roteador e pacotes através do roteador (SCOTTI, 2013).

A interface entre o caminho de dados (datapath) e o canal openflow é uma implementação específica, portanto todas as mensagens do canal openflow devem ser formatados de acordo com o protocolo openflow. O canal openflow pode ser criptografado usando TLS ou ser executado diretamente sobre TCP (SCOTTI, 2013).

O protocolo Openflow suporta três tipos de mensagens, controller-to-switch, asynchronous, e symmetric, cada um com vários sub-tipos. Controller-to-switch é iniciada pelo controlador para administrar ou inspecionar o estado do roteador. A mensagem asynchronous são iniciadas pelo roteador e são utilizadas para informar o controlador sobre os eventos da rede como chegada de um pacote, algum erro ou alterações no estado do roteador. As mensagens symmetric são iniciadas pelo controlador ou roteador e enviadas sem solicitação, são apenas mensagens de controle (SCOTTI, 2013).

2.2.5 Controlador Openflow

O controlador openflow foi desenvolvido em várias linguagens de programação em diferentes projetos de pesquisa. Alguns deles são mais populares que outros, como Beacon, Maestro, NOX, Trema e Ryu (SCOTTI, 2013).

Esses controladores possuem muitas funções que não serão utilizadas neste projeto. Iremos explorar apenas o balanceamento de carga, o que não necessita de um controlador muito robusto. Portanto, decidimos utilizar um controlador mais simples. Foi decidido utilizar o controlador Ryu.

O controlador Ryu é um software de código aberto que implementa um controlador openflow em um computador com o sistema operacional Linux. Através dele, é possível fazer o controle do hardware da estação mesh que nesse caso se trata do driver IEEE 802.11 do computador. Portanto o controlador Ryu consegue tomar a decisão sobre o encaminhamento dos pacotes recebidos pela placa de rede sem fio.

Para fazer o controle dos pacotes recebidos o controlador Ryu conta com o apoio do switch virtual openflow que cria uma ponte entre a placa de rede do computador e o controlador openflow. Os detalhes do switch virtual openflow são apresentados na seção 2.2.6.

No controlador Ryu, é apresentada a configuração de uma regra apresentada na figura 2.14.

```
$ dpctl add-flow tcp:127.0.0.1:6634 in_port=1,actions=output:2
$ dpctl add-flow tcp:127.0.0.1:6634 in_port=2,actions=output:1
```

Figura 2.14: Exemplo de uma regra configurada através do controlador Ryu.

O controlador openflow Ryu foi instalado em cada roteador mesh, ou seja, nos computadores com o sistema operacional linux. A instalação do Ryu é feita através do programa pip que auxilia na instalação de pacotes da liguagem de programação Python. Para instalar o Ryu foi utilizado o comando install do pip. O comando completo pode ser visto na figura 2.15

(PROJECT, 2014).

```
% pip install ryu
```

Figura 2.15: Comando de instalação do controlador Ryu.

2.2.6 Openvswitch

O openvswitch é um software de código aberto que implementa o protocolo openflow. Ele é um switch virtual sobre a licensa de código aberto Apache 2.0 sendo projetado para permitir a automatização de redes através da sua programação e configuração. Assim o openvswitch oferece uma interfaces de gerenciamento do protocolo Openflow.

O openvswitch tem em sua estrutura a função de portas virtuais, onde essas podem ser vinculadas a outras portas virtuais. Portanto através dessa função foi possível vincular as portas dos canais sem fio de cada estação mesh a porta do controlador ryu que está presente em cada estação mesh de forma descentralizada. Através dele então é possível identificar eventos que ocorrem nos múltiplos canais das estações, como por exemplo o recebimento de um pacote da rede sem fio mesh. Na figura 2.16 é possível verificar as portas virtuais que estão vinculadas ao openvswitch.

Portanto é possível observar na figura 2.16 as portas virtuais 1(mesh0) e 2(mesh1). A porta virtual 1 do switch está vinculada a interface sem fio mesh0. Mesh0 é a antena sem fio instalada fisicamente no computador que está atuando como estação mesh. É possível identificar o endereço MAC de cada porta virtual do switch. Além das portas virtuais vinculadas aos canais mesh0 e mesh1, existe a porta virtual local br0 que representa a porta virtual do controlador Ryu. A porta virtual (br0) nada mais é do que uma bridge utilizada para interligar os canais mesh ao controlador Ryu.

```
root@labic107:/home/andersonrosa8/Downloads# ovs-ofctl -O OpenFlow12 sho
OFPT_FEATURES_REPLY (OF1.2) (xid=0x2): dpid:0000001a3fc0271b
n_tables:254, n_buffers:256
capabilities: FLOW_STATS TABLE_STATS PORT_STATS QUEUE_STATS
 1(mesh0): addr:00:1a:3f:c0:27:1b
     config:     0
     state:      0
     speed: 0 Mbps now, 0 Mbps max
 2(mesh1): addr:00:1a:3f:c0:27:1b
     config:     0
     state:      0
     speed: 0 Mbps now, 0 Mbps max
 LOCAL(br0): addr:00:1a:3f:c0:27:1b
     config:     0
     state:      0
     speed: 0 Mbps now, 0 Mbps max
OFPT_GET_CONFIG_REPLY (OF1.2) (xid=0x4): frags=normal miss_send_len=0
root@labic107:/home/andersonrosa8/Downloads#
```

Figura 2.16: Portas da estação mesh que estão vinculadas ao controlador openflow.

3 *Implementação de uma rede mesh com múltiplos canais*

Em uma rede IEEE802.11 todas as estações mesh de um mesmo domínio broadcast devem estar configuradas para utilizarem o mesmo canal de transmissão. Quando uma estação próxima está transmitindo, outras estações que estão no mesmo alcance de cobertura da transmissão devem esperar até o fim da transmissão para então utilizarem o meio sem fio. Isso acontece porque as estações mesh utilizam-se do mesmo canal de transmissão, fazendo com que a rede mesh fique limitada no que diz respeito a taxa de transmissão. No intuito de aumentar a taxa de transmissão da rede mesh, foi implementado uma rede mesh onde cada estação mesh utiliza dois ou mais canais de transmissão. Com isso, é possível uma estação mesh encaminhar uma quadro por um dos canais que apresenta uma melhor estimativa de qualidade de enlace ou estar menos ocupado.

Esse capítulo aborda a proposta de um modelo da rede mesh utilizando múltiplos canais de transmissão por meio de um controlador Openflow. Também é apresentado o experimento onde uma estação mesh troca o seu canal de transmissão através do controlador Openflow.

3.1 Arquitetura

A arquitetura da rede mesh foi desenvolvida de forma descentralizada, ou seja, cada estação mesh tem a responsabilidade de processar seus pacotes recebidos e encaminhá-los. Como apresentado na figura 3.1 uma estação mesh é composta, basicamente, por três componentes, sendo eles o computador com sistema operacional Linux, controlador openflow Ryu e switch virtual openflow Openvswitch.

Quando uma estação mesh recebe um quadro, são verificados os endereços MACs de origem e destino contidos em seus campos e também a porta de entrada no switch virtual Openflow. Esses campos são comparados aos campos das regras (*match fields*) contidas da tabela Openflow. Se os campos corresponderem aos campos do quadro, este será executado conforme a ação associada a regra de comparação da tabela Openflow. Senão ele será enviado para o controlador decidir o que deve ser feito com o quadro.

Figura 3.1: Arquitetura da estação mesh com duas antenas Wi-Fi.

Switch virtual Openvswitch

O switch virtual Openvswitch é capaz de criar portas virtuais e interligá-las entre si. Com isso pode-se interligar os canais sem fio da estação mesh com o controlador openflow.

É através do Openvswitch que quadros da rede mesh serão recebidos e transmitidos em cada estação mesh. Se um quadro for recebido por uma de suas portas virtuais, por exemplo a porta 1, essa pode ser conectada a sua porta virtual porta 2. Portanto o Openvswitch switch no modelo proposto tem a função de criar uma "ponte" entre as portas virtuais de entrada da estação mesh com suas portas de saída virtuais.

Para configurá-lo são utilizados comandos definidos em seu manual de configuração. Para o modelo proposto, foi configurado para possuir três portas virtuais. As portas que foram definidas são porta virtual LOCAL, porta virtual 1 e porta virtual 2. Como apresentado na figura 3.2 a porta LOCAL corresponde a interface de rede br0 do computador. Essa interface possui um endereço IP, e se integra à pilha de protocolos TCP/IP do sistema operacional, e através dela o computador pode se comunicar com a rede mesh. Portanto, a interface br0 funciona como um elo de ligação entre a rede mesh, por meio do Openvswitch, e os processos no computador que se comunicam por essa rede. Já as portas virtuais 1 e 2 correspondem respectivamente ao canal 1 (mesh0) e canal 11 (mesh1) da estação mesh.

As portas virtuais foram criadas com os comandos do Openvswitch. Cada porta virtual foi criada com os comandos listados na tabela 3.1. Esses comandos precisam ser executados uma única vez, quando da implantação do Openvswitch em uma estação mesh.

Tabela 3.1: Descrição do hardware de testes.

Comando Openvswitch	Ação
ovs-vsctl add-br br0	Cria a porta br0 no Openvswitch
ovs-vsctl add-port br0 mesh0	Vincula a porta mesh0 a porta br0
ovs-vsctl add-port br0 mesh1	Vincula a porta mesh1 a porta br0

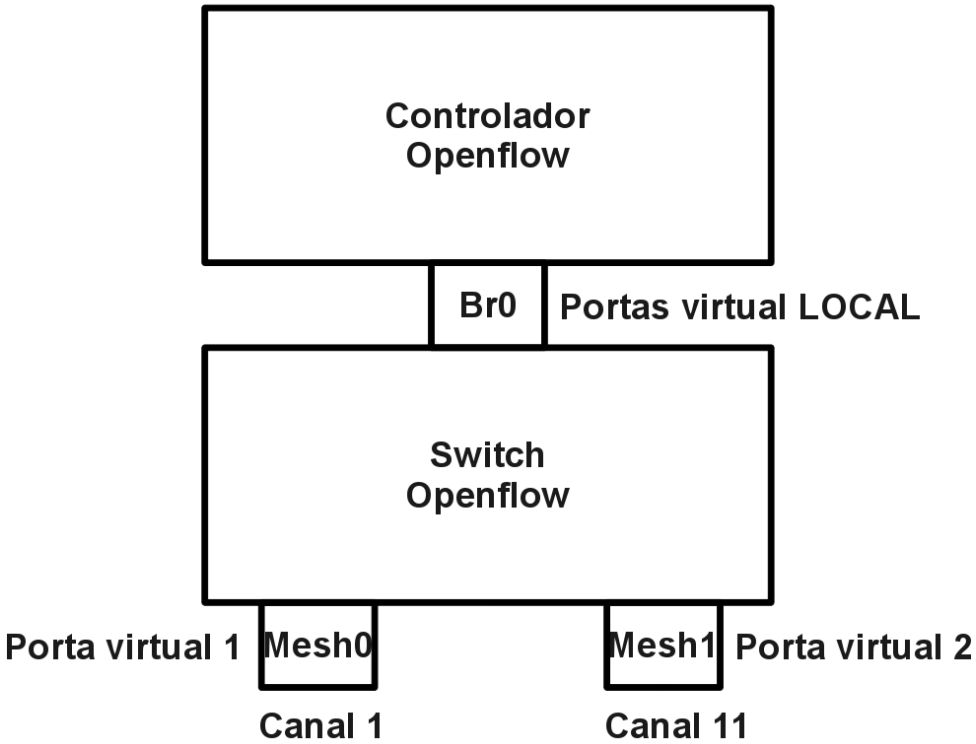

Figura 3.2: Portas virtuais configuradas no Openvswitch.

Controlador openflow Ryu

Os controladores openflow definem como o switch virtual comuta seus quadros através da configuração das regras da tabela openflow. Ele é responsável por escolher o canal de saída de um quadro de acordo com algum critério pré-programado.

O controlador foi programado para receber uma cópia de um quadro através do openvswitch e encaminhá-lo por algum canal do switch virtual. Ele considera três situações ao receber um quadro. 1) Identifica a porta de entrada do quadro, também 2) o endereço MAC de origem do quadro e 3) endereço MAC de destino do quadro.

Através das informações contidas no quadro, o controlador pode tomar algum tipo de decisão no encaminhamento dos quadros através das portas do switch. Portanto se um quadro contém o endereço de destino igual ao endereço MAC da estação que recebeu o quadro, então o controlador deve encaminhar o quadro para as camadas superiores de rede. Por outro lado se o controlador recebe um quadro que não seja da sua estação mesh, deve encaminhar o quadro por um canal de transmissão através das portas virtuais mesh0 e mesh1 do switch openflow. No diagrama 3.3 apresenta as ações do controlador openflow dependendo das características dos pacotes.

Como apresentado no fluxograma 3.3, o fluxo de pacotes pode começar a partir do switch Openvswitch em direção ao controlador Openflow ou então começar da pilha de protocolos da própria estação, ou seja, geração de pacotes dentro do sistema sendo encaminhados de br0 para Mesh0 ou Mesh1.

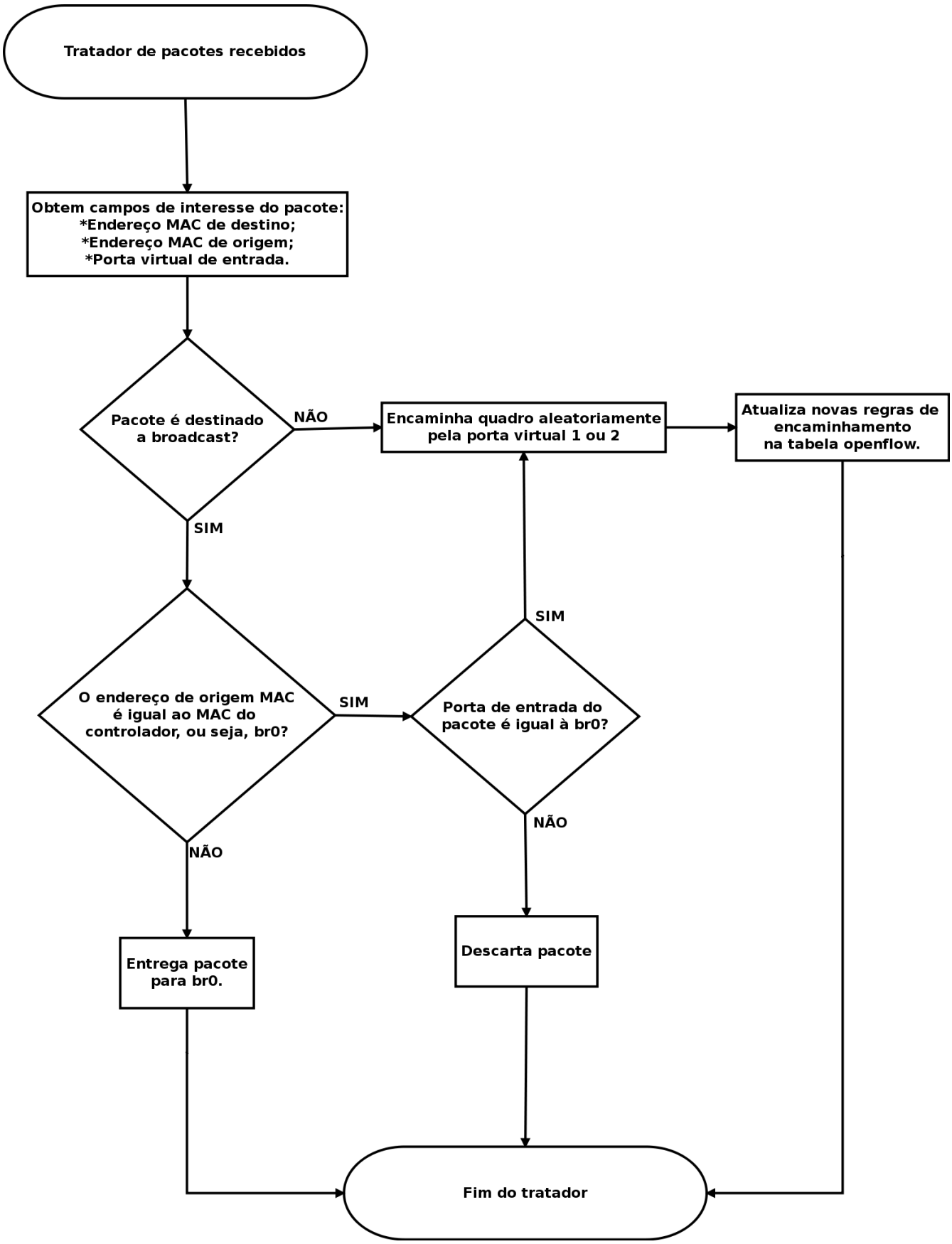

Figura 3.3: Fluxo de encaminhamento do controlador Openflow.

Quando a estação mesh recebe um quadro, o switch Openflow verifica os campos MACs de origem e destino assim como a porta de entrada e compara com todas as regras da tabela openflow que é composta por seletores (math fields) que são compostos pelos MACs de origem e destino e a porta de entrada dos quadros.

Através da figura 3.4 é possível visualizar como acontece o encaminhamento dos quadros entre os canais da estação mesh. 1) A estação mesh recebe um quadro através do canal 1. Através da tabela Openflow é verificado se algum campo do quadro corresponde com alguma regra da tabela openflow. Se algum campo corresponder, o quadro recebe a ação correspondente, por exemplo, será encaminhado pela mesma porta de entrada. 2) Agora se o quadro não conferir com nenhuma regra da tabela Openflow, ele será encaminhado para o controlador Openflow. 3) O controlador openflow é programado para 4) encaminhar o quadro para uma porta diferente da porta de entrada.

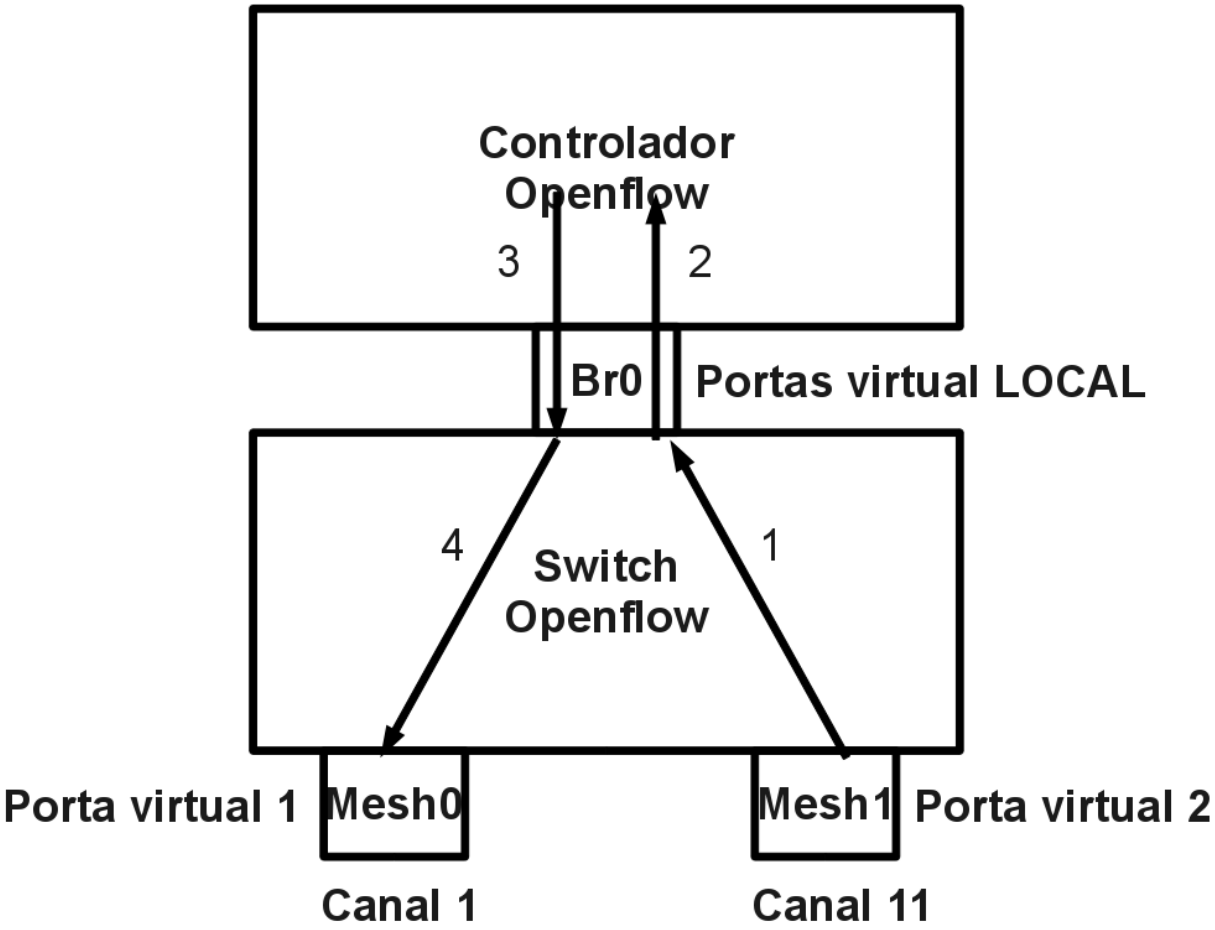

Figura 3.4: Fluxo de encaminhamento de quadros IEEE 802.11 pelo switch virtual.

Computador Linux

Para que fosse possível interceptar os quadros da rede mesh, foi necessário alterar o driver da rede sem fio das estações mesh. Como as estações mesh são implementadas em computadores com o sistema operacional linux, o driver WiFi do kernel linux tem total autonomia para gerenciar quadros IEEE802.11. Portanto o driver acaba atrapalhando o objetivo do modelo pro-

posto que é controlar o encaminhamento dos quadros através da escolha do canal de transmissão. Para que a estação mesh tenha acesso aos quadros gerenciados pelo driver, foi necessário alterar uma linha de código do arquivo rx.c do driver.

Na figura 3.8 é possível observar uma linha do driver, mais especificamente do arquivo rx.c responsável por tratar quadros recebidos da rede sem fio, sendo comentada.

Na figura 3.5 é apresentada a posição do driver MAC80211 dentro da estrutura da estação mesh. É possível observar que o MAC80211 está antes do Openvswitch, portanto o quadro da rede sem fio seria processado direto sem que o Openvswitch pudesse capturá-lo.

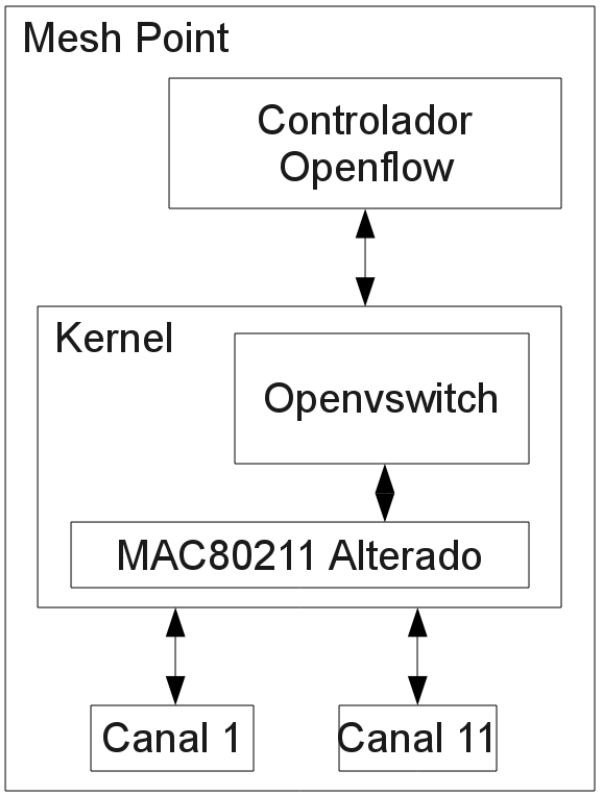

Figura 3.5: Estrutura completa da estação mesh com o MAC80211 alterado.

O controlador recebe uma cópia do quadro recebido pelo switch virtual da estação mesh. Quando ocorre a chegada de um quadro no switch virtual, ele identifica o endereço MAC de origem, o endereço MAC de destino e também a porta de entrada. Através dessas informações o controlador openflow consegue tomar uma decisão e encaminhar o quadro por uma determina porta virtual do switch virtual. A decisão tomada é gravada na tabela openflow do switch.

Quando uma estação sem fio recebe um quadro, o próprio driver do linux faz um teste para saber se o quadro recebido pertence à própria estação. Esse teste é feito comparando os endereços MAC ADDR3 do quadro recebido com o endereço MAC da estação que recebeu o quadro. Se os endereços MAC forem iguais então o driver simplesmente entrega o quadro localmente para as camadas superiores da rede.

A linha comentada na figura 3.8 evita que o quadro mesh seja encaminhado pelo driver possibilitando a sua captura através do switch openflow. Assim, o quadro pode ser comparado

com a tabela openflow através do driver seguindo uma trajetória dependendo das regras da tabela openflow como apresentado na 3.6. Sem essa alteração, a trajetória do quadro ficaria restrita apenas até ao driver da placa sem fio como apresentado na 3.7.

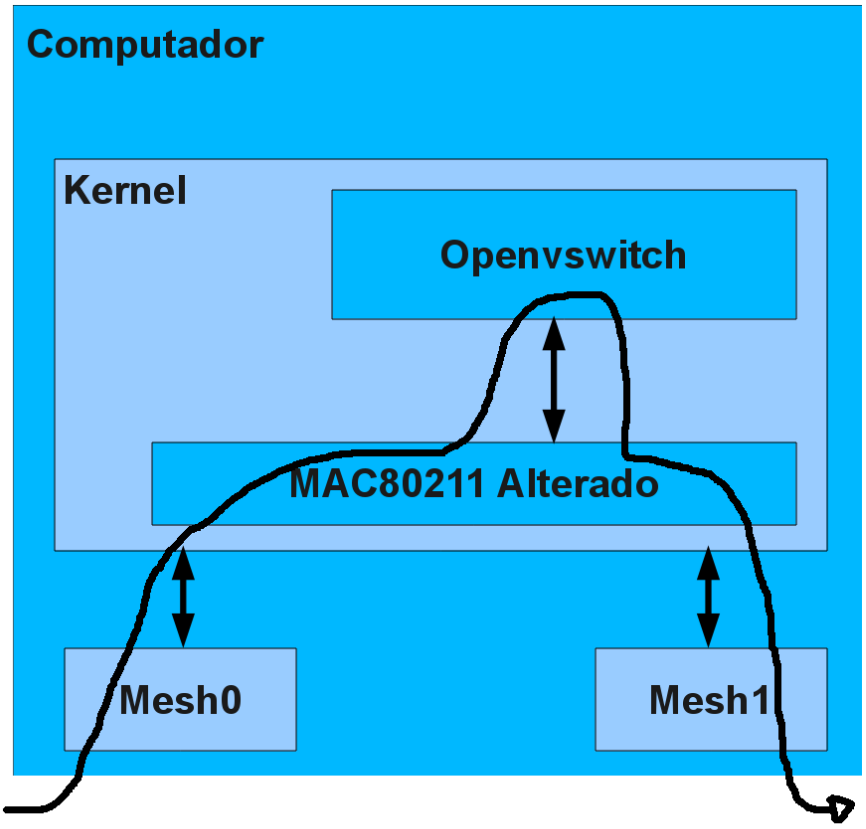

Figura 3.6: Trajeto do quadro em uma estação mesh com driver alterado.

O switch openflow está posicionado entre os canais sem fio e o controlador openflow de cada estação mesh. Portanto o switch openflow recebe os fluxos de pacotes endereçados para os canais das estações mesh. Quando o switch virtual de uma estação mesh recebe um pacote, ele executa as funções mostradas no fluxograma 3.3. Se a tabela openflow conter uma ação de envio imediato para o controlador, o pacote será enviado imediatamente para o controlador.

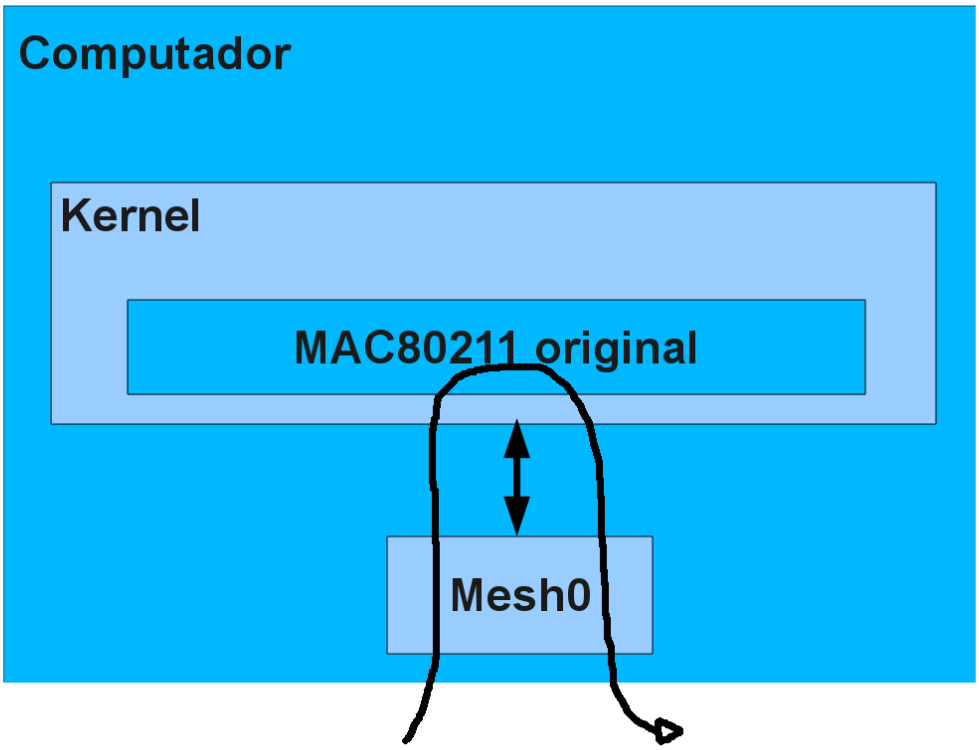

Figura 3.7: Trajeto do quadro em uma estação mesh com driver original.

```
/* Frame has reached destination.  Don't forward */
 //if (!is_multicast_ether_addr(hdr->addr1) && ether_addr_equal(sdata->vif.addr, hdr->addr3))
        return RX_CONTINUE;
```

Figura 3.8: Trecho de código alterado do driver da placa de rede sem fio.

4 Experimento

O exeperimento é baseado em uma rede mesh composta por três estações mesh. Neste ambiente existem duas antenas, um controlador openflow e um switch openflow em cada estação mesh. O exeperimento foi realizado no laboratório LabIC do campus do IFSC. A disposição das estações mesh está apresentado na figura 4.1.

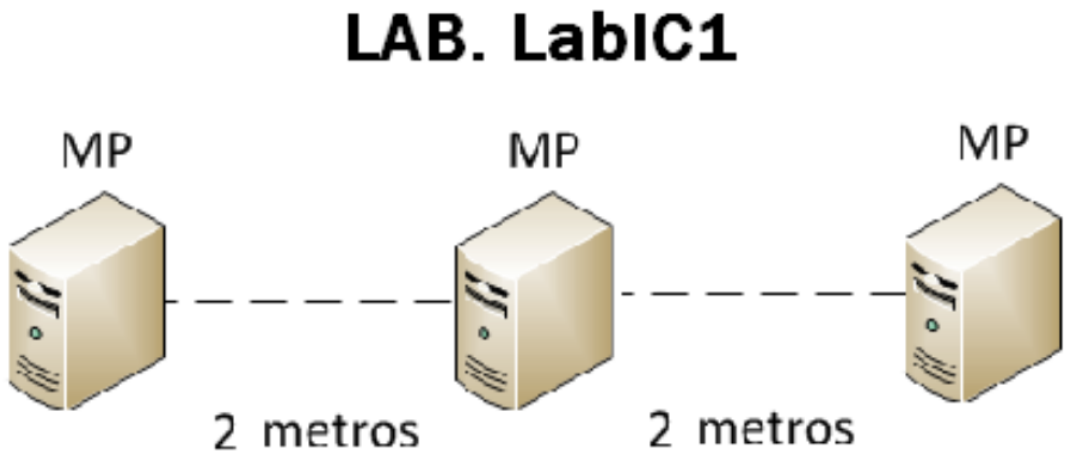

Figura 4.1: Disposição das estações mesh no LabIC.

O principal objetivo deste experimento foi demonstrar a rede mesh atuando com dois canais de transmissão. Também observar a mudança de canal ocorrendo durante uma transmissão de pacotes e o que isso pode ocasionar na transmissão.

4.1 Resultados

No primeiro cenário apresentado na figura 4.2, os controladores openflow foram progamados apenas para transmitir os quadros pela mesma porta de entrada no openvswitch. No primeiro teste foi utilizado apenas um canal de transmissão, ou seja, os quadros entravam e saíam pela mesma porta virtual do switch. O objetivo foi verificar apenas se o controlador era capaz de reencaminhar um quadro na rede.

No segundo cenário apresentado na figura 4.3 foi efetuado um teste de ping simples 4.4 onde o controlador recebe o quadro na porta 1 do switch virtual e encaminha o quadro pela porta porta virtual 2. Na figura 4.4 é possível observar na linha “chegou pacote de 112738716152 para tabela 0”. Essa linha mostra que um pacote chegou através do switch virtual “112738716152”

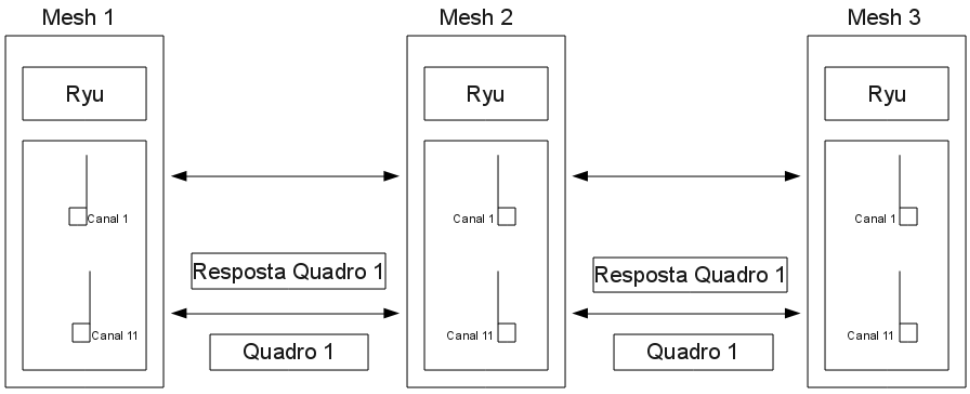

Figura 4.2: Resposta do pacote através do mesmo canal de transmissão.

que representa o número do switch Openflow conectado no controlador. Ainda na linha "...para tabela 0" significa que o pacote vai ter seus campos comparados com alguma regra da tabela openflow. Se a tabela Openflow não conter nenhuma regra de encaminhamento, o controlador vai ser acionado e tomará a decisão do encaminhamento na linha "Chegou quadro de 00:1A:3F:C0:27:1b para 00:1F:1F:12:3C:57 pelo port 2".

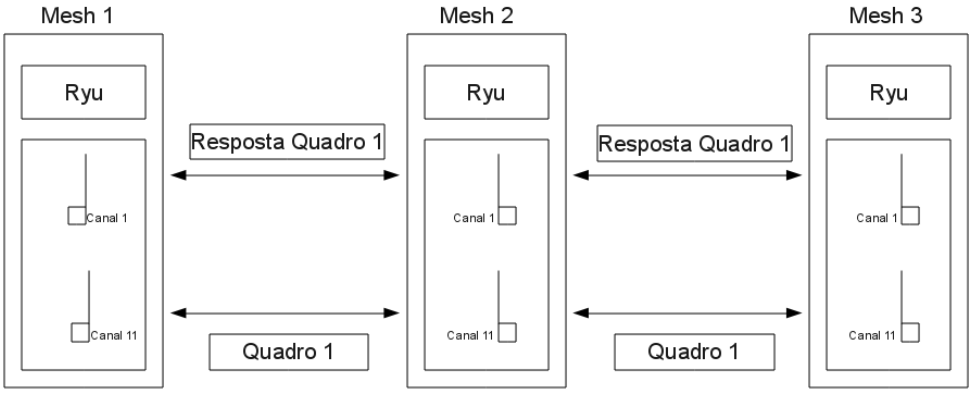

Figura 4.3: Resposta do pacote através de outro canal de transmissão.

```
chegou pacote de 112738716152 para tabela 0
Chegou quadro de 00:1A:3F:C0:27:1B para 00:1F:1F:12:3C:57 pelo port 2
```

Figura 4.4: Teste de encaminhamento entre duas estações mesh utilizando o controlador Openflow.

O controlador openflow reconhece os quadros como sendo quadros ethernet. Ele apenas leva em consideração o MAC de destino e de origem dos quadros para o encaminhamento. No exemplo da figura 4.5 é possível visualizar como o controlador programa uma regra de encaminhamento levando em consideração os endereços MACs do quadro. No exemplo da figura 4.5 é possível observar que uma regra foi programada na tabela openflow composta por campo de comparação (seletor da tabela openflow) como o MAC de origem ($dl\_src = 00:1a:3f:c0:27:1b$) e MAC de destino ($dl\_dst = 00:1f:1f:12:3c:57$). Se algum quadro possuir os MACs de origem e destinos continos na regra, a ação ($actions = output:2$) será executada e o quadro será encaminhado pela porta virtual 2 do switch que corresponde a um canal de transmissão da estação. Portanto através da programação das regras da tabela openflow

do switch é possível encaminhar quadros entre os diferentes canais de transmissão.

```
priority=123,in_port=1,dl_src=00:1a:3f:c0:27:1b,dl_dst=00:1f:1f:12:3c:57 actions
=output:2
```

Figura 4.5: Tabela Openflow com regras configuradas pelo controlador Openflow.

Vários testes de vazão foram efetuados na rede utilizando 1 e 2 canais de transmissão. Na figura 4.6 é possível visualizar o teste de vazão realizado entre duas estações utilizando um canal de trasmissão. A taxa de vazão obtida ficou em torno de 6Mbps que é a taxa referente a duas estações mesh se comunicando ponto a ponto. Para efetuar o teste de vazão foi utilizado o programa netperf (PACKARD, 2012) que é um software para teste de largura de banda entre duas estações na rede.

```
root@labic102:/home/aluno# netperf -f k -H 10.0.0.2 -l 10
MIGRATED TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 10.0.0.2 (10.0
.0.2) port 0 AF_INET : demo
Recv   Send    Send
Socket Socket  Message  Elapsed
Size   Size    Size     Time     Throughput
bytes  bytes   bytes    secs.    10^3bits/sec

 87380  16384  16384    10.12    5909.94
root@labic102:/home/aluno# netperf -f k -H 10.0.0.2 -l 10
MIGRATED TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 10.0.0.2 (10.0
.0.2) port 0 AF_INET : demo
Recv   Send    Send
Socket Socket  Message  Elapsed
Size   Size    Size     Time     Throughput
bytes  bytes   bytes    secs.    10^3bits/sec

 87380  16384  16384    11.36    6314.09
root@labic102:/home/aluno# netperf -f k -H 10.0.0.2 -l 10
MIGRATED TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 10.0.0.2 (10.0
.0.2) port 0 AF_INET : demo
Recv   Send    Send
Socket Socket  Message  Elapsed
Size   Size    Size     Time     Throughput
bytes  bytes   bytes    secs.    10^3bits/sec

 87380  16384  16384    10.52    6936.32
root@labic102:/home/aluno#
```

Figura 4.6: Teste de vazão utilizando um canal de transmissão.

Na figura 4.7 é apresentada as taxas de transmissão obtidas entre duas estações mesh ponto a ponto, porém cada uma delas utiliza dois canais de transmissão. Foi observado uma ligeira queda na taxa de transmissão que ficou em torno de 5Mbps. Para esse teste, a tabela openflow foi configurada com uma regra de encaminhamento de quadros que espirava a cada 2 segundos. Portanto a cada 2 segundo o controlador era acionado e então programava a regra novamente na tabela openflow do switch. Porém quando a regra era criada novamente, a porta de saída era alterada de modo que cada canal era utilizado apenas por 2 segundos. Essa constante mudança nas regras de encaminhamento da tabela openflow acabou gerando uma queda na taxa de transmissão. Isso foi causado porque quando a regra da tabela openflow expira, o controlador leva um tempo para programar outra regra, causando atrasos no encaminhamento. O que não ocorre na rede que utiliza apenas um canal.

Uma avaliação de desempenho da rede mesh implantada depende de testes de taxa de transmissão com redes maiores e também com situações mais realísticas, onde um canal sofre com

```
root@labic102:/home/aluno# netperf -f k -H 10.0.0.2 -l 10
MIGRATED TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 10.0.0.2 (10.0
.0.2) port 0 AF_INET : demo
Recv   Send    Send
Socket Socket  Message  Elapsed
Size   Size    Size     Time     Throughput
bytes  bytes   bytes    secs.    10^3bits/sec

 87380  16384  16384    10.21    5598.18
root@labic102:/home/aluno# netperf -f k -H 10.0.0.2 -l 10
MIGRATED TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 10.0.0.2 (10.0
.0.2) port 0 AF_INET : demo
Recv   Send    Send
Socket Socket  Message  Elapsed
Size   Size    Size     Time     Throughput
bytes  bytes   bytes    secs.    10^3bits/sec

 87380  16384  16384    12.22    5272.44
root@labic102:/home/aluno# netperf -f k -H 10.0.0.2 -l 10
MIGRATED TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 10.0.0.2 (10.0
.0.2) port 0 AF_INET : demo
Recv   Send    Send
Socket Socket  Message  Elapsed
Size   Size    Size     Time     Throughput
bytes  bytes   bytes    secs.    10^3bits/sec

 87380  16384  16384    12.11    4002.20
root@labic102:/home/aluno#
```

Figura 4.7: Teste de vazão utilizando dois canais de transmissão.

interferência ou então transmissões longas que atravessam várias estações na rede. Portanto os testes efetuados nesse trabalho não são suficientes para mensurar possíveis melhorias na taxa de transmissão da rede mesh. Os resultados da utilização de múltiplos canais na rede mesh devem ser investigados em trabalhos futuros levando em consideração a arquitetura desenvolvida nesse trabalho.

5 *Conclusões*

O objetivo desse trabalho foi mostrar que a rede mesh pode trabalhar com mais canais de transmissão. Na arquitetura proposta, uma estação mesh possui uma interface de rede sem-fio para cada canal a ser acessado. Essas interfaces são usadas como portas de um switch virtual, o qual é controlado por meio de um controlador Openflow. O controlador Openflow tem o papel de decidir por qual canal cada pacote deve ser transmitido, programando o switch virtual apropriadamente.

Para a utilização de dois canais em cada estação mesh, o switch openflow foi o componente com maior importância. Este componente possibilitou a utilização das portas virtuais e também encaminhar os pacotes entre os canais das estações através da programação da tabela openflow.

O controlador openflow também teve sua importância, porque sem ele não poderia ser feita a programação do encaminhamento de pacotes através do switch Openvswitch. Assim foi possível controlar todos os pacotes através da programação da tabela openflow de cada estação mesh contida na rede.

Através dos experimentos pode-se verificar que um pacote pode ser recebido por um canal distinto do canal de encaminhamento em uma estação mesh operando com os componentes citados acima.

5.1 Trabalhos futuros

A primeira opção para trabalhos futuros seria considerar a métrica e a fila de pacotes em cada canal de transmissão da estação mesh, para assim aumentar a taxa de transmissão efetuando balanceamento de carga entre os canais sem fio.

A segunda opção seria utilizar um controlador openflow centralizado que possua uma visão geral da rede, para assim então decidir quais canais determinadas estações devem utilizar em determinados momentos. Esse controlador poderia estar contido tanto dentro de uma estação na rede mesh ou em uma máquina rodando apenas o controlador.

A terceira opção seria desenvolver a arquitetura da estação mesh em um sistema embarcado com todos os componentes necessário sem a necessidade de um computador. Assim seria mais fácil implementar a rede mesh sem necessidade de muito espaço físico.

A última opção seria avaliar a capacidade resultante da rede mesh com dois ou mais canais de transmissão comparando capacidade de transmissão e outras características inerente da rede sem fio infra-estruturada.

Referências Bibliográficas

FOUNDATION, O. N. *Software-Defined Networking: The New Norm for Networks*. 2012. https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf. Acessado em 16 de Outubro de 2013.

MCKEOWN, N. et al. *OpenFlow: Enabling Innovation in Campus Networks*. 2008. http://archive.openflow.org/documents/openflow-wp-latest.pdf. Acessado em 16 de Outubro de 2013.

PACKARD, H. *Netperf*. 2012. http://www.netperf.org/netperf/. Acessado em 18 de Outubro de 2013.

PERKINS, C. E.; ROYER, E. M. *Ad-hoc On-Demand Distance Vector Routing*. 2003. http://www.cs.cornell.edu/people/egs/615/aodv.pdf. Acessado em 16 de Outubro de 2013.

PROJECT, R. *Controlador Ryu*. 2014. http://osrg.github.io/ryu/index.html. Acessado em 13 de Agosto de 2014.

SAADE, D. C. M. et al. *Multihop MAC: Desvendando o padrão 802.11s*. 2008. http://www2.ic.uff.br/~celio/papers/minicurso-sbrc08.pdf. Acessado em 18 de Outubro de 2013.

SCOTTI, K. *Balanceamento de Tráfego em LANS Ethernet usando uma abordagem SDN/Openflow*. 2013. http://wiki.sj.ifsc.edu.br/wiki/index.php/Balanceamento_de_tr%C3%A1fego_em_LANS_ethernet_usando_uma_abordagem_SDN/Openflow. Acessado em 18 de Outubro de 2013.

TEIXEIRA, E. R. D. *Wireless Mesh Networks*. 2004. http://www.teleco.com.br/tutoriais/tutorialwmn/. Acessado em 22 de Outubro de 2013.

WIKIPEDIA. *Rede Mesh*. 2014. http://pt.wikipedia.org/w/index.php?title=Redes_Mesh&oldid=39648862. Acessado em 12 de Agosto de 2014.