

TRABALHO DE GRADUAÇÃO

DESEMPENHO DE REDES *AD HOC*
PARA ESCOAMENTO DE TRÁFEGO
DE REDES CELULARES

Camila Lumy Nakano

Thayane Rodrigues Viana

Brasília, Julho de 2014

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

TRABALHO DE GRADUAÇÃO

**DESEMPENHO DE REDES *AD HOC*
PARA ESCOAMENTO DE TRÁFEGO
DE REDES CELULARES**

Camila Lumy Nakano

Thayane Rodrigues Viana

*Relatório submetido ao Departamento de Engenharia
Elétrica como requisito parcial para obtenção
do grau de Engenheiro de Redes de Comunicação*

Banca Examinadora

Prof. Marcelo Menezes de Carvalho, ENE/UnB _____
Orientador

Prof. Renato Mariz de Moraes, ENE/UnB _____
Examinador interno

Prof. Paulo Lira Gondim, ENE/UnB _____
Examinador interno

Dedicatórias

Dedico esse trabalho a todos os leitores que buscam mais conhecimento na área de Redes de Comunicação e a todos que acreditam na importância da educação para o desenvolvimento do nosso país.

Thayane Rodrigues Viana

Dedico esse trabalho a todos que dedicam seu tempo na missão de ensinar e aos que possam encontrar neste, alguma forma de conhecimento.

Camila Lumy Nakano

Agradecimentos

Agradeço primeiramente à minha família, por terem me dado tanto apoio para chegar até aqui, principalmente aos meus pais, Franki e Marta, que sempre me proporcionaram as melhores oportunidades de estudo e nunca me deixaram desistir da graduação mesmo em tempos difíceis e que me mostram como ser uma pessoa melhor todos os dias. Agradeço a todos os professores por compartilharem seus conhecimentos para me tornar uma boa profissional, especialmente ao professor Marcelo Carvalho, que foi muito além de um orientador e nos guiou com toda dedicação para que pudessemos concluir este trabalho. Agradeço à Thayane Viana, minha dupla que esteve sempre presente, mesmo nos momentos difíceis da realização deste trabalho. Agradeço à Larissa e ao Fadhil pelos conhecimentos que compartilharam conosco e ao Eduardo Vergne e ao Eduardo Vargas, colegas que estiveram o nosso lado durante todo o desenvolvimento do trabalho e nos ajudaram muito na construção deste. Por fim agradeço aos colegas que conheci ao longo do curso e que caminharam junto à mim nessa jornada e a todos os amigos que de alguma forma contribuíram para que eu pudesse concluir mais essa etapa da minha vida me proporcionando momentos de muita alegria.

Camila Lumy Nakano

Agradeço em primeiro lugar a Deus, pois sem Ele jamais teria conseguido chegar até aqui. Ele é a minha força, minha esperança e minha alegria, Ele me capacitou e me deu a oportunidade maravilhosa de poder fazer a minha graduação. A Ele seja toda honra e glória, pois Ele é o Criador de todas as coisas, o Alfa e o Ômega, o princípio e o fim, o Deus eterno. À minha família, que me apoiou e especialmente aos meus pais Nivaldo (in memoriam) e Geralda, que sempre lutaram para me dar a oportunidade de ter uma educação de qualidade e me deram palavras de ânimo, me ajudando nos momentos difíceis e me ensinando a lutar pelos meus objetivos. Agradeço também ao meu namorado, Lucas Parreira, pelo apoio e compreensão, e aos meus amigos, em especial aos do NVC - Núcleo de Vida Cristã da UnB, que estiveram ao meu lado durante toda minha graduação, me ensinando a cada dia e me fazendo estar mais perto de Deus. Também agradeço muito a todos os meus professores da graduação, especialmente ao meu professor e orientador Marcelo Carvalho, que acreditou em nós e nos ajudou tanto com paciência e dedicação; à minha dupla, Camila Nakano, pelo esforço e pelas madrugadas de sono perdidas. Agradeço também ao pessoal dos NERds - Núcleo de Estudos de Redes da UnB, em especial ao Fadhil e à Larissa, que nos ensinaram muito sobre o ns-3 e ao Eduardo Vergne e ao Eduardo Vargas, que deram muitas contribuições ao nosso projeto. Agradeço especialmente às minhas amigas e colegas de curso Ludimila, Marina e Vívian, por terem me incentivado, e por estarem ao meu lado em momentos alegres ou difíceis, sendo minhas melhores amigas durante toda a graduação. Por fim agradeço a todos aqueles que direta ou indiretamente contribuíram para o desenvolvimento desse trabalho.

Thayane Rodrigues Viana

RESUMO

Este trabalho faz a análise do desempenho de uma rede que escoar o tráfego de uma rede LTE (do inglês *Long Term Evolution*) para uma rede WiFi no modo *ad hoc*. A rede *ad hoc* possui mobilidade, ou seja, é uma Rede Móvel *Ad Hoc* (MANET do inglês *Mobile Ad hoc Network*) que utiliza o protocolo de roteamento OLSR (do inglês *Optimized Link State Routing*) para descobrir rotas e encaminhar pacotes de dados. A análise é feita em relação a diversas medidas de desempenho da rede: vazão, taxa de recebimento de pacotes, atraso na entrega dos pacotes e índice de justiça. Além destes também são avaliados em relação ao impacto no desempenho da rede: a taxa de envio de dados pela aplicação, o desvanecimento em relação ao parâmetro K de Rice e a velocidade dos nós com modelo de mobilidade *Random Direction 2d*. Para realizar este estudo, foi proposta a implementação de uma rede LTE com escoamento de tráfego para uma rede WiFi *ad hoc* e o ns-3 foi o simulador de rede utilizado para este trabalho. A avaliação em relação à mobilidade dos nós é feita com três valores distintos de velocidade. O desvanecimento é avaliado com três valores para o parâmetro K . Em relação à taxa de transmissão de dados pela aplicação, também são utilizadas três taxas para avaliação. Todo o tráfego é escoado por um nó proxy, que é o gargalo de todas as transmissões. Os resultados das simulações são apresentados em gráficos de: taxa de entrega de pacotes, vazão média por fluxo, atraso médio por pacote e índice de justiça, todos em relação ao número de fluxos. Cada gráfico apresenta o resultado para as três velocidades, com determinada taxa de envio de dados e determinado K . Por fim, é apresentada uma análise da viabilidade da implementação de uma rede com tais características através dos resultados de desempenho apresentados com a variação dos diversos parâmetros.

ABSTRACT

This work studies the performance of data offloading to WiFi network from LTE (Long Term Evolution) network. The WiFi network is a MANET (Mobile Ad Hoc Network) and uses OLSR (Optimized Link State Routing) as the routing protocol to discover routes and send data packets. The Analysis is done regarding various measures of network performance: throughput, packet delivery ratio, delay on packet delivery and fairness. In addition, according to the impact in network performance are evaluated: application transmission data rate, fading with Rice's parameter K and the speed of the nodes with the Random Direction 2d mobility model. To make these studies, an implementation was suggested with data offloading from LTE network to WiFi *ad hoc* network and the ns-3 was the network simulator used for this work. The evaluation according to nodes' mobility is done with three different values of speed. Fading is evaluated with three values of K . According to application transmission data rate three values were used. All traffic is offloaded by a proxy node that works like bottleneck in all transmissions. The results of the simulations are presented in graphics of packet delivery ratio, average throughput by flow, average delay by packet and fairness, all of them according to the number of flows. Each graphic presents the result to all three speeds, with one certain transmission data rate and value of K . Lastly, an analysis of the implementation of a network with these features is made through the results of network performance with variation of the parameters.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	OBJETIVOS	4
1.2	ESTRUTURA DO PROJETO	4
2	FUNDAMENTAÇÃO TEÓRICA	5
2.1	ESCOAMENTO DE DADOS MÓVEIS	5
2.2	LONG TERM EVOLUTION - LTE	8
2.2.1	ARQUITETURA DO LTE	9
2.2.2	LTE-ADVANCED	10
2.3	O PADRÃO IEEE 802.11	12
2.3.1	ARQUITETURA	13
2.3.2	CAMADA FÍSICA	14
2.3.3	SUB-CAMADA DE CONTROLE DE ACESSO AO MEIO (MAC)	15
2.4	REDES MÓVEIS <i>Ad Hoc</i>	18
2.4.1	DESAFIOS EM MANETS	19
2.4.2	PADRÃO IEEE 802.11 PARA MANETS	20
2.4.3	PROTOCOLOS DE ROTEAMENTO PARA MANETS	20
2.4.4	MANETS E 4G	21
2.5	PROTOCOLO DE ROTEAMENTO OLSR	21
2.5.1	<i>MultiPoint Relays</i> - MPRs	23
2.5.2	SENSORIAMENTO DOS NÓS VIZINHOS	23
2.5.3	SELEÇÃO E DECLARAÇÃO DE MPR	24
2.5.4	CÁLCULO DA TABELA DE ROTEAMENTO	25
2.6	O SIMULADOR NS-3	25
2.7	REVISÃO BIBLIOGRÁFICA	26
3	ESCOAMENTO DE TRÁFEGO VIA REDES <i>Ad Hoc</i>	28
3.1	O CENÁRIO DE ESCOAMENTO DE TRÁFEGO	28
3.2	CENÁRIO DE SIMULAÇÃO	30
3.2.1	LTE NO NS-3	30
3.2.2	CAMADA FÍSICA	31
3.2.3	CAMADA DE ENLACE	33
3.2.4	CAMADA DE TRANSPORTE	33

3.2.5	CAMADA DE REDE.....	33
3.2.6	CAMADA DE APLICAÇÃO	34
4	AVALIAÇÃO DE DESEMPENHO	36
4.1	MEDIDAS DE DESEMPENHO	36
4.2	VARIAÇÃO DE PARÂMETROS	37
4.3	RESULTADOS DE SIMULAÇÕES	38
4.3.1	TAXA DE ENTREGA DE PACOTES	38
4.3.2	VAZÃO MÉDIA POR FLUXO.....	45
4.3.3	ATRASSO MÉDIO POR PACOTE	50
4.3.4	ÍNDICE DE JUSTIÇA.....	55
5	CONCLUSÕES	61
	ANEXOS.....	66
I	SCRIPT	67

LISTA DE FIGURAS

1.1	Previsão do crescimento de tráfego de dados móveis até 2018 [1].	2
1.2	Previsão do crescimento da participação da tecnologia 4G na geração de dados até 2018, em comparação com outras tecnologias [1].	3
1.3	Projeção do descarregamento de dados móveis para os próximos anos [1].	3
2.1	(a) Rede celular e (b) Escoamento de dados móveis para um rede WiFi.	6
2.2	Arquitetura plana de eNBs (<i>evolved NodeB</i>) para acesso à rede LTE.....	9
2.3	Estrutura do EPS (<i>Evolved Packet System</i>), formado pelo EPC e pelo E-UTRAN. ...	10
2.4	Na figura, CP é o componente de portadora, (a) Representa a agregação de portadora contígua na banda, (b) representa agregação não-contígua na mesma banda e (c) agregação não-contígua em múltiplas bandas.	11
2.5	Arquitetura básica do repasse no LTE-Advanced.....	12
2.6	Rede WiFi em modo <i>ad hoc</i> . Os dispositivos se comunicam diretamente uns com os outros, sem necessidade de um AP central.	14
2.7	Rede em modo infraestrutura. O nó central é o AP, por onde todo tráfego é canalizado. Os dispositivos estão conectados ao AP e se comunicam uns com os outros através dele.	14
2.8	Problema do terminal oculto: o terminal A está fora da faixa de cobertura do terminal C. O terminal C não pode então detectar uma transmissão de A para B e pode tentar transmitir um quadro para B simultaneamente com a transmissão de A para B, gerando colisão.....	17
2.9	Envio de RTS e CTS: o terminal fonte espera um tempo DIFS e envia um quadro RTS ao terminal de destino. O terminal de destino recebe o quadro RTS e então envia um quadro CTS para o terminal fonte após um período SIFS. O terminal fonte recebe o CTS e então envia o quadro de dados após um período SIFS. O nó de destino recebe o quadro de dados e envia um ACK para o nó fonte após um período SIFS.	18
2.10	Seleção de MPRs: o nó fonte seleciona apenas alguns nós vizinhos para a retransmissão dos pacotes de controle para nós a dois saltos do nó fonte.	23
3.1	Topologia montada: a estrutura LTE contém o nó remoto, o P-GW, a eNB e o UE. O UE é o nó proxy que escoar o tráfego para a rede WiFi. A rede WiFi contém os nós se movendo e transferindo dados uns para os outros através de uma rede <i>ad hoc</i>	31

3.2	Arquivo de <i>pcap</i> obtido através do software de análise de pacotes Wireshark. É possível visualizar os pacotes UDP enviados do nó fonte até o nó de destino.....	34
4.1	Taxa de Entrega de Pacotes para Cenário 1: $K = \infty$ e taxa= 117.760 b/s.....	39
4.2	Taxa de Entrega de Pacotes para Cenário 2: $K = 2$ e taxa= 117.760 b/s.	40
4.3	Taxa de Entrega de Pacotes para Cenário 3: $K = 0$ e taxa= 117.760 b/s.	40
4.4	Taxa de Entrega de Pacotes para Cenário 4: $K = \infty$ e taxa= 176.640 b/s.....	41
4.5	Taxa de Entrega de Pacotes para Cenário 5: $K = 2$ e taxa= 176.640 b/s.	41
4.6	Taxa de Entrega de Pacotes para Cenário 6: $K = 0$ e taxa= 176.640 b/s.	42
4.7	Taxa de Entrega de Pacotes para Cenário 7: $K = \infty$ e taxa= 235.520 b/s.....	43
4.8	Taxa de Entrega de Pacotes para Cenário 8: $K = 2$ e taxa= 235.520 b/s.	43
4.9	Taxa de Entrega de Pacotes para Cenário 9: $K = 0$ e taxa= 235.520 b/s.	44
4.10	Vazão média por fluxo para Cenário 1: $K = \infty$ e taxa= 117.760 b/s.....	45
4.11	Vazão média por fluxo para Cenário 2: $K = 2$ e taxa= 117.760 b/s.	46
4.12	Vazão média por fluxo para Cenário 3: $K = 0$ e taxa= 117.760 b/s.	46
4.13	Vazão média por fluxo para Cenário 4: $K = \infty$ e taxa= 176.640 b/s.....	47
4.14	Vazão média por fluxo para Cenário 5: $K = 2$ e taxa= 176.640 b/s.	47
4.15	Vazão média por fluxo para Cenário 6: $K = 0$ e taxa= 176.640 b/s.	48
4.16	Vazão média por fluxo para Cenário 7: $K = \infty$ e taxa= 235.520 b/s.....	48
4.17	Vazão média por fluxo para Cenário 8: $K = 2$ e taxa= 235.520 b/s.	49
4.18	Vazão média por fluxo para Cenário 9: $K = 0$ e taxa= 235.520 b/s.	49
4.19	Atraso médio por pacote para Cenário 1: $K = \infty$ e taxa= 117.760 b/s.....	50
4.20	Atraso médio por pacote para Cenário 2: $K = 2$ e taxa= 117.760 b/s.	51
4.21	Atraso médio por pacote para Cenário 3: $K = 0$ e taxa= 117.760 b/s.	51
4.22	Atraso médio por pacote para Cenário 4: $K = \infty$ e taxa= 176.640 b/s.....	52
4.23	Atraso médio por pacote para Cenário 5: $K = 2$ e taxa= 176.640 b/s.	53
4.24	Atraso médio por pacote para Cenário 6: $K = 0$ e taxa= 176.640 b/s.	53
4.25	Atraso médio por pacote para Cenário 7: $K = \infty$ e taxa= 235.520 b/s.....	54
4.26	Atraso médio por pacote para Cenário 8: $K = 2$ e taxa= 235.520 b/s.	54
4.27	Atraso médio por pacote para Cenário 9: $K = 0$ e taxa= 235.520 b/s.	55
4.28	Índice de justiça para Cenário 1: $K = \infty$ e taxa= 117.760 b/s.....	56
4.29	Índice de justiça por pacote para Cenário 2: $K = 2$ e taxa= 117.760 b/s.	57
4.30	Índice de justiça para Cenário 3: $K = 0$ e taxa= 117.760 b/s.	57
4.31	Índice de justiça para Cenário 4: $K = \infty$ e taxa= 176.640 b/s.....	58
4.32	Índice de justiça para Cenário 5: $K = 2$ e taxa= 176.640 b/s.	58
4.33	Índice de justiça para Cenário 6: $K = 0$ e taxa= 176.640 b/s.	59
4.34	Índice de justiça para Cenário 7: $K = \infty$ e taxa= 235.520 b/s.....	59
4.35	Índice de justiça para Cenário 8: $K = 2$ e taxa= 235.520 b/s.	60
4.36	Índice de justiça para Cenário 9: $K = 0$ e taxa= 235.520 b/s.	60

LISTA DE TABELAS

3.1	Parâmetros das simulações.	30
3.2	Parâmetros de simulação para camada PHY da rede WiFi.	32
4.1	Cenários de simulação utilizados.	38

LISTA DE ABREVIATURAS

Acrônimos

3GPP	Third-Generation Partnership Project
ACK	Acknowledgement
ANDSF	Discovery Network Access and Selection Function
AODV	Ad hoc On Demand Distance Vector
APs	Access Point
BSA	Basic Area Service
BSS	Basic Service Set
CBR	Constant Bit Rate
CCA	Clear Channel Assignment
CFP	Contention-Free Period
CP	Contention Period
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CSMA/CD	Carrier Sense Multiple Access With Collision Detection
CTS	Clear to Send
DBPSK	Differential Binary Phase Shift Keying
DCF	Distributed Coordination Function
DeNB	Donor evolved NodeB
DIFS	Distributed Inter-Frame Space
DQPSK	Differential Quadrature Phase Shift Keying
DS	Distribution System
DSDV	Destination-Sequenced Distance-Vector
DSR	Dynamic Source Routing
DSSS	Direct-Sequence Spread-Spectrum
DTLS	Datagram Transport Layer Security
EAP	Extensible Authentication Protocol
eNBs	evolved Nodeb
EPC	Evolved Packet Core
EPS	Evolved Packet System
ESS	Extended Service Set
E-UTRA	Evolved UMTS Terrestrial Radio Access
E-UTRAN	Evolved UMTS Terrestrial Radio Access Network
FHSS	Frequency-Hopping Spread-Spectrum
HDR	High Data Rate
HSS	Home Subscriber Service

Acrônimos

IP	Internet Protocol
IR	Infrared
ISM	Industrial, Scientific and Medical
LAN	Local Area Network
LTE	Long Term Evolution
MAC	Medium Access Control
MANETs	Mobile Ad hoc Networks
MIMO	Multiple-Input Multiple-Output
MME	Mobility Management Entity
MPR	MultiPoint Relays
NS-3	Network Simulator 3
OFDM	Orthogonal Frequency Division Multiplexing
OFDMA	Orthogonal Frequency Division Multiplexing Access
OLSR	Optimized Link State Routing
OTDOA	Observed Time Difference Of Arrival
PAN	Personal Area Network
PCRF	Policy Control and Charging Rules Function
PDCP	Packet Data Convergence Protocol
PDNs	Packet Data Networks
PDR	Packet Delivery Ratio
PDU	Protocol Data Unit
P-GW	PDN Gateway
PHY	Physical Layer (Camada física)
PPM	Pulse Position Modulation
QoS	Quality of Service
RN	Relay Node
RLC	Radio Link Control
RTS	Request to Send
SAE	Architecture Evolution System
SC-FDMA	Single Carrier Frequency Division Multiple Access
S-GW	Serving Gateway
SIFS	Short Inter Frame Space
TC	Topology Control
TCP	Transport Control Protocol
UCAN	Unified Cellular and Ad-Hoc Network
UDP	User Datagram Protocol
UE	User Equipment
UMTS	Universal Mobile Telecommunication System
VoIP	Voice over IP
WAN	Wide Area Network
WiFi	Wireless Fidelity
WLAN	Wireless Local Area Network

Capítulo 1

Introdução

Atualmente é indiscutível o crescente aumento do tráfego de dados de redes móveis. A demanda dos usuários de dispositivos móveis como *tablets* e *smartphones* é cada vez maior, assim como o número de estudos e pesquisas na área de tecnologias de redes móveis. Entre as tecnologias desenvolvidas ou em desenvolvimento estão o escoamento de dados móveis, o LTE (do inglês *Long Term Evolution*) ou 4G e as Redes *Ad Hoc*.

Nos últimos anos, o uso de Internet móvel tem crescido muito. A popularidade dos *smartphones* e *tablets* tem proporcionado um aumento da demanda por um serviço de Internet móvel rápida. Muitas aplicações para *smartphones* têm sido desenvolvidas e o acesso à Internet por dispositivos móveis tem sido algo muito frequente e muitas vezes indispensável.

Todos os anos novos dispositivos sem fio de diferentes marcas e modelos estão sendo introduzidos no mercado. Esse crescimento é um dos principais contribuintes para o crescimento do tráfego global móvel. Esses dispositivos possuem recursos de conexão de rede que criam uma crescente demanda por redes mais capazes e inteligentes. Além disso a explosão de aplicativos móveis e a implementação de conectividade móvel gera a necessidade de um melhor gerenciamento de banda e de uma maior implementação do serviço 3G/4G.

Segundo as previsões divulgadas em Fevereiro de 2014 pela empresa Cisco, no documento *Cisco Visual Networking Index (VNI)*, o tráfego de dados móveis global deve crescer para 15,9 exabytes por mês em 2018, um aumento de quase 11 vezes em relação a 2013 [1]. O tráfego de dados móveis deve crescer a uma taxa anual média de 61% de 2013 para 2018, como pode ser visto no gráfico da Figura 1.1.

Os dispositivos móveis estão ficando cada vez mais sofisticados, suportando aplicações multimídias avançadas que contribuem para o aumento do tráfego móvel 3G/4G e WiFi (do inglês *Wireless Fidelity*). Dessa forma, os provedores de serviços 3G e 4G ao redor do mundo precisam atender à crescente demanda do usuário por mais largura de banda, maior segurança e conectividade mais rápida em movimento.

Como pode ser visto no gráfico da Figura 1.2, o uso de tecnologias 4G tende a crescer muito nos próximos anos. A tecnologia LTE de 4G pode fornecer altas taxas de dados, menor latência

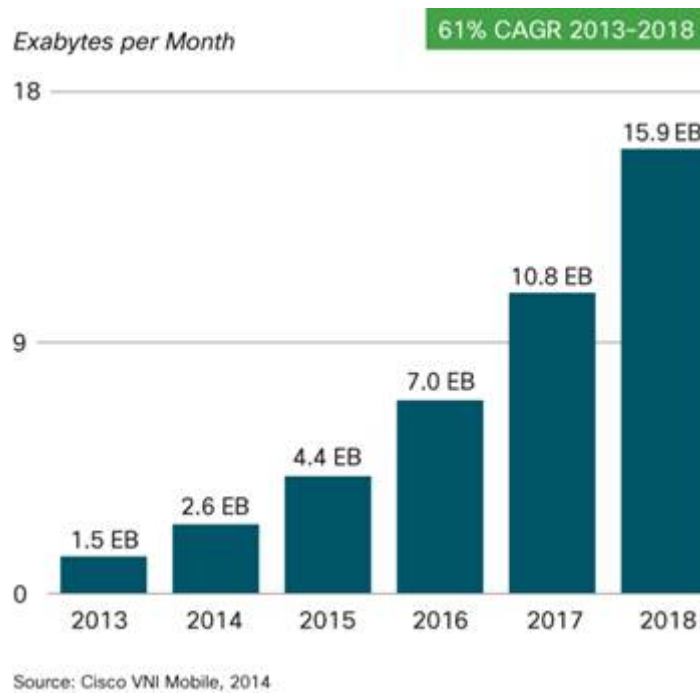


Figura 1.1: Previsão do crescimento de tráfego de dados móveis até 2018 [1].

e maior eficiência espectral. O LTE tem sido considerado a tecnologia móvel do futuro [2], e o seu impacto no tráfego móvel tende a crescer exponencialmente, como pode ser visto no gráfico da Figura 1.2. A maioria das operadoras e fabricantes de dispositivos móveis têm investido e apoiado o LTE. Essa tecnologia será abordada com mais detalhes nas próximas seções.

Considerando os dados sobre o crescimento do tráfego móvel nos últimos anos, torna-se preocupante para as operadoras como esse acesso poderá ser provido por rede celular, dadas as limitações em infraestrutura e disponibilidade de faixas do espectro eletromagnético. Assim, as atuais redes celulares não têm capacidade de suportar todo o rápido crescimento de dados móveis. Para evitar a sobrecarga das redes celulares, soluções de escoamento de tráfego de dados móveis têm sido propostas de forma a descarregar parte do tráfego celular em redes WiFi, por exemplo. Essa seria uma solução de baixo custo e de fácil implementação. A Figura 1.3 mostra a previsão de crescimento do descarregamento de dados móveis, relativo à mudança de conexão celular para WiFi incluindo os tráfegos de pontos de acesso públicos bem como redes WiFi residenciais.

Muitos trabalhos têm sido produzidos sobre escoamento de dados móveis e várias soluções têm sido propostas a fim de aumentar a taxa de transferência nas células e manter a qualidade de serviço. O escoamento de tráfego das operadoras de telefonia móvel para WiFi se apresenta como uma opção nova e promissora, de forma a complementar a rede celular para entregar dados tolerantes a atraso a determinada parcela de usuários da rede celular. Algumas dessas propostas serão analisadas na Seção 2.1.

Nesse trabalho é apresentada uma análise de desempenho de uma rede em que há o escoamento do tráfego de dados móveis de uma rede 4G para uma rede WiFi em modo *ad hoc*. Implementamos

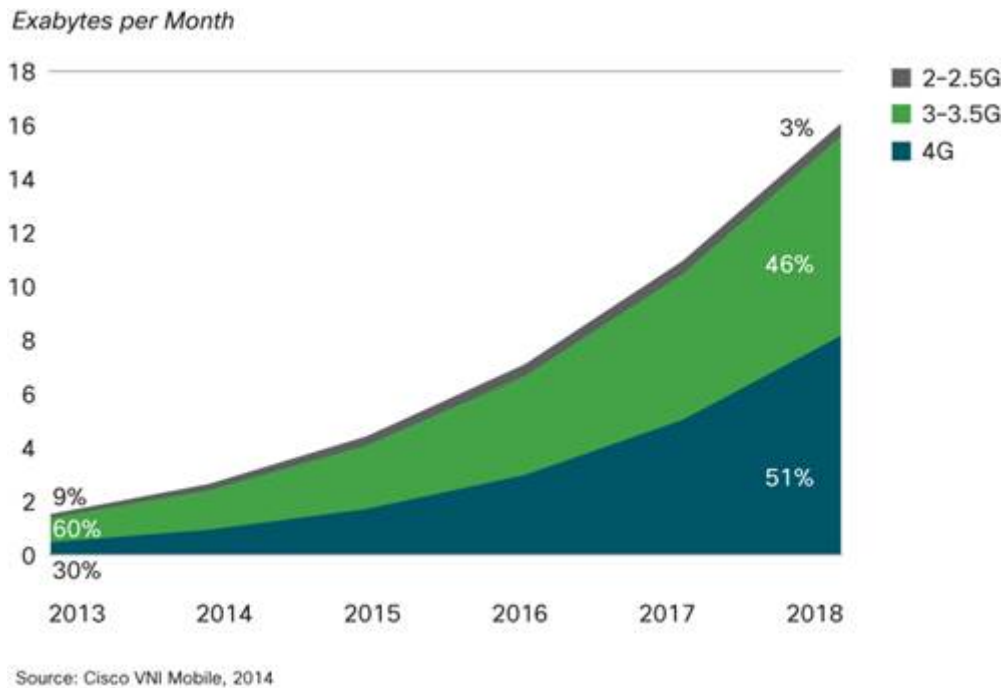


Figura 1.2: Previsão do crescimento da participação da tecnologia 4G na geração de dados até 2018, em comparação com outras tecnologias [1].



Figura 1.3: Projeção do descarregamento de dados móveis para os próximos anos [1].

uma rede *ad hoc* que utiliza o protocolo de roteamento OLSR (do inglês *Optimized Link State Routing*) para descobrimento de rotas e encaminhamento de pacotes. Nosso estudo se concentra em avaliar o desempenho da rede simulada através dos resultados obtidos com as medidas efetuadas.

1.1 Objetivos

Os principais objetivos desse trabalho são apresentados a seguir:

- Implementação no ns-3 do escoamento de tráfego de dados móveis de uma rede LTE/4G para uma rede WiFi em modo *ad hoc*;
- Simulação no ns-3 da rede com diversos cenários, variando-se a taxa de envio de dados, velocidade dos nós, quantidade de fluxos e parâmetro de desvanecimento;
- Medição de Taxa de Recebimento de Pacotes, Vazão Fim-a-Fim, Atraso Médio por Pacote e Índice de Justiça da rede através dos resultados obtidos;
- Representação gráfica dos resultados obtidos para os diferentes parâmetros medidos e para os diversos cenários de simulação montados;
- Análise do desempenho da rede através dos resultados gráficos obtidos no trabalho;
- Análise da viabilidade da rede implementada de acordo com o desempenho obtido.

1.2 Estrutura do Projeto

No Capítulo 2, Fundamentação Teórica, são abordados os principais fundamentos estudados e utilizados nesse projeto. Na Seção 2.1 é dada uma explicação sobre o descarregamento de dados móveis, que é um tema que recentemente tem sido pesquisado e abordado nos últimos anos. Na Seção 2.2 é abordada a tecnologia LTE. Na Seção 2.3 são apresentadas as especificações do padrão IEEE 802.11 e sua utilização. Na Seção 2.4 é dada uma explicação básica sobre a estrutura das redes *ad hoc*. Na Seção 2.5 é apresentado o protocolo de roteamento OLSR, que foi utilizado nesse projeto.

No Capítulo 3, Implementação no ns-3, é apresentada a estrutura montada nas simulações para o descarregamento de dados móveis de uma rede LTE para uma rede *ad hoc*. Nesse capítulo é abordado a implementação dessa rede no simulador de redes ns-3, contendo uma explicação sucinta de como o ns-3 foi utilizado para a realização das simulações.

No Capítulo 4, Avaliação de Desempenho, são expostos os resultados obtidos com as simulações. Os resultados são analisados através dos gráficos obtidos com os dados capturados. Assim, esse capítulo contém uma análise de cada parâmetro medido com as simulações, comparando os resultados e expondo as nossas conclusões sobre eles.

Por último, o Capítulo 5, Conclusão, contém as principais conclusões obtidas com as simulações e os resultados analisados. Esse capítulo também contém as perspectivas de trabalhos futuros a partir desse projeto.

Capítulo 2

Fundamentação Teórica

Esse capítulo apresenta os principais fundamentos teóricos nos quais esse trabalho foi embasado. Apresentamos nesse capítulo os conceitos e mecanismos de funcionamento das tecnologias e dos recursos utilizados ao longo do desenvolvimento desse trabalho. Assim, os conceitos mais estudados e necessários para o desenvolvimento do projeto são expostos de forma a trazer um conhecimento básico teórico para o entendimento da rede implementada e analisada através das simulações realizadas.

2.1 Escoamento de Dados Móveis

O objetivo do escoamento de rede é descarregar o tráfego celular 3G/4G através de redes complementares. Assim, os dados originalmente direcionados para redes celulares são entregues através de redes complementares, aliviando o congestionamento das redes celulares, como pode ser visto na Figura 2.1, em que no cenário (a) é possível observar uma rede em que o tráfego de dados é enviado diretamente da torre de transmissão para cada dispositivo por rede celular. Já no cenário (b) apenas alguns dispositivos recebem o tráfego diretamente pela rede celular e funcionam como gargalo para que os outros dispositivos recebam o tráfego, transmitindo os dados através de uma rede WiFi. As principais tecnologias consideradas para descarregar os dados móveis são WiFi, *femtocell* e *Integrated Mobile Broadcast*. Nesse trabalho será abordado o escoamento de tráfego para redes WiFi.

WiFi é uma tecnologia WLAN (do inglês *Wireless Local Area Network*) baseada no padrão IEEE 802.11, que será abordado na Seção 2.3. Hoje em dia o WiFi tem ganho muita popularidade e é usado em uma variedade de localidades, como em casas, comércios, universidades, restaurantes e aeroportos. As redes WiFi trabalham na banda de frequência não-licenciada ISM (do inglês *Industrial, Scientific and Medical*), portanto não causam interferência em redes 3G/4G.

Como a maioria dos dispositivos móveis já vem com a interface WiFi instalada, é possível realizar a mudança de conexão celular para WiFi de forma simples e barata. O WiFi é visto pelas operadoras como um meio rentável de descarregar grande quantidade de tráfego de dados móveis, pois já é implementado amplamente, é suportado por grande parte dos dispositivos móveis e não

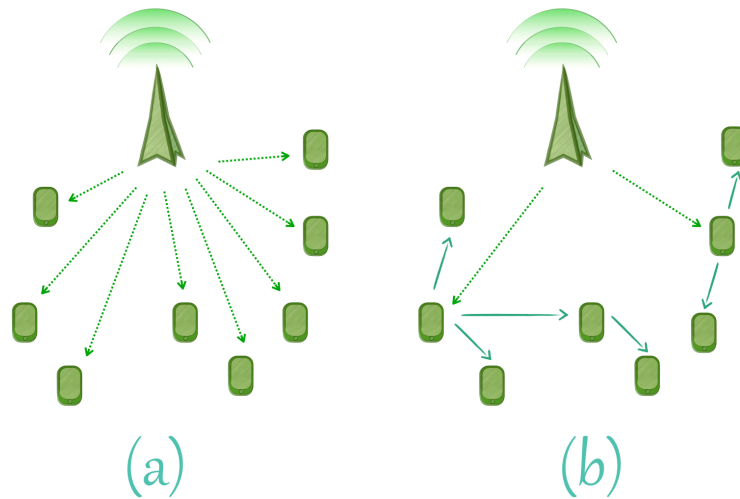


Figura 2.1: (a) Rede celular e (b) Escoamento de dados móveis para um rede WiFi.

requer assinatura móvel.

As operadoras podem mover o acesso dos usuários da rede 3G/4G para redes WiFi adicionando mais capacidade de forma acessível e flexível. Assim, pode-se implementar o escoamento de tráfego para redes WiFi em áreas com intenso uso de banda larga móvel, como universidades, estádios, shoppings, e assim conseguir reduzir os custos para oferecer um melhor serviço aos usuários. As operadoras de telefonia móvel podem entrar em parceria com provedores de WiFi existentes ou podem construir sua própria rede WiFi, já que redes WiFi oferecem um baixo custo de implantação. Alguns dos desafios para que as operadoras implementem o escoamento do tráfego para redes WiFi estão relacionados a questões de como selecionar dispositivos móveis para descarregar parte do tráfego 3G/4G e como estabelecer um método de autenticação para o acesso dos usuários [3].

Para descarregar o tráfego em redes WiFi, os dispositivos móveis habilitados para o uso de WiFi e com sistemas operacionais avançados como Android, BlackBerry, iOS da Apple e Windows Mobile tentarão passar automaticamente das conexão de dados 3G/4G para uma rede WiFi, assim que estiver ao alcance. Com a futura norma 802.11u, os mecanismos de seleção de redes serão melhorados, fornecendo uma seleção de redes WiFi para o escoamento do tráfego sem qualquer interação do usuário. Assim, os dispositivos precisarão de um tipo de gerenciador de conexões para tornar esse processo automático e transparente para o usuário. Esse gerenciador de conexões precisará então decidir onde e quando o descarregamento de 3G/4G para WiFi (e vice-versa) deve ocorrer. Para controlar esse mecanismo, existem políticas definidas pelo 3GPP (do inglês *Third-Generation Partnership Project*), contidas no ANDSF (do inglês *Discovery Network Access and Selection Function*) que comunicam as decisões para o gerenciador de conexões. O 3GPP é um projeto que reúne organizações que desenvolvem padrões para telecomunicações, abrangendo tecnologias de rede de telefonia celular, incluindo o acesso a rádio e a interação com redes WiFi. O ANDSF é uma entidade dentro do EPC (do inglês *Evolved Packet Core*) do SAE (do inglês *Architecture Evolution System*) para redes móveis compatíveis com 3GPP. O objetivo do ANDSF

é ajudar o equipamento do usuário (UE do inglês *User Equipment*) a descobrir redes de acesso que não sejam 3GPP, tais como WiFi ou WiMAX para oferecer ao UE regras para controle de políticas de acesso à rede. Assim, dispositivos como notebooks, modems e telefones celulares podem descobrir e se comunicar com redes que não sejam 3GPP. O ANDSF permite que as operadoras móveis possam coletar informações em tempo real sobre as condições da rede do assinante. Dessa forma, as operadoras de telefonia móvel não perdem a visibilidade da atividade de seus assinantes quando esses mudam para uma rede WiFi, por exemplo [4].

Quando o dispositivo for selecionado para a rede WiFi, o usuário deve ser autenticado para ser admitido à rede. Essa autenticação deve ser feita de forma rápida e fácil para não causar atraso ao descarregamento. Algumas empresas utilizam o protocolo EAP-SIM (do inglês *Extensible Authentication Protocol - Subscriber Identification Module*), para que a autenticação seja feita através de um algoritmo de autenticação SIM entre o cliente e o servidor, sem a necessidade de uma senha pré-estabelecida entre eles.

O 3GPP diferencia dois tipos de acesso WiFi: confiável e não-confiável. O acesso não confiável inclui qualquer tipo de acesso WiFi que não está sob o controle do operador, por exemplo: *hotspot* público aberto e acesso residencial em WLAN, que normalmente não oferecem segurança suficiente com autenticação, criptografia, etc. Essa arquitetura de acesso permite aos assinantes utilizarem qualquer tipo de rede de acesso que possam se conectar. Sua especificação é feita no documento 3GPP TS 23.234 para 3G e no 3GPP TS 23.402 para LTE. O acesso confiável se refere às redes WiFi implantadas e gerenciadas pela operadora com método de autenticação segura e criptografia dos dados, através de autenticação baseada no IEEE 802.1x, criptografia baseada no IEEE 802.11i com uso opcional de controle e provisionamento de pontos de acesso WiFi e do protocolo DTLS (do inglês *Datagram Transport Layer Security*). O foco desse trabalho não é o aprofundamento em questões de autenticação, mais detalhes sobre esse assunto são abordados em [5].

Há três abordagens principais para descarregar o tráfego de dados em redes WiFi, que dependem do grau de interação entre WiFi e redes celulares [6]. A primeira abordagem é o caso do escoamento não-gerenciado, em que o escoamento ocorre de maneira transparente ao usuário sempre que houver cobertura WiFi. Assim, a rede celular é ignorada para o serviço de dados, mas o serviço de voz continua a ser entregue através da rede principal. Essa seria uma solução de fácil implantação, porém traz a desvantagem das operadoras perderem a visibilidade dos seus assinantes enquanto eles estiverem usando o WiFi gratuito, por exemplo. Outra abordagem, para que as operadoras não percam o controle sobre os assinantes, é conseguida colocando um *gateway* de reconhecimento da sessão WiFi do assinante. Mas, ainda assim, a operadora teria controle sobre o seu assinante, porém não poderia entregar algum conteúdo contratado. Existe ainda outra abordagem integrada de escoamento de dados em que a operadora tem controle total sobre seus assinantes e tem a capacidade de entregar qualquer conteúdo contratado mesmo quando o assinante estiver utilizando WiFi gratuito. Isso é feito através da integração de redes celulares e WiFi, formando uma ponte entre as duas redes através da qual o fluxo de dados possa ser estabelecido. Para essa arquitetura, as duas redes podem estar ligadas indiretamente de forma independente ou podem ser ligadas por um sistema acoplado, compartilhando um núcleo comum para a maioria das funções de redes, gerenciando os recursos de forma centralizada. Nesse trabalho é utilizada a primeira abordagem,

de escoamento não-gerenciado.

2.2 Long Term Evolution - LTE

O *Long Term Evolution*, também conhecido como 4G, é a próxima geração de tecnologias de rádio para aumentar a capacidade e a velocidade de redes de comunicações móveis celulares, ou seja, é um conjunto de melhorias para o UMTS (do inglês *Universal Mobile Telecommunication System*). O padrão LTE tem sido desenvolvido pelo projeto 3GPP e está especificado no *Release 8* com melhorias especificadas nos *Release 9* e *Release 10*. Utiliza modulação OFDM para o *downlink*, ou seja, para comunicação no sentido da estação rádio-base para o UE, e o SC-FDMA (do inglês *Single Carrier Frequency Division Multiple Access*) para o *uplink*, ou seja, para comunicação no sentido do UE para a estação rádio-base. Em comparação com os antigos sistemas 3G, o LTE utiliza interface aérea e estrutura de pacote diferenciadas. O E-UTRA (do inglês *Evolved UMTS Terrestrial Radio Access*) é a interface aérea utilizada no LTE, a qual é baseada no OFDMA (do inglês *Orthogonal Frequency Division Multiple Access*). O LTE é inteiramente baseado em pacotes IP (do inglês *Internet Protocol*) e VoIP (do inglês *Voice over IP*).

A parte IP do LTE é chamada EPS (do inglês *Evolved Packet System*). Embora o LTE seja frequentemente comercializado como 4G, o primeiro *release* deste não obedece completamente os requisitos de 4G do *International Mobile Telecommunications (IMT) Advanced*. O padrão pré-4G está um passo atrás do LTE-Advanced, um padrão 4G de tecnologia de rádio, desenvolvida para incrementar a capacidade e a velocidade de redes de telefonia móvel. O LTE-Advanced é compatível com o LTE e utiliza as mesmas faixas de frequência, enquanto o LTE não é compatível com sistemas 3G [7].

O LTE *Release 8* é implementado principalmente em uma configuração de macrocélulas ou microcélulas. Provê redução de custos operacionais, operação da largura de banda flexível e integração com sistemas já existentes. Além disso, o LTE *Release 8* também provê elevadas taxas de dados de usuários em combinação com baixa latência baseada em MIMO (do inglês *Multiple Input Multiple Output*), que são técnicas de múltiplas antenas que tiram vantagem da diversidade espacial no canal de rádio.

O LTE *Release 9* fornece métodos de posicionamento mais precisos. Um desses métodos utiliza o princípio OTDOA (do inglês *Observed Time Difference Of Arrival*). Este método utiliza sinais de referência inseridos nas transmissões de *downlink* no LTE que são enviados por diferentes estações rádio-base para calcular com maior precisão o posicionamento dos UEs. Os UEs são dispositivos de acesso para o usuário e provêm medições que indicam as condições do canal para a rede.

O LTE *Release 10* ou LTE-Advanced deve suportar significantes aumentos instantâneos nos picos das taxas de dados (100 Mb/s para alta mobilidade e 1 Gb/s para baixa mobilidade foram estabelecidos como alvos para pesquisa) [8]. Quanto à mobilidade, o sistema deve ser capaz de suportar mobilidade em velocidades de até 350 km/h (ou mais) dependendo da banda de frequência.

2.2.1 Arquitetura do LTE

O acesso à rede LTE é basicamente através de uma rede de estações rádio-base, chamadas de eNBs (do inglês *evolved NodeB*), que geram uma arquitetura plana como a da Figura 2.2. A eNB hospeda a camada PHY, a camada MAC, o RLC (do inglês *Radio Link Control*), e camadas PDCP (do inglês *Packet Data Convergence Protocol*). Controla o plano de usuário, a compressão do cabeçalho e a cifragem de dados.

A interface aérea do LTE conhecida como E-UTRA combinada com UEs e eNBs é chamada de E-UTRAN (do inglês *Evolved UMTS Terrestrial Radio Access Network*). Complementar ao E-UTRAN no LTE está o SAE (do inglês *Architecture Evolution System*), que é basicamente um item de estudo dentro do 3GPP para definir o EPC (do inglês *Evolved Packet Core*), que será detalhado a seguir. Juntos, o EPC e o E-UTRAN formam o EPS, visto na Figura 2.3.

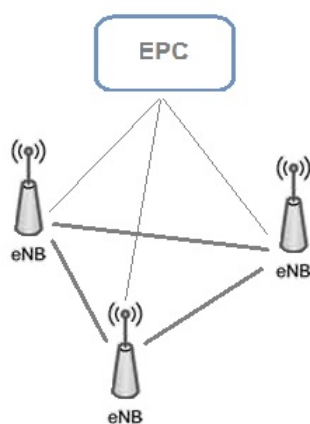


Figura 2.2: Arquitetura plana de eNBs (*evolved NodeB*) para acesso à rede LTE.

O EPC inclui o P-GW (do inglês *PDN Gateway*), o S-GW (do inglês *Serving Gateway*), o MME (do inglês, *Mobility Management Entity*), o HSS (do inglês *Home Subscriber Service*) e o PCRF (do inglês *Policy Control and Charging Rules Function*). O P-GW provê conectividade entre o usuário e as redes de pacotes de dados externas (PDNs do inglês *Packet Data Networks*), sendo o ponto de saída e entrada para o tráfego dos UEs, e é um elemento mediador entre redes 3GPP e não 3GPP. Um UE pode se conectar a outros P-GWs caso queira se conectar com outras redes externas.

O S-GW faz o roteamento e o repasse de pacotes e atua nos processos de *handover* inter-eNB ou no *handover* entre o acesso LTE e outras tecnologias 3GPP. O MME é como um nó de controle da rede LTE que acompanha o modo ocioso dos UEs e o processo de *paging* incluindo retransmissões. *Paging* é um processo de gerenciamento da memória, onde pode-se recuperar dados em um armazenamento secundário e utilizá-lo na memória principal. O MME também escolhe o S-GW para um UE em uma ligação inicial e no momento do *handover* intra-LTE, entre outras funções.

A informação sobre PDNs as quais cada UE pode se conectar está localizada no HSS. O PCRF

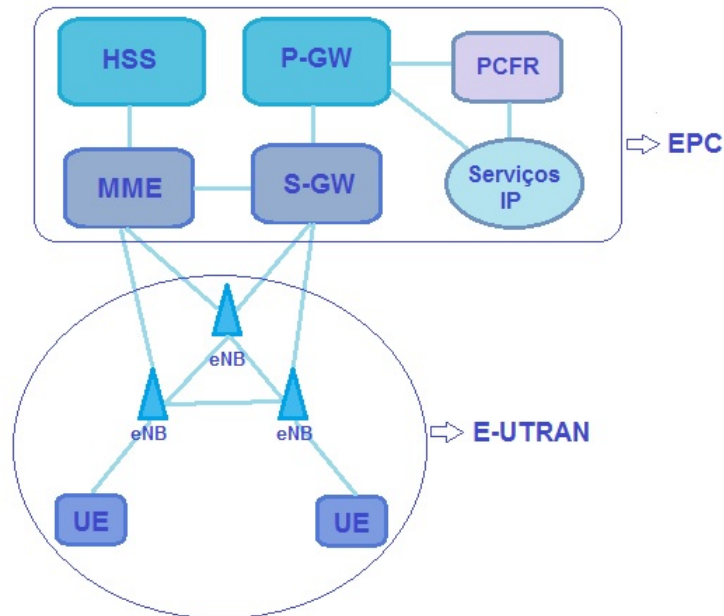


Figura 2.3: Estrutura do EPS (*Evolved Packet System*), formado pelo EPC e pelo E-UTRAN.

gerencia funcionalidade de carregamento e políticas de controle, assim como os níveis de QoS.

2.2.2 LTE-Advanced

As versões mais atuais do LTE são chamadas de LTE-Advanced, mais especificamente o *Release 10*, como dito anteriormente. O LTE-Advanced possui aprimoramentos em relação ao LTE *Release 8*, porém é compatível com o mesmo, o que significa que é possível implementar o LTE-Advanced em espectros já utilizados pelo LTE. Além disso, para implementação do LTE-Advanced são necessárias apenas algumas melhorias nos equipamentos que compõem a rede, o que gera um baixo custo de implementação. Uma das melhorias do LTE-Advanced em relação ao LTE é o OFDMA como o esquema de múltiplo acesso no *downlink*, que permite alta flexibilidade no número de usuários simultâneos e diferentes taxas de dados para diferentes usuários. Entretanto, para o *uplink* é utilizado o SC-FDMA, pois geralmente o *uplink* não requer altas taxas de dados. A seguir serão discutidas as tecnologias utilizadas no LTE-Advanced.

2.2.2.1 Agregação de Portadora

A agregação de portadora foi proposta para o LTE-Advanced a fim de prover uma maior largura de banda para os usuários, enquanto os equipamentos do LTE continuam vendo o espectro como componentes de portadoras separadas. A agregação de portadora é feita agrupando-se vários componentes de portadora do LTE. Existem três tipos de agregação de portadora: a agregação contígua na banda, a agregação não-contígua na mesma banda e a agregação não-contígua em múltiplas bandas. A primeira se aplica melhor em situações onde o operador possui uma grande

banda disponível, a segunda em situações onde há uma extensão da banda não disponível para o LTE-Advanced e a terceira é a mais atrativa dos tipos de agregação, pois permite combinar componentes de portadora distribuídas em diferentes bandas. A Figura 2.4 ilustra os tipos de agregação de portadora especificados.

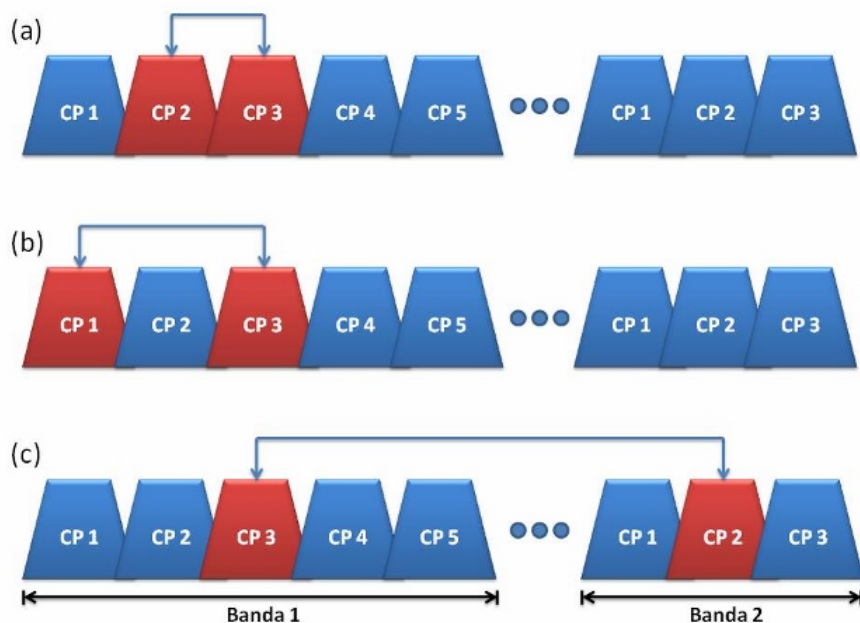


Figura 2.4: Na figura, CP é o componente de portadora, (a) Representa a agregação de portadora contígua na banda, (b) representa agregação não-contígua na mesma banda e (c) agregação não-contígua em múltiplas bandas.

2.2.2.2 Sistemas MIMO Avançados

De acordo com os requisitos do LTE-Advanced, a máxima eficiência espectral deve ser maior que 30 bps/Hz no *downlink*, que requer o uso da multiplexação espacial MIMO 8x8 [9]. Grande parte da tecnologia MIMO necessária já está implementada no LTE, porém são necessárias melhorias. Essas melhorias podem ser alcançadas com certa quantidade de informação sobre o estado do canal na estação base, que pode assim se adaptar às condições do canal de rádio. Essas informações são retiradas de sinais de referência projetados de tal maneira que minimize o excesso de informações coletadas e a degradação devido à estimação de erros do canal no UE.

2.2.2.3 Repasse no LTE-Advanced

O LTE-Advanced introduziu repasse no E-UTRAN a fim de aumentar a performance do LTE em termos de cobertura e vazão. Esta pode ser chamada de uma arquitetura *multihop*, pois utiliza recursos de redes *ad hoc* e celular. A arquitetura *multihop* do LTE-Advanced aumenta a cobertura e a capacidade, especialmente na borda das células, reduzindo a distância de transmissão entre

enlaces *multihop*, o que faz com que sejam alcançadas altas taxas de dados comparadas com os longos enlaces únicos [10].

A arquitetura básica do repasse no LTE-Advanced é um RN (do inglês *Relay Node*), o nó repassador, entre a eNB e o UE. O RN utiliza uma conexão sem fio para se conectar à célula doadora. A eNB da célula doadora é chamada eNB doadora (DeNB do inglês *Donor eNB*) e de acordo com o padrão desenvolvido, o UE se conecta com o RN através da interface Uu e a DeNB se conecta ao RN pela interface Un, este esquema está ilustrado na Figura 2.5.



Figura 2.5: Arquitetura básica do repasse no LTE-Advanced.

A comunicação entre o RN e a DeNB pode ocorrer de duas formas, na banda e fora da banda. A primeira é referente ao uso da mesma banda em Uu e Un, nesta o RN é do Tipo 1. A segunda utiliza bandas diferentes para Uu e Un e o RN é do Tipo 1a. No caso de uma comunicação na banda há o risco de auto-interferência no nó repassador quando estiver recebendo dados em Uu e transmitindo dados em Un ao mesmo tempo, o que pode ser evitado através de compartilhamento de tempo entre Uu e Un. O RN suporta grande parte das funcionalidades do eNB - entretanto o DeNB será responsável pela seleção de MME (do inglês *Mobility Management Entity*).

2.3 O Padrão IEEE 802.11

Um grande número de equipamentos hoje é baseado no padrão IEEE 802.11. Esse é o padrão internacional dominante para sistemas comerciais de comunicação sem fio, definindo as operações das redes locais conhecidas como WiFi. O IEEE 802.11 opera originalmente na faixa de frequência ISM (do inglês *Industrial, Scientific and Medical*) nas bandas de 2,4, 3,6, 5 e 60 GHz. O padrão IEEE 802.11 é um membro da família IEEE 802, que é uma série de especificações para tecnologias de redes locais LAN (do inglês *Local Area Network*). As especificações individuais da série 802 são identificadas por um segundo número, por exemplo, 802.3 é a especificação para uma rede CSMA/CD (do inglês *Carrier Sense Multiple Access With Collision Detection*) e 802.5 para a especificação *Token Ring* [11]. As especificações IEEE 802.11 são focadas nas duas camadas mais baixas do modelo OSI porque elas incorporam tanto componentes físicos quanto componentes de enlace. A camada MAC (do inglês *Medium Access Control*) estabelece um conjunto de regras para determinar como acessar o meio e enviar dados. Já a camada PHY (do inglês *Physical Layer*) estabelece os detalhes de transmissão e recepção. A base da especificação 802.11 inclui o 802.11 MAC e três camadas físicas: FHSS (do inglês *Frequency-Hopping Spread-Spectrum*), DSSS (do inglês *Direct-Sequence Spread-Spectrum*) e IR (do inglês *Infrared*).

O IEEE publicou em 1997 o padrão original 802.11 com uma especificação para a transmissão e protocolo de controle de acesso ao meio para WLANs. Anos depois versões revisadas surgiram e foram incluídos subpadrões como o 802.11a, 802.11b, 802.11d, 802.11e, 802.11f, 802.11g, etc. O padrão 802.11a descreve uma camada física baseada em multiplexação por divisão de frequência ortogonal (OFDM, do inglês *Orthogonal Frequency Division Multiplexing*), utiliza a faixa de frequência de 5,8 GHz e pode oferecer uma velocidade de 54 Mb/s. O padrão 802.11b alcança uma taxa de transmissão de 11 Mb/s e opera na frequência de 2,4 GHz, sendo amplamente usado por provedores de redes sem fio. O 802.11e agrega qualidade de serviço (QoS do inglês *Quality of Service*) às redes IEEE 802.11, já o 802.11f recomenda a prática para os equipamentos WLAN de tal forma que os APs (do inglês *Access Point*) possam interoperar. O 802.11g baseia-se na compatibilidade com os dispositivos 802.11b e oferece uma velocidade de 54 Mb/s.

2.3.1 Arquitetura

A arquitetura do IEEE 802.11 é formada fundamentalmente pelo conjunto de serviços básicos BSS (do inglês *Basic Service Set*). Um BSS é definido como um conjunto de estações que estão sob o controle de uma única função de coordenação distribuída, o DCF (do inglês *Distributed Coordination Function*). A área básica de serviço, BSA (do inglês *Basic Area Service*) é a área geográfica de cobertura do BSS, semelhante à célula em uma rede de comunicação celular [12]. Em um BSS todas as estações podem se comunicar diretamente, mas devido às degradações do sinal causadas por perda de propagação e interferências de BSS próximas, pode ocorrer o fenômeno conhecido como “terminal oculto”, explicado mais adiante.

O IEEE 802.11 pode funcionar em modo *ad hoc* e em modo em infraestrutura. No modo *ad hoc* um grupo de estações em um único BSS podem se interconectar sem necessidade de uma infraestrutura pré-definida. Ou seja, cada estação pode estabelecer uma comunicação direta com outra estação no BSS, sem canalizar todo o tráfego através de um ponto de acesso (AP) central, como pode ser visto na Figura 2.6, em que os dispositivos se conectam diretamente uns com os outros. Assim, não há um nó central para qual convergem todas as informações vindas de outro nó, sem necessidade portanto de um AP central. Dessa forma, cada conexão entre os nós é independente de outras ligações existentes na rede. Essa arquitetura será melhor discutida na seção 2.4.

Já o modo em infraestrutura utiliza APs para prover a integração necessária entre os nós e a conectividade entre múltiplos BSSs. Assim, todo o tráfego é canalizado através de um AP centralizado, como pode ser visto na Figura 2.7, em que o nó central é o AP por onde todo o tráfego passa para chegar até o nó de destino. Cada dispositivo é conectado ao AP para poder se comunicar com os outros dispositivos. Vários BSSs podem ser integrados usando um sistema de distribuição comum DS (do inglês *Distribution System*), constituindo assim um conjunto de serviços estendidos, ESS (do inglês *Extended Service Set*). Os APs de diferentes BSSs podem então se conectar, provendo conexão entre os BSSs. Quando o dispositivo muda de BSS, seu adaptador de rede pode mudar sua conexão para outro ponto de acesso do BSS em que ele se encontra.

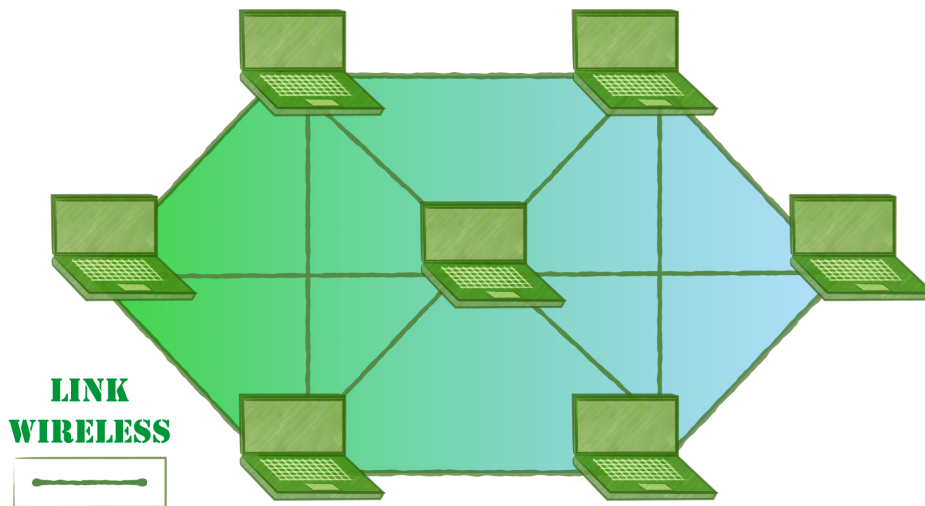


Figura 2.6: Rede WiFi em modo *ad hoc*. Os dispositivos se comunicam diretamente uns com os outros, sem necessidade de um AP central.

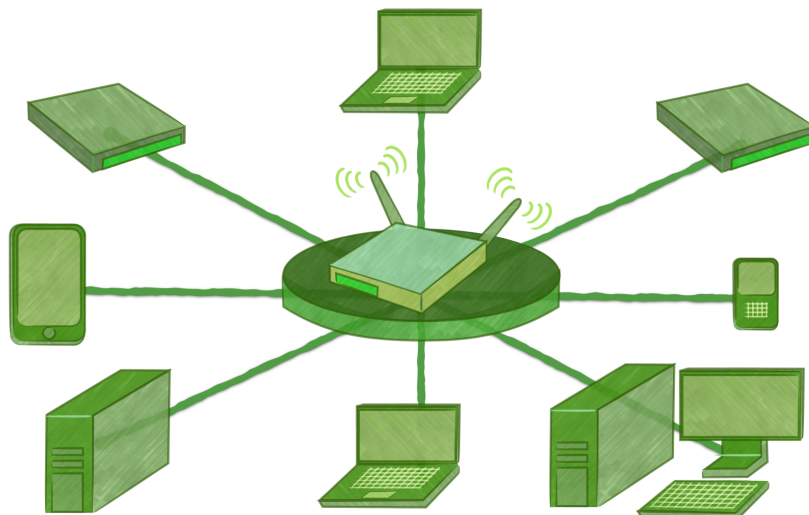


Figura 2.7: Rede em modo infraestrutura. O nó central é o AP, por onde todo tráfego é canalizado. Os dispositivos estão conectados ao AP e se comunicam uns com os outros através dele.

2.3.2 Camada Física

O padrão IEEE 802.11 é originalmente definido para três tipos de camadas físicas (PHY), espalhamento espectral por sequência direta (DSSS), espalhamento espectral por salto em frequência (FHSS) e infravermelho (IR) [13]. Com o DSSS todas as estações operam na mesma frequência central de um dado canal. O DSSS usa a banda de frequência 2,4 GHz ISM, com taxa básica de 1 Mb/s proporcionada pela modulação por chaveamento por deslocamento binário diferencial de fase, DBPSK (do inglês *Differential Binary Phase Shift Keying*) e 2 Mb/s proporcionada pela modulação por chaveamento por deslocamento diferencial em quadratura de fase, DQPSK (do inglês

Differential Quadrature Phase Shift Keying). A banda disponível é dividida em 11 subcanais, cada um com 22 MHz de largura [12].

Usando o FHSS, grupos de estações de comunicação operam em um canal de frequência comum por pequenos intervalos de tempo, depois selecionam outro canal de frequência para continuar a comunicação. Todas as estações do grupo ficam sabendo sobre qual canal de frequência será usado, seguindo uma lista de frequências pseudo-aleatória. O FHSS utiliza a banda de 2,4 GHz ISM, sendo o primeiro canal na frequência de 2,402 GHz, e todos os outros canais subsequentes são espaçados em 1 MHz de separação. Assim, vários BSSs podem coexistir na mesma área geográfica, sendo importante para aliviar a escassez de frequência e maximizar a vazão total em um único BSS. Além disso, as listas de frequência reduzem a probabilidade das estações operarem com a mesma frequência central [12].

O padrão IEEE 802.11 também especifica o infravermelho como camada física. A especificação IR utiliza a faixa de comprimento de onda a partir de 850-950 nm e taxa básica de acesso de 1 Mb/s, utilizando modulação 16 - PPM (do inglês *Pulse Position Modulation*). O IR não obteve sucesso comercialmente, apesar de ser atrativo para aplicações residenciais no futuro.

2.3.3 Sub-Camada de Controle de Acesso ao Meio (MAC)

Em uma rede IEEE 802.11, uma estação que está associada a um AP pode começar a enviar e receber quadros de dados de um AP. Mas pode acontecer de várias estações quererem transmitir quadros de dados ao mesmo tempo sobre o mesmo canal. Assim, é necessário um protocolo de controle de acesso para coordenar as múltiplas transmissões.

A camada MAC é responsável pela alocação de canais, pelo endereçamento de unidade de dados de protocolo (PDU do inglês *Protocol Data Unit*), pela formatação dos quadros, verificação de erros, fragmentação e remontagem. Assim, os protocolos da camada de controle de acesso ao meio são fundamentais para gerenciar a utilização da capacidade do enlace. Esses protocolos irão gerenciar a forma como as estações podem acessar o meio, prevenindo a colisão.

O meio de transmissão pode operar no modo de contenção, em que todas as estações disputam o acesso ao canal para transmissão de um pacote de dados, ou pode alternar entre o período de contenção, CP (do inglês *Contention Period*) e o período livre de contenção, CFP (do inglês *Contention-Free Period*). Durante o CFP, o AP controla o uso do canal, sem que as estações precisem disputar pelo acesso ao canal. A camada MAC do IEEE 802.11 utiliza o método baseado em serviço de contenção, assim cada estação deve disputar o acesso ao canal para cada quadro que desejar enviar.

A DCF, função de coordenação distribuída, é um método de acesso fundamental utilizado para a transferência de dados assíncrona na base do melhor esforço, em que todos os usuários com dados para transmitir têm uma chance igualmente justa para acessar a rede [12]. A função DCF deve ser suportada em todas as estações, e é baseada no protocolo de múltiplo acesso por detecção de portadora com prevenção de colisão, CSMA/CA (do inglês *Carrier Sense Multiple Access with Collision Avoidance*), que é o protocolo utilizado pelas LANs sem fio 802.11. Com esse protocolo,

cada estação sonda o canal antes de transmitir e assim não transmite quando detecta que o canal está ocupado. O acesso aleatório por detecção de portadora do 802.11 usa técnicas de prevenção de colisão, diferentemente da Ethernet, que usa técnicas de detecção de colisão, com o protocolo CSMA/CD (do inglês *Carrier Sense Multiple Access with Collision Detection*).

O protocolo CSMA/CA utiliza uma técnica de prevenção de colisão para reduzir a probabilidade de colisões. A prevenção de colisão é feita através do procedimento de retardo aleatório (*backoff*), em que a estação primeiro escuta o canal antes de transmitir um quadro, e caso o canal esteja ocupado, a estação precisa esperar por um período de *backoff* até que o canal fique ocioso para que ela tente transmitir novamente no fim do período de *backoff*. Assim, o procedimento de transmissão e recepção com o CSMA/CA é feito da seguinte forma [14]:

1. Quando a estação percebe que o canal está ocioso, ela transmite o quadro após um curto período de tempo denominado Espaçamento Interquadros Distribuído, o DIFS (do inglês *Distributed Inter-Frame Space*).
2. Caso a estação perceba que o canal está ocupado, ela escolhe um valor aleatório de recuo (*backoff*) e faz uma contagem regressiva a partir desse valor quando percebe que o canal está ocioso. Mas se a estação perceber que o canal está ocupado, o valor do contador é congelado.
3. Quando o contador chegar ao zero, a estação transmite então o quadro inteiro e fica esperando por um quadro de reconhecimento ACK (do inglês *Acknowledgement*).
4. Quando a estação fonte recebe um ACK, sabe que o quadro foi recebido corretamente na estação de destino. Se ela tiver outro quadro para transmitir, volta para a etapa 2. Se ela não receber um reconhecimento, ela volta novamente para a etapa de *backoff* e escolhe um valor aleatório dentro de um intervalo maior.

Quando a estação de destino recebe um quadro de dados, ela espera um curto período de tempo, conhecido como Espaçamento Curto Interquadros, o SIFS (do inglês *Short Inter-Frame Spacing*) e envia um quadro de reconhecimento para a estação fonte. Caso a estação fonte não receba um reconhecimento dentro de um dado período de tempo, ela admite que ocorreu um erro e retransmite o quadro, usando o protocolo CSMA/CA para acessar o canal. Se um número fixo de retransmissões já tiverem sido feitas e a estação fonte ainda não tiver recebido um reconhecimento, ela descarta o quadro.

2.3.3.1 Terminais Ocultos: RTS e CTS

O problema do terminal oculto ocorre quando duas estações estão dentro da faixa de cobertura de um AP e ambas se associam a ele, mas cada uma das estações está oculta a outra. Isso ocorre quando as estações estão fora de seus respectivos raios de detecção, mesmo estando associadas a um único AP. Assim, elas não são capazes de detectar uma transmissão já em andamento, causando transmissões simultâneas e aumentando o risco de colisão.

Outro exemplo de ocorrência do problema de terminal oculto pode ser visto na Figura 2.8, em uma rede *ad hoc*. Vamos considerar que temos 3 estações A, B e C. Suponha que, quando a estação

A está transmitindo um quadro, a estação C quer enviar um quadro para B também. A estação A está oculta para C, por isso C não escuta a transmissão de A. Assim C espera um intervalo DIFS, percebe o canal livre e então transmite o quadro, provocando colisão com o quadro de A. Isso resulta então em desperdício do canal durante o período de retransmissão de A e de C.

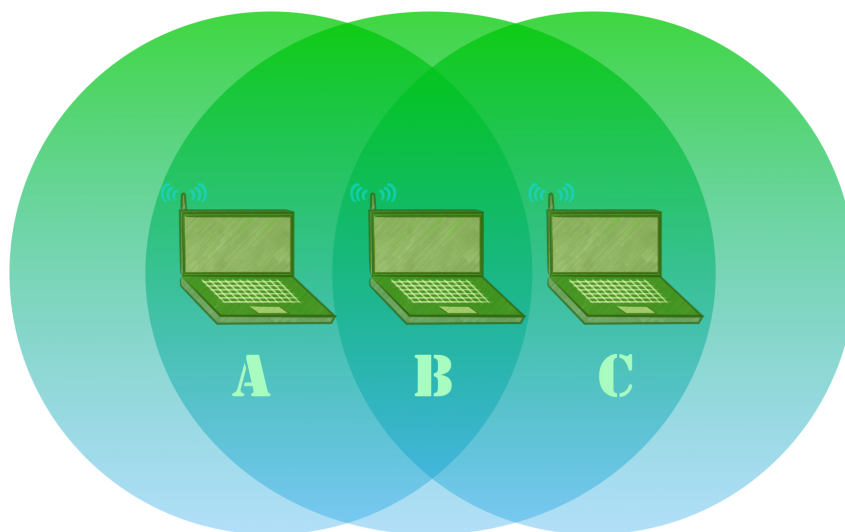


Figura 2.8: Problema do terminal oculto: o terminal A está fora da faixa de cobertura do terminal C. O terminal C não pode então detectar uma transmissão de A para B e pode tentar transmitir um quadro para B simultaneamente com a transmissão de A para B, gerando colisão.

O protocolo IEEE 802.11 propõe um esquema de reserva inteligente opcional a fim de evitar o problema do terminal oculto, permitindo o envio de quadros de controle para reservar acesso ao canal. Nessa solução, quando o terminal fonte quer enviar um quadro de dados ao terminal destino, ele envia primeiro um quadro de controle RTS (do inglês *Request to Send*) ao terminal de destino para solicitação de envio. Quando um terminal destino recebe o quadro RTS, ele responde transmitindo em *broadcast* um quadro CTS (do inglês *Clear to Send*). Esse quadro dá ao terminal fonte permissão explícita para enviar um quadro de dados, instruindo os outros terminais a não enviarem durante o tempo reservado, como mostrado na Figura 2.9. O quadro RTS indica o tempo total requerido para transmitir o quadro de dados e o quadro de reconhecimento ACK. Com essa informação, as estações atualizam o seu vetor de alocação do canal, NAV (do inglês *Network Allocator Vector*), onde é registrado o tempo necessário para a atual transmissão ser completa, ou seja, o tempo necessário para que o canal permaneça ocioso novamente. Quando as estações recebem o CTS, elas lêem o campo de duração e atualizam novamente os seus NAVs.

Dessa forma, todos os outros terminais dentro da faixa de alcance do terminal de destino ouvem o quadro CTS, deixando de transmitir durante o tempo especificado no quadro CTS. Então um quadro de dados só é transmitido por um terminal após o canal ter sido reservado. Assim, quando terminar o período NAV, os terminais esperam um tempo DIFS e iniciam a disputa pelo canal. Se os quadros RTS e CTS forem corretamente transmitidos, os quadros de dados e de ACK subsequentes são transmitidos sem colisões. Mas se houver colisão nos quadros de RTS/CTS, a colisão terá curta duração, já que esses quadros são de curta duração.

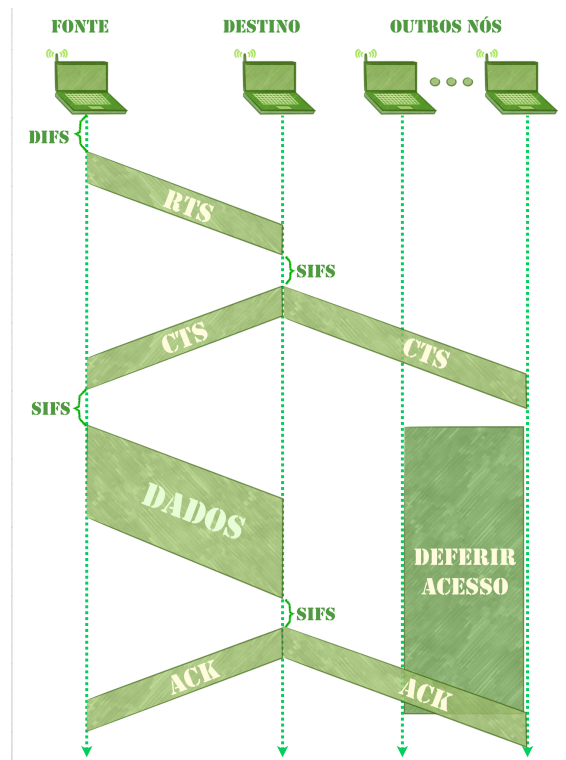


Figura 2.9: Envio de RTS e CTS: o terminal fonte espera um tempo DIFS e envia um quadro RTS ao terminal de destino. O terminal de destino recebe o quadro RTS e então envia um quadro CTS para o terminal fonte após um período SIFS. O terminal fonte recebe o CTS e então envia o quadro de dados após um período SIFS. O nó de destino recebe o quadro de dados e envia um ACK para o nó fonte após um período SIFS.

Como visto, a troca de quadros RTS/CTS é um bom procedimento para reduzir as colisões, principalmente em redes *ad hoc*, em que o uso desses quadros é essencial quando os nós não se encontram em alcance um dos outros. Mas o mecanismo de envio de RTS/CTS também pode gerar atraso e consumir recursos do canal. Por isso, o uso de RTS/CTS no 802.11 é opcional e quando utilizado, é feito para transmissões de quadros de dados longos, permitindo que cada terminal estabeleça um tamanho mínimo do quadro para que a troca RTS/CTS seja feita. Assim, a sequência RTS/CTS só é utilizada para quadros de tamanho maior que o tamanho mínimo definido. Muitas vezes, esse tamanho mínimo é escolhido como um comprimento maior que o comprimento máximo do quadro, omitindo então a utilização da troca RTS/CTS.

2.4 Redes Móveis *Ad Hoc*

Nas Redes Móveis *Ad Hoc* (MANETs do inglês *Mobile Ad hoc Networks*) os nós móveis sem fio se auto-organizam dinamicamente e aleatoriamente em uma topologia de rede *ad hoc*. As redes *ad hoc* vem sendo utilizadas há aproximadamente 20 anos na área militar e atualmente têm sido desenvolvidas tecnologias, como o *Bluetooth* e o IEEE 802.11, para torná-la uma MANET comercial.

As MANETs tem como finalidade fazer com que os nós móveis não troquem dados somente entre si, mas também recebam serviços da Internet através de um nó *gateway* para Internet. Assim as atividades da Internet poderiam alcançar locais antes inalcançáveis por falta de infraestrutura.

Os nós em uma MANET são livres para se organizarem e se moverem aleatoriamente, não possuem uma infraestrutura de rede e nem uma administração centralizada. Em geral, rotas entre nós em uma rede *ad hoc* podem ter múltiplos saltos, ou seja, todos os nós são passíveis de se comunicar com qualquer outro nó dentro de seu alcance, porém se este não estiver dentro deste alcance, os nós intermediários podem repassar as mensagens. A maioria dos estudos sobre as MANETs são feitas em ferramentas de simulação de rede, como OPNET [15] e ns-3 [16]. Algumas das vantagens de se utilizar esses simuladores são que eles possuem modelos pré-definidos para a maior parte dos protocolos de comunicação, além de possuírem uma interface gráfica que ajuda na visualização da composição e do funcionamento da rede.

As MANETs possuem uma série de problemas a serem estudados e resolvidos. Entre eles se encontram os problemas herdados das redes sem fio, como interferência no canal e o problema do terminal oculto, e também os seus próprios problemas, como a dificuldade de gerenciamento da rede. Além desses, existem questões a serem resolvidas na camada de transporte em relação ao TCP (do inglês *Transmission Control Protocol*). Os principais desafios encontrados nas MANETs serão melhor discutidos a seguir.

2.4.1 Desafios em MANETs

As redes *ad hoc* possuem muitas vantagens, porém não são perfeitas, pois possuem os mesmos problemas encontrados em redes sem fio. Alguns desses problemas são: a interferência que pode ocorrer no canal e assimetria nas propriedades de propagação do canal e o fenômeno do terminal oculto. Além desses problemas herdados de redes sem fio, as MANETs também trazem seus próprios problemas. Por não possuírem uma infraestrutura e uma administração central, os nós podem agir independentemente, gerar dados, fazer o roteamento da rede, o que pode gerar algumas dificuldades de gerenciamento e detecção de erros na rede. Existe também a questão dos nós estarem em movimento, o que pode gerar uma variação imprevisível na topologia da rede. Desse modo, ocorrem mudanças na rota dos pacotes que estão sendo enviados, e estes podem ser perdidos.

Na camada de transporte existem muitas questões a serem estudadas e resolvidas para MANETs em relação ao TCP, que é um protocolo de transporte orientado à conexão, ou seja, oferece uma transmissão confiável fim-a-fim e foi aperfeiçoado ao longo dos anos para funcionar em redes infraestruturadas. Um primeiro desafio se encontra na mobilidade das MANETs, que pode causar falhas na construção de rotas e conseqüentemente perda de pacotes e atrasos. O TCP interpretará esses atrasos e perdas como congestionamento na rede e iniciará seu controle de congestionamento retransmitindo pacotes desnecessariamente enquanto ocorre a reconstrução da rota, além de ocasionar a degradação da vazão.

Um segundo problema está na interação dos nós na camada MAC já que, mesmo em um cenário em que os nós se encontram estáticos, uma rede *ad hoc* está limitada às interações entre os nós

vizinhos e pelas interferências causadas pelos terminais ocultos e expostos. Um terceiro desafio se encontra na interação entre as camadas MAC e o TCP. Em um ambiente *multihop*, no caso de diversos fluxos TCP simultâneos, podem ocorrer problemas de injustiça como a captura do canal por alguns poucos fluxos. Além disso, em um cenário com apenas uma conexão TCP, a vazão instantânea pode ser bastante instável. Tais fenômenos não ocorrem, ou ocorrem com menos frequência quando é utilizado o UDP (do inglês *User Datagram Protocol*), protocolo de transporte não-orientado à conexão.

2.4.2 Padrão IEEE 802.11 para MANETs

Atualmente existem dois padrões principais desenvolvidos para tornar a MANET uma tecnologia de rede comercializável: o IEEE 802.11 para WLANs e o *Bluetooth* para comunicações sem fio de curta distância. Devido a sua extrema simplicidade, o padrão IEEE 802.11 é uma boa plataforma para implementar uma rede *ad hoc* WLAN *single-hop*. Além disso, redes *multihop* que cubram áreas de vários metros quadrados podem potencialmente ser construídas explorando-se a tecnologia IEEE 802.11 [17]. O padrão 802.11 define dois modos de operação para WLANs: o baseado em infraestrutura e um sem infraestrutura ou *ad hoc*. No modo *ad hoc* uma estação pode se comunicar com qualquer outra, basta que estejam dentro do raio de alcance uma da outra. Além disso, a rede *ad hoc* pode se comunicar com a Internet se alguma das estações estiver ligada a uma rede cabeada.

Em um primeiro momento, os esforços da padronização do IEEE 802.11 foram voltados às soluções para WLANs baseadas em infraestrutura e não para redes *ad hoc*, porém o uso generalizado do IEEE 802.11 o torna uma solução interessante para estas. Vários estudos já foram feitos para investigar o desempenho do protocolo MAC do IEEE 802.11 em um ambiente *ad hoc* e a maior parte dos problemas mencionados são referentes à interação das características do canal sem fio (como os problemas dos terminal oculto e exposto), o protocolo MAC do IEEE 802.11 (principalmente o esquema de *backoff*) e os mecanismos do TCP (como o controle de congestionamento e o *time-out*).

2.4.3 Protocolos de Roteamento para MANETs

Devido às características das MANETs, como a auto-organização, a dinâmica e a volatilidade da rede, os protocolos para roteamento precisam ser adaptados à elas. Primeiramente é necessário um serviço de localização dentro da rede, para encontrar o nó destinatário, uma das maneiras para alcançar tal objetivo é inundar a rede com solicitações de localização dos nós. Mas, obviamente essa solução só é possível em redes pequenas.

Após localizar o nó destinatário é preciso um algoritmo de roteamento e encaminhamento para encontrar uma rota para a troca de dados através da MANET. E aí está o grande desafio, roteamento nas MANETs. Muitas pesquisas e estudos têm sido feitos nessa área para testar o desempenho dos algoritmos de roteamento em diferentes cenários propostos, com vários ambientes de rede e condições de tráfego.

Protocolos de roteamento para MANETs são tipicamente subdivididos em duas categorias:

protocolos de roteamento pró-ativos e protocolos de roteamento reativos sob demanda [17]. Os protocolos de roteamento proativos mantêm sempre uma rota entre dois pares de nós, eles fazem isso mantendo uma tabela de roteamento que é atualizada em intervalos de tempo fixos. Já os protocolos de roteamento reativos por demanda só estabelecem uma rota para o destinatário quando existe demanda. E a rota deixa de existir se o destinatário se tornar inacessível ou se esta não for mais utilizada. Um exemplo de protocolo proativo é o OLSR (do inglês *Optimized Link State Routing*), que será explicado com mais detalhes na próxima seção. E um exemplo de protocolo reativo é o AODV (do inglês *Ad hoc On Demand Distance Vector*).

2.4.4 MANETs e 4G

Mencionadas anteriormente na Seção 2.2, as redes 4G são conhecidas por serem redes de banda larga híbridas que integram diferentes topologias e plataformas [17]. Em um primeiro nível de integração, as redes 4G podem ser integradas a redes sem fio heterogêneas, como as WLANs (do inglês *Wireless Local Area Network*), WAN (do inglês *Wide Area Network*), PAN (do inglês *Personal Area Network*) e também às redes móveis *ad hoc*. Por serem totalmente baseadas em IP, as redes 4G têm muitas vantagens, pois o IP é compatível e independente da atual tecnologia de rádio. Assim o núcleo da rede 4G pode ser desenvolvida e aperfeiçoada independentemente das redes de acesso.

As MANETs podem se tornar parte importante das redes 4G, pois são redes que não precisam de uma infraestrutura ou de uma administração centralizada para funcionar. São redes que se formam dinamicamente através de um conjunto transiente de nós móveis. Os próprios equipamentos móveis dos usuários funcionam como os equipamentos necessários para o funcionamento da rede, como roteadores e *switches*.

2.5 Protocolo de Roteamento OLSR

Os protocolos de roteamento para redes *ad hoc* são baseados em algoritmos que fazem o mapeamento da topologia da rede e constroem tabelas de roteamento, que são construídas e atualizadas de diferentes formas, dependendo do método de roteamento de cada protocolo. Os algoritmos utilizados calculam a rota de um nó de origem para um nó de destino de diferentes formas, sendo que cada nó que recebe um pacote verifica o destino do pacote em seu cabeçalho IP e assim repassa o pacote para o próximo salto de acordo com a tabela de roteamento, a fim de que o pacote possa chegar em seu nó de destino.

Com o desenvolvimento de tecnologias para redes sem fio móveis, surge a demanda por protocolos de roteamento adaptados a ambientes com alta mobilidade dos nós e com largura de banda limitada. Protocolos de roteamento para redes sem fio móveis são bem mais complexos que para redes com nós estáticos. O protocolo de roteamento para essas redes precisa ser capaz de se adaptar rapidamente às rupturas de enlaces e à mudança na topologia causada pelo movimento dos nós.

Uma rede *ad hoc* pode se expandir rapidamente em suas dimensões, trazendo a possibilidade de

laços nas rotas e falta de informação sobre alguns nós da rede. Ainda existe também a possibilidade dos enlaces serem uni-direcionais e das rotas serem perdidas, ou de nós entrarem e saírem da rede rapidamente.

Para o roteamento em redes *ad hoc* móveis existem algumas categorias de protocolos:

- Pró-ativos: Os protocolos pró-ativos são baseados na troca periódica de mensagens de controle. São enviadas mensagens para os nós vizinhos para que a vizinhança local seja conhecida e também são enviadas mensagens para toda a rede para que se possa conhecer todos os nós da rede e sua topologia. Exemplos desse tipo de protocolo são OLSR (do inglês *Optimized Link State Routing*) e DSDV (do inglês *Destination-Sequenced Distance-Vector*).
- Reativos: Os protocolos reativos não tomam nenhuma iniciativa para encontrar uma rota para o destino até que isso seja requerido. Esses protocolos fazem uma consulta na rede através de inundação, tentando descobrir as rotas de acordo com a demanda. Assim, há uma redução da sobrecarga de tráfego com pacotes de controle, mas há um aumento da latência no descobrimento de rota para o destino. Exemplos desse tipo de protocolo são o AODV e o DSR (do inglês *Dynamic Source Routing*).
- Híbridos: Combinam características de protocolos pró-ativos e reativos para utilizar vantagens de ambos.
- Baseados na localização: Estabelece as rotas através das posições físicas dos nós, ou seja, suas coordenadas geográficas.
- Baseados na energia: Faz o roteamento levando em consideração a energia contida em cada nó.

O protocolo OLSR é um protocolo pró-ativo para redes móveis *ad hoc*. Por ser um protocolo pró-ativo, tem a vantagem de ter rotas imediatamente disponíveis quando necessário. O OLSR é uma otimização de um protocolo de estado de enlace puro, nesse tipo de protocolo, todos os enlaces são declarados através da inundação de informação na rede. Já o OLSR declara apenas um subconjunto de ligações com seus vizinhos, além de reduzir o tamanho dos pacotes de controle. O OLSR ainda minimiza as inundações do tráfego de controle, utilizando apenas nós selecionados para difundir as mensagens de controle na rede. Essa técnica pode então reduzir o número de retransmissão dos pacotes de inundação [18].

O OLSR é adequado para redes grandes e densas e foi projetado para trabalhar de forma distribuída e não depende de uma entidade central. O protocolo também não requer transmissão confiável para as mensagens de controle, envia suas mensagens de controle periodicamente, tolerando eventuais perdas de pacotes. Além disso, quando os pacotes não são entregues na ordem certa, os pacotes são reordenados de acordo com seu número de sequência, conseguindo interpretar então quais são os pacotes mais recentes.

2.5.1 *MultiPoint Relays* - MPRs

Uma das principais vantagens do OLSR é que ele é capaz de selecionar apenas alguns nós, conhecidos como MPRs (do inglês *MultiPoints Relays*) para retransmitir os pacotes *broadcast*. Assim, um nó N envia os pacotes para os seus vizinhos, mas os vizinhos que não são MPRs daquele nó só lêem e processam o pacote recebido, mas não retransmitem aquele pacote. Assim, só os nós que são MPRs do nó N retransmitem os pacotes. Esse mecanismo minimiza o inundamento da rede com pacotes *broadcast*, reduzindo retransmissões duplicadas na mesma região [18].

Os nós MPRs selecionados podem ser mudados ao longo do tempo. Essa informação é transmitida através de mensagens de HELLO. A técnica de MPR evita que cada nó receba o mesmo pacote diversas vezes de diferentes vizinhos, para que a rede não fique sobrecarregada de informações redundantes. Essa técnica então se mostra eficiente para limitar a quantidade de nós na rede que encaminha os estados de enlace através de pacotes *broadcast*, eliminando assim as mensagens redundantes.

Os nós MPRs são selecionados entre os vizinhos a um salto do nó, de forma que cada nó vizinho a dois saltos tenha um enlace bidirecional com esse nó, através dos nós vizinhos de apenas um salto, como pode ser visto na Figura 2.10. Assim, quanto menor for o número de nós MPRs selecionados, melhor é o protocolo de roteamento. As rotas para os destinos conhecidos são calculadas através dos nós selecionados como MPRs, ou seja, esses nós são os intermediários em cada rota. Assim a rota é uma sequência de saltos através dos nós MPRs da fonte até o destino.

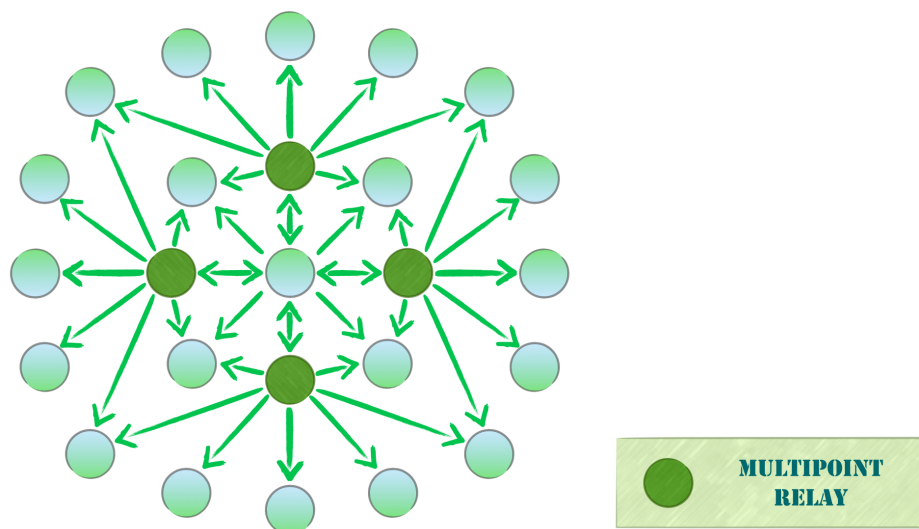


Figura 2.10: Seleção de MPRs: o nó fonte seleciona apenas alguns nós vizinhos para a retransmissão dos pacotes de controle para nós a dois saltos do nó fonte.

2.5.2 Sensoriamento dos Nós Vizinhos

Para que um enlace seja considerado válido, ele deve ser bidirecional. Então cada enlace deve ser verificado nas duas direções para ser validado. Para que isso seja possível, o OLSR

utiliza mensagens *broadcast* HELLO, em que cada nó envia periodicamente mensagens contendo a informação sobre seus vizinhos e seu estado de enlace. Essas são mensagens de controle enviadas em modo *broadcast* e são recebidas por todos os nós vizinho de um salto, mas só são retransmitidas por nós MPRs.

As mensagens de HELLO contêm a lista de endereços dos vizinhos para quais existem enlaces bidirecionais e a lista dos endereços dos nós vizinhos que são ouvidos por esse nó. As mensagens de HELLO permitem que cada nó tenha conhecimento sobre seus vizinhos de até dois saltos. Baseado nessa informação cada nó pode selecionar seus MPRs e o status do nó como MPR é indicado nas mensagens de HELLO.

Cada nó armazena em uma tabela a informação sobre seus vizinhos de um nó, o estado de enlace com esses nós e uma lista dos seus vizinhos de dois saltos. O estado de enlace com um nó vizinho pode ser unidirecional, bidirecional ou MPR. Se for MPR, necessariamente esse enlace será bidirecional. Essa tabela também contém um número de sequência que especifica o mais recente MPR selecionado, sendo incrementado cada vez que os MPRs selecionados são atualizados.

2.5.3 Seleção e Declaração de MPR

O processo de seleção de MPRs é feito de forma independente em cada nó da rede. Cada conjunto de nós MPRs selecionados contém um subconjunto de nós vizinhos de uma salto e abrange todos os nós vizinhos de dois saltos. Os conjuntos de MPRs não deve ser muito grande, para que não se perca os benefícios de selecionar apenas determinados nós para retransmitir os pacotes de controle.

O MPRs são declarados nas mensagens de HELLO de cada nó, pois podem ocorrer mudanças nos MPRs selecionados por cada nó. Os MPRs são recalculados quando ocorrem mudanças na vizinhança, quando por exemplo um enlace bidirecional for falho ou outro enlace bidirecional for adicionado. Assim, cada nó constrói sua tabela de seleção de MPR de acordo com as informações recebidas nas mensagens de HELLO e adiciona na tabela os endereços dos seus vizinhos de um salto que são selecionados como MPRs. Essa tabela fica sendo atualizada com a chegada de mensagens de HELLO e seu número de sequência é incrementado em caso de modificação.

Para fazer a declaração dos nós MPRs, mensagens de controle de topologia são enviadas periodicamente por cada nó, chamadas de mensagens TC (do inglês *Topology Control*). As mensagens TC são enviadas por *broadcast* para declarar a lista de nós que selecionaram o nó remetente como MPR. Cada mensagem TC contém a lista de vizinhos que selecionou o nó remetente como MPR e também contém o número de sequência associado a esse conjunto.

As informações contidas nas mensagens TC ajudam cada nó a construir sua tabela de topologia. Essas mensagens são enviadas em um intervalo de tempo que depende se o conjunto de nós seletores do nó remetente é alterado. Assim quando uma alteração ocorre nesse conjunto, a próxima mensagem TC pode ser enviada antes do horário agendado. Um intervalo mínimo é determinado entre o envio de uma mensagem TC e outra, então quando esse tempo já tiver decorrido, a próxima mensagem TC pode ser enviada imediatamente. Com base nas informações obtidas nas

mensagens TC, a tabela de roteamento é calculada.

2.5.4 Cálculo da Tabela de Roteamento

Para que o roteamento ocorra, cada nó mantém uma tabela de roteamento que permite que os pacotes sejam roteados para outros destinos na rede. Depois que os nós recebem uma mensagem TC, são armazenados na forma de pares como [último salto, nó], em que esse nó do par é um dos endereços contidos na mensagem TC. Dessa forma, para encontrar um caminho para um nó remoto R, é preciso encontrar um par [X, R], depois um par [Y, X] e assim por diante, até que seja encontrado um nó Y no conjunto de vizinhos do nó de origem.

A tabela de roteamento tem como entrada: o endereço de destino, o próximo endereço (próximo salto), o número de saltos até o destino e o endereço da interface local. Essas informações são obtidas do conjunto topológico, através das mensagens TC, e da base de informações do estado de enlace, através das mensagens HELLO. As entradas da tabela são registradas para cada destino na rede para qual a rota é conhecida. Assim, todos os destinos para os quais a rota estiver quebrada ou só parcialmente conhecida não são inseridas nessa tabela. Quando acontece alguma mudança nessas bases de informações, a tabela de roteamento é recalculada. A tabela de roteamento pode ser alterada quando *links* com vizinhos aparecem ou desaparecem da rede, vizinhos de dois saltos são incluídos ou removidos e quando a informação sobre as interfaces associadas são mudadas.

2.6 O Simulador ns-3

O ns-3 é um simulador de redes a eventos discretos, desenvolvido especialmente para pesquisa e uso educacional. O projeto ns-3 iniciou em 2006 e várias versões já foram lançadas com novos modelos, correção de erros e contribuições dos usuários. O ns-3 é escrito na linguagem C++ e possui código aberto, ou seja, é um ambiente aberto para pesquisadores que queiram contribuir e compartilhar software com o projeto. Em nossa simulação foi utilizada a versão 3.18 do ns-3, por ser a mais recente no período de realização do projeto.

O código fonte do ns-3 é organizado principalmente na pasta **src**. Existem componentes que são comuns para todos os protocolos, hardware e modelos de ambiente. Esses componentes estão no núcleo do simulador, que é implementado no **src/core**. Os pacotes, que são objetos fundamentais em uma simulação de rede, estão implementadas no **src/network**. O ns-3 é estruturado em módulos e para cada módulo existem diversas classes. As classes utilizadas nessa simulação são descritas ao longo desse capítulo. O ns-3 gera arquivos de rastreamento de pacotes (*packet trace*) no formato *pcap*, assim é possível realizar o rastreamento através da análise de pacotes em aplicativos como o Wireshark. A utilização do ns-3 exige dos usuários um conhecimento básico de C++ e conceitos de programação orientada a objetos.

Uma grande vantagem do ns-3 é que ele possui uma vasta documentação e outras formas de compartilhamento de informações e dúvidas em grupos dos usuários na Internet. Assim, o ns-3 tem se tornado uma ferramenta muito utilizada por pesquisadores em vários países e os resultados

obtidos com ele têm sido de grande relevância para os estudos em redes de comunicação, divulgados em muitos artigos e conferências.

2.7 Revisão Bibliográfica

O escoamento de dados tem sido muito estudado e vários trabalhos estão sendo feitos sobre isso. Em [19] é apresentado um estudo quantitativo da execução do escoamento de dados móveis 3G para redes WiFi. Foram feitas coletas estatísticas usando 100 iPhones em áreas metropolitanas, verificando sua conectividade WiFi. Segundo os estudos realizados, o WiFi já descarregava 65% do tráfego total de dados móveis e economizava 55% da bateria dos dispositivos. Segundo os autores, diversas soluções têm sido propostas para resolver o problema do crescimento explosivo de tráfego, como aumentar a estrutura para que existam mais células e estações base, podendo diminuir assim o tamanho das células. Porém, essa não seria uma solução fácil nem barata. Já o escoamento do tráfego móvel para redes WiFi tem sido visto como a solução mais viável do momento, pois instalar mais *hotspots* WiFi é significativamente mais barato que mudar a estrutura física da rede.

Pelo menos dois tipos de escoamento têm sido considerados na literatura: *on-the-spot* e *delayed*. No tipo *on-the-spot*, quando os usuários se movem pra fora da cobertura WiFi, o escoamento é interrompido e todas as transferências incompletas são completadas em redes celulares. No tipo *delayed*, considera-se que as transmissões são tolerantes a atraso, assim cada transmissão é associada a um prazo de entrega e os usuários podem entrar e sair da cobertura WiFi até que a transferência de dados seja concluída. Caso o prazo de entrega seja esgotado, as transferências passam a ser concluídas em redes celulares. Na análise feita em [19], obteve-se o resultado de redução de 65% do tráfego usando o escoamento *on-the-spot* para redes WiFi, ou seja, apenas 35% do tráfego precisaria ser transmitido por redes 3G, mostrando um ganho significativo. Esse ganho ainda poderia crescer rapidamente com mais incentivos para o escoamento tolerante a atraso.

Em [20] foi relatado um estudo de caso em que foi investigado o problema de selecionar nós alvos para realizar o escoamento de dados móveis para entrega de informações em Redes Sociais Móveis (MoSoNets). Nesse cenário, os prestadores de serviços de conteúdo entregavam as informações através de redes celulares apenas para os usuários selecionados. Assim foi explorada uma forma de comunicação oportunista para facilitar a disseminação de informações e assim reduzir a quantidade de tráfego celular. Nesse estudo foram selecionados um certo número de usuários para possíveis alvos do descarregamento de tráfego. Dessa forma os alvos escolhidos propagam as informações entre todos os usuários inscritos. É relatado nesse trabalho que grande parte do tráfego de dados móveis se deve ao uso de serviços de redes sociais *online*, incentivado pela proliferação de *smartphones* e sistemas operacionais móveis. Nesse trabalho foi estudado como escolher o nó inicial dentro de um número de possíveis nós alvo, usando três tipos de algoritmos, de natureza distintas: guloso, heurístico e aleatório. Foi concluído que o algoritmo guloso apresentou melhores resultados, seguido pelo algoritmo heurístico.

Balasubramanian *et al.* [21] investigaram em seu trabalho se o acesso WiFi pode ser usado para aumentar a capacidade 3G em ambientes móveis. Foi então proposto um sistema chamado *Wiffler*,

utilizando duas ideias principais: aproveitar a tolerância a atraso e a comutação rápida de pacotes para superar os problemas de desempenho e disponibilidade WiFi. Dessa forma, foi investigada a viabilidade de um método capaz de reduzir a pressão no espectro 3G utilizando a conectividade WiFi para transferência de dados sempre que possível. Para isso os assinantes 3G são incentivados a reduzir seu uso 3G mudando sua conexão para o WiFi de sua casa. Mas para o caso de redes veiculares, o uso de WiFi a partir de veículos em movimento é mais desafiador, pois os APs WiFi tem curto alcance e geralmente não são implantados para fornecer cobertura a estradas. Nesse artigo foi apresentado o sistema *Wiffler*, que aproveita o fato de que algumas aplicações como e-mail e transferência de dados permitem um pequeno atraso, e assim podem esperar que o WiFi esteja disponível para então transmitir. Esse sistema foi implementado em veículos e foi observado que para transferências de tamanho 5MB, que podem tolerar atrasos de até 60 segundos, o *Wiffler* reduziu o uso de 3G em 30%.

Luo *et al.* propuseram a arquitetura UCAN (do inglês *Unified Cellular and Ad Hoc Network*) em [22]. Na UCAN, um cliente móvel tem enlace para rede celular e enlaces ponto-a-ponto baseados no IEEE 802.11. Nesta proposta, a estação base 3G encaminha os pacotes para os clientes de destino via clientes proxy com melhor qualidade de canal. Em seguida, os clientes proxy usam uma rede *ad hoc* composta por outros clientes sem fio e enlace IEEE 802.11 para encaminhar os pacotes para os destinos apropriados, melhorando assim os rendimento da célula. Nessa arquitetura são propostos protocolos para descoberta de clientes proxy e para o roteamento *ad hoc*. É proposto também um mecanismo de crédito seguro para motivar os usuários a participarem da retransmissão de pacotes para os outros nós. Nesse trabalho são utilizadas as tecnologias HDR (do inglês *High Data Rate*), parte da família CDMA2000 de padrões 3G, e WiFi baseada no padrão IEEE 802.11.

Inspirando-se na arquitetura UCAN é possível projetar uma outra arquitetura que, ao invés de utilizar a tecnologia HDR, faz uso da tecnologia LTE, a qual é capaz de fornecer maiores taxas de dados e já tem sido amplamente desenvolvida e implementada em um muitos países. Portanto seria ainda mais interessante um trabalho que unisse a tecnologia LTE com o IEEE 802.11 de forma oportunista, escoando parte do tráfego LTE para clientes proxy repassarem os pacotes através de uma rede *ad hoc*. Essa é parte da proposta apresentada nesse trabalho e será melhor discutida na Seção 3.2.

Capítulo 3

Escoamento de Tráfego via Redes *Ad Hoc*

Neste capítulo é apresentado o cenário visualizado para escoamento de tráfego de dados de redes de telefonia móvel para redes *ad hoc* formadas pelos próprios dispositivos móveis dos clientes da operadora. Em particular, estudamos o escoamento de tráfego de uma rede de 4ª geração, representada pela tecnologia LTE, via uma rede *ad hoc* operada segundo a norma IEEE 802.11.

Além de apresentar o cenário em estudo, incluindo sua topologia e infraestrutura, apresentamos também os detalhes relativos à sua implementação no simulador a eventos discretos ns-3. Em particular, apresentamos as informações relativas aos diversos parâmetros utilizados e configurados para o estudo em questão.

3.1 O Cenário de Escoamento de Tráfego

Como visto nos capítulos anteriores, a demanda por maiores taxas de dados nas redes móveis cresce exponencialmente. Em vista desta situação, foi criado um cenário de simulação para escoamento de tráfego de redes 4G para redes WiFi. Avaliamos um equipamento de usuário (UE) que sempre irá tentar fazer o escoamento de dados, ou seja, passará automaticamente de uma conexão 4G para uma rede *ad hoc*.

Neste trabalho, os objetivos maiores estão focados no desempenho da rede *ad hoc* para permitir o escoamento de tráfego. A infraestrutura LTE e seus diversos parâmetros não são estudados. O maior desafio para esse trabalho é a integração entre a infraestrutura LTE e a rede *ad hoc*, através de um nó proxy que faz o escoamento do tráfego. Logo, esse nó precisa suportar uma interface para a infraestrutura LTE e outra interface para a rede *ad hoc*, sendo capaz de transmitir os dados originais de um nó remoto da infraestrutura LTE para nós de destino na rede *ad hoc*.

O módulo LTE, no simulador de rede ns-3 utilizado neste trabalho, permite criar modelos da infraestrutura do 3GPP E-UTRAN e redes LTE. Alguns recursos dos modelos LTE no ns-3 são: uma implementação básica dos equipamentos dos usuários (UEs) e eNBs, a administração dos

portadores de dados de rádio, das filas MAC e dos exemplos de RLC, suporte para o agendamento de pacotes no *downlink* e no *uplink*, um modelo de canal com o modelo de perda de canal E-UTRAN externo, entre outros.

Assim, o módulo LTE nos permite implementar os equipamentos básicos da rede, incluindo os modelos de propagação nas camadas MAC e PHY, e simular uma interface E-UTRAN onde uma eNB e vários UEs podem se comunicar entre eles utilizando um canal compartilhado. Foram utilizadas essas e outras ferramentas para montar a rede 4G que gera o tráfego a ser descarregado.

A rede WiFi é uma rede *ad hoc*. A rede *ad hoc* é composta por 25 nós espalhados em um terreno de dimensão 600×600 m. Com essa dimensão, os nós podem se encontrar distantes um do outro a mais de 150 m, dessa forma se faz necessário o repasse dos pacotes através de outros nós mais próximos. Esses nós podem permanecer estáticos ou se moverem no terreno com velocidade de 1 m/s e 3 m/s de acordo com o modelo de mobilidade *Random Direction 2d*. A velocidade de 1 m/s representa uma pessoa caminhando, já a velocidade de 3 m/s representa uma pessoa correndo rapidamente. Nesse trabalho variamos a velocidade dos nós a fim de obter dados para comparação da eficiência da rede de acordo com a mobilidade dos nós.

Também variamos o número de fluxos de um a cinco fluxos. Dessa forma, o nó fonte pode entregar dados para quantidades de destinos diferentes e é possível avaliar a distribuição do tráfego para diferentes números de usuários. Além disso é variada também a taxa de envio de dados da aplicação em 117.760 b/s, 174.640 b/s e 235.520 b/s. A menor taxa permite que a rede não fique sobrecarregada, evitando filas de pacotes. Já a maior taxa representa um rede com maior fluxo de dados, com geração de mais pacotes, deixando a rede sobrecarregada. E também acrescentamos uma taxa intermediária entre a menor e a maior, com o propósito de obter uma análise comparativa posteriormente. Essas taxas foram escolhidas após um grande número de simulações, para que fosse observado para quais taxas seria possível obter uma variação significativa nos resultados.

A fim de analisar o resultado que o desvanecimento em pequena escala pode causar no desempenho da rede, foram feitas simulação com diferentes valores para o parâmetro K de Rice. As simulações foram feitas para $K = \infty$, $K = 2$ e $K = 0$. $K = \infty$ representa um cenário em que o sinal recebido não sofre com os efeitos de múltiplos sinais chegando no receptor, ou seja, um canal sem o efeito do desvanecimento em pequena escala. Já para $K = 2$, múltiplos sinais chegam ao receptor, mas existe uma componente dominante, representando situações com linha de visada, em que não existem muitos obstáculos entre o transmissor e o receptor. Já $K = 0$ representa um cenário sem linha de visada, em que o canal sofre reflexão ou a difração do sinal em obstáculos, permitindo assim representar por exemplo, um espaço fechado com muitos obstáculos entre o transmissor e o receptor. A Tabela 3.1 mostra os valores dos parâmetros utilizados nas simulações.

A rede *ad hoc* montada funciona com o protocolo de roteamento OLSR. Esse protocolo foi utilizado por ser o protocolo que melhor se adaptou ao cenário simulado e por ser um protocolo mais bem elaborado dentro do ns-3. Há muito mais documentação sobre ele, o que nos dá mais suporte para trabalhar com esse.

Parâmetros das simulações		
Velocidade dos nós (m/s)	Taxa de envio de dados b/s	K de Rice
0	117.760	∞
1	176.640	2
3	235.520	0

Tabela 3.1: Parâmetros das simulações.

3.2 Cenário de Simulação

Em nossa simulação foi montada uma estrutura de uma topologia de rede LTE e de uma topologia de rede sem fio *ad hoc*. É gerado tráfego de dados a partir de um nó remoto para os nós de destino na rede *ad hoc*. O tráfego é recebido em um UE a partir da rede LTE e assim o tráfego é escoado para uma rede *ad hoc* a fim de chegar até o nó de destino. Dessa forma, os pacotes são roteados na rede *ad hoc* até chegarem no destinatário.

Consideramos apenas um nó da rede WiFi como gargalo para o escoamento de todo o tráfego proveniente da rede LTE. Assim, esse nó escolhido contém uma interface celular por onde recebe os dados provenientes da rede LTE e uma interface WiFi por onde encaminha os pacotes para a rede *ad hoc*. A topologia montada pode ser vista na Figura 3.1. Dessa forma, o protocolo de roteamento OLSR foi instalado em todos os nós WiFi e os pacotes recebidos no nó proxy são encaminhados para o nó do próximo salto de acordo com sua tabela de roteamento.

A topologia LTE é composta de um nó remoto, do P-GW, do eNB e do UE. No nó remoto foi instalada a aplicação para geração de tráfego, o P-GW provê a conectividade do UE com a rede externa, o eNB é a estação base e o UE representa o dispositivo de acesso do usuário, o nó proxy para o escoamento dos dados.

Para essa simulação foi implementado um cenário com 25 nós se movendo (ou não) em uma rede *ad hoc* limitada a um espaço de 600×600 m. Foram geradas 15 topologias de forma pseudo-aleatória para a posição inicial dos nós. As simulações foram feitas para três situações: para os nós em posição estática, para os nós se movendo em uma velocidade de 1m/s e para os nós se movendo em uma velocidade de 3m/s. O modelo de mobilidade utilizado foi o modelo implementado na classe `ns3::RandomDirection2dMobilityModel`, sendo configurado para as velocidades de 1m/s ou 3m/s com período de pausa determinado por uma variável aleatória exponencial de média 0,5 segundo.

3.2.1 LTE no ns-3

O ns-3 contém um módulo LTE que permite a criação de modelos de infraestrutura 3GPP E-UTRAN e redes LTE. Nesse modelo são implementados os equipamentos de usuários (UEs) e as estações base (eNBs). Também são implementadas as entidades RRC para UE e eNB, esquemas de modulação para *dowlink*, filas MAC, instâncias RLC, gestão do Indicador de Qualidade de

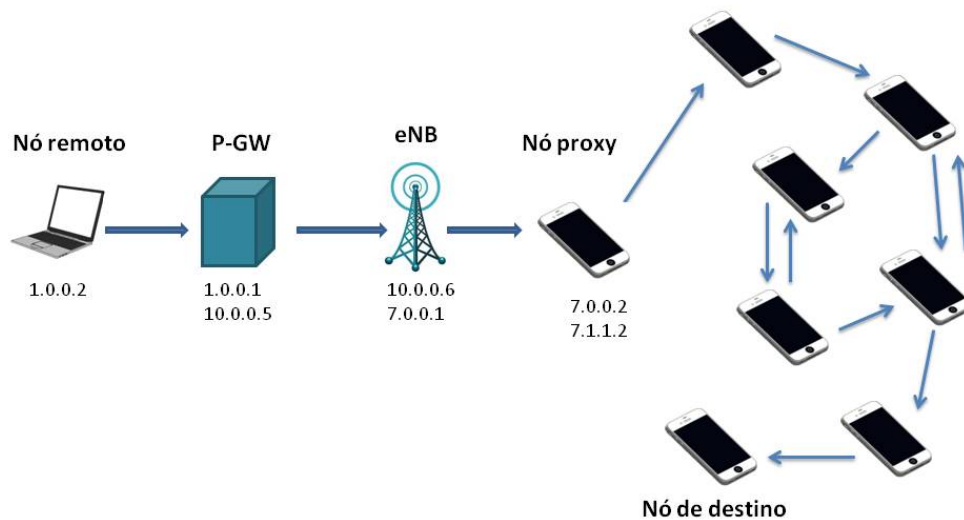


Figura 3.1: Topologia montada: a estrutura LTE contém o nó remoto, o P-GW, a eNB e o UE. O UE é o nó proxy que escoia o tráfego para a rede WiFi. A rede WiFi contém os nós se movendo e transferindo dados uns para os outros através de uma rede *ad hoc*.

Canal (CQI), agendamento de pacotes para *downlink* e *uplink*, entre outros recursos. Além disso, o módulo também inclui um modelo de camada PHY e de canal com perda de propagação E-UTRAN exterior.

O módulo LTE do ns-3 oferece uma implementação básica de dispositivos LTE, incluindo modelos de propagação PHY e camadas MAC. Ele permite simular uma interface E-UTRAN em que um eNB e vários UEs podem se comunicar entre si usando canal *downlink* ou *uplink* compartilhado. As classe `ns3::LteHelper` e `ns3::EpcHelper` foram as classes utilizadas para implementação do LTE nessa simulação.

3.2.2 Camada Física

Para a camada PHY da rede WiFi foram configurados os parâmetros indicados na Tabela 3.2:

Foi configurada a técnica de modulação por espalhamento de espectro por sequência direta para operar com taxa de 1 Mb/s para modo de dados e de controle. Como o padrão é o IEEE 802.11b, a modulação digital utilizada é o DBPSK, que já está inclusa no padrão 802.11b do ns-3.

Para o canal WiFi, foi utilizada a classe `ns3::ConstantSpeedPropagationDelayModel` e foi acrescentada perda de propagação de acordo com o Modelo de Dois Raios, implementado no ns-3 com a classe `ns3::TwoRayGroundPropagationLossModel`. O Modelo de Dois Raios foi adicionado

Parâmetros Camada PHY	
Potência de transmissão	10 dBm
Ganho de transmissão	0 dB
Ganho de recepção	0 dB
Limiar de detecção de energia	-73.8764 dBm
Limiar CCA	-80.9201 dBm
Ruído	7 dB

Tabela 3.2: Parâmetros de simulação para camada PHY da rede WiFi.

na simulação, já que raramente um caminho com uma única linha de visada entre dois nós móveis é o único meio de propagação do sinal. O Modelo de Dois Raios considera tanto o caminho direto quanto o caminho de reflexão no chão, e dá uma previsão mais precisa para longas distâncias. A equação para a potência recebida para uma distância d do transmissor é dada por [23]

$$P_r(d) = \left(\frac{h_t^2 h_r^2}{d^4 L} \right) P_t G_t G_r, \quad (3.1)$$

em que h_t é a altura do transmissor, h_r é a altura do receptor, P_t é a potência do transmissor, G_t é o ganho da antena transmissora e G_r é o ganho da antena receptora. Essa equação é mencionada em [23] e assume $L = 1$, que é o fator de perda do sistema. Mas o modelo do ns-3 adiciona o L para ser mais consistente com a equação de espaço livre.

O desvanecimento rápido (do inglês *fast fading*) é causado pela combinação dos múltiplos sinais que chegam no receptor devido a fenômenos como reflexão ou difração do sinal em obstáculos. Devido a esses fenômenos, ocorrem flutuações rápidas e aleatórias de amplitudes e fases de sinais em um curto período de tempo. Algumas distribuições como as distribuições de Rayleigh, Rice e Nakagami- m são utilizadas para descrever o comportamento estatístico do desvanecimento em pequena escala (*fast fading*), ou seja, a variabilidade do sinal em um pequeno número de comprimentos de onda.

A distribuição de Rice é utilizada quando existe uma componente do sinal dominante e é mais adequada para situações com linha de visada. O parâmetro K de Rice é a razão entre a potência do sinal em linha de visada e dos sinais advindos de múltiplos percursos. Nessa simulação é utilizada a distribuição de Rice com o parâmetro $K=0$ e $K=2$.

Quando $K=0$, a função densidade de probabilidade de Rice tende a se aproximar da função densidade de probabilidade de Rayleigh, assim, quando $K=0$, temos a distribuição de Rayleigh. Em nossa simulação, utilizamos a classe `ns3::NakagamiPropagationLossModel` para implementar o modelo de Rice, calculando o m (número de *clusters*) de Nakagami como

$$m = \frac{(K + 1)^2}{2K + 1}, \quad (3.2)$$

de acordo com Molisch [24]. Assim, quando $K=0$ temos uma distribuição de Rayleigh e quando $K=2$ temos uma distribuição de Rice.

Foi definido um limiar (dBm) para que a camada PHY detecte um sinal recebido através do atributo `EnergyDetectionThresold`. Assim, a energia de um sinal recebido deve ser superior a esse limiar para permitir que a camada PHY detecte o sinal. O atributo `CcaMode1Thresold` foi configurado para que a camada PHY, através do limiar de CCA (do inglês *Clear Channel Assignment*), declare o canal como ocupado quando a energia do sinal recebido for superior a esse limiar (dBm). Os valores foram escolhidos de forma que os nós não pudessem detectar um sinal muito baixo, recebido quando a distância entre dois nós diretos for maior do que 150 m. Os limiares escolhidos são mostrados na Tabela 3.1.

3.2.3 Camada de Enlace

Para os nós funcionarem em uma rede WiFi foi utilizado o padrão IEEE 802.11b nos nós. Para isso foi utilizado o padrão `WIFI PHY STANDARD 80211b` do ns3. Foi utilizado também o modelo do ns3 `YansWifiPhyHelper::DLT IEEE802 11 RADIO` para o link WiFi. Foi utilizada a classe `ns3::NqosWifiMacHelper` para criar camadas MAC sem QoS ativado para os dispositivos móveis da simulação. Essa classe pode criar MACs do tipo `ns3::ApWifiMac`, `ns3::StaWifiMac` e `ns3::AdhocWifiMac`.

Para implementar uma rede *ad hoc* nessa simulação, foi utilizada a classe `ns3::AdhocWifiMac`, que herda da classe `ns3::WifiMac`, que é a classe base para todos os objetos WiFi. Essa classe encapsula todas as funcionalidades de baixo nível MAC como DCA, EDCA e todas as funcionalidades de alto nível MAC como máquinas de estado de associação e desassociação.

Para evitar a colisão de pacotes devido ao efeito do terminal oculto, o envio de mensagens RTS/CTS foi incluído na simulação. O envio de pacotes RTS/CTS foi ativado para todos os pacotes, para isso o tamanho mínimo do pacote foi configurado com zero, pois assim o envio de RTS/CTS é ativado para todos os tamanhos de pacotes.

3.2.4 Camada de Transporte

O protocolo UDP (do inglês *User Datagram Protocol*) foi utilizado na camada de transporte. Esse protocolo permite que a aplicação escreva um datagrama encapsulado em um pacote IPv4 ou IPv6, mas não há garantia de entrega do pacote. O UDP foi implementado através da classe `ns3::UdpSocketFactory`, que permite a criação de sockets UDP nos nós fonte e destino. É possível visualizar as mensagens UDP através do arquivos de *pcap* gerados, conforme mostra a Figura 3.2.

3.2.5 Camada de Rede

Para o roteamento, foi utilizado roteamento estático do nó remoto, passando pela estrutura LTE até chegar no nó móvel proxy. As rotas foram definidas manualmente para que fosse possível integrar a estrutura LTE com a estrutura WiFi. Dessa forma foram adicionadas rotas estáticas para que os pacotes fossem encaminhados do nó remoto, passando pelo P-GW e pelo eNB até chegar no UE, que é o nó proxy, gargalo para a transmissão na rede WiFi. Para isso foi utilizada a

No.	Time	Source	Destination	Protocol	Length	Info
204	31.434017	00:00:00_00:00:21 (TA)	00:00:00_00:00:1d (RA)	802.11	42	Request-to-send, Flags=0.....
205	31.434684	00:00:00_00:00:21 (TA)	00:00:00_00:00:21 (RA)	802.11	38	Clear-to-send, Flags=0.....
206	31.434694	1.0.0.2	172.1.1.22	UDP	1558	Source port: 49155 Destination port: http
207	31.447489	00:00:00_00:00:21 (TA)	00:00:00_00:00:21 (RA)	802.11	38	Acknowledgement, Flags=0.....
208	31.447879	00:00:00_00:00:21 (TA)	00:00:00_00:00:11 (RA)	802.11	42	Request-to-send, Flags=0.....
209	31.448545	00:00:00_00:00:21 (TA)	00:00:00_00:00:21 (RA)	802.11	38	Clear-to-send, Flags=0.....
210	31.448555	1.0.0.2	172.1.1.23	UDP	1558	Source port: 49156 Destination port: http
211	31.461349	00:00:00_00:00:11 (TA)	00:00:00_00:00:21 (RA)	802.11	38	Acknowledgement, Flags=0.....
212	31.461751	00:00:00_00:00:11 (TA)	00:00:00_00:00:08 (RA)	802.11	44	Request-to-send, Flags=0.....
213	31.462066	00:00:00_00:00:11 (TA)	00:00:00_00:00:11 (RA)	802.11	38	Clear-to-send, Flags=0.....
214	31.474556	1.0.0.2	172.1.1.23	UDP	1560	Source port: 49156 Destination port: http
215	31.475812	00:00:00_00:00:11 (TA)	00:00:00_00:00:11 (RA)	802.11	38	Clear-to-send, Flags=0.....
216	31.477440	00:00:00_00:00:11 (TA)	00:00:00_00:00:08 (RA)	802.11	44	Request-to-send, Flags=0.....

▶ Frame 1: 108 bytes on wire (864 bits), 108 bytes captured (864 bits)
 ▶ Radiotap Header v0, Length 24
 ▶ IEEE 802.11 Data, Flags: 0.....
 ▶ Logical-Link Control
 ▼ Internet Protocol Version 4, Src: 172.1.1.15 (172.1.1.15), Dst: 172.1.1.255 (172.1.1.255)
 Version: 4
 Header length: 20 bytes
 ▶ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
 Total Length: 48
 Identification: 0x0000 (0)
 ▶ Flags: 0x00
 Fragment offset: 0
 Time to live: 64
 Protocol: UDP (17)

Figura 3.2: Arquivo de *pcap* obtido através do software de análise de pacotes Wireshark. É possível visualizar os pacotes UDP enviados do nó fonte até o nó de destino.

classe `ns3::Ipv4StaticRouting`, que implementa o roteamento estático, fornecendo um conjunto básico de métodos para inserir rotas *unicast* e *multicast* no sistema de roteamento do IPv4. Assim, as rotas foram pré-definidas e inseridas com essa classe para que os pacotes fossem encaminhados do nó remoto até o nó proxy e fosse possível juntar a estrutura LTE com uma rede WiFi.

Na rede *ad hoc* foi utilizado o protocolo OLSR, implementado no ns-3 através da classe `ns3::OlsrHelper`, que adiciona o roteamento OLSR nos nós WiFi. O protocolo OLSR possui um módulo no ns-3 e é definido principalmente na classe `ns3::olsr::RoutingProtocol`. Essa classe implementa as funcionalidades do protocolo OLSR, como seleção de MPR, mensagens HELLO, mensagens TC, tabela de roteamento, etc.

Como o OLSR é um protocolo pró-ativo, foi adicionado um tempo de 30 segundos no início da simulação sem envio de pacotes, pois de acordo com as simulações realizadas, esse seria um bom tempo de convergência para o protocolo OLSR. Uma grande vantagem do OLSR é que ele pode ser instalado em apenas uma interface de um nó. Assim, foi possível instalar tanto o roteamento pelo OLSR em uma das interfaces do nó proxy como o roteamento estático em sua outra interface.

3.2.6 Camada de Aplicação

Quanto à aplicação, foi instalada a aplicação `OnOff` no nó remoto, definida no ns3 pela classe `ns3::OnOffHelper`. Essa aplicação gera um tráfego CBR (do inglês *Constant Bit Rate*) durante o tempo em que estiver ativa (“on”) e não gera nenhum tráfego no período em que estiver inativa

("off"). A aplicação foi configurada para ficar ativa por 1 segundo e inativa por 1 segundo. O tempo total de duração aplicação é de 60 segundos. Durante esse período a aplicação gera dados com taxas de envio de 117.760 b/s, 176.640 b/s e 235.520 b/s. Assim, são enviados, aproximadamente, 10 pacotes a cada 2 segundos para a primeira taxa, 15 pacotes a cada 2 segundos para a segunda taxa e 20 pacotes a cada 2 segundos para a terceira taxa, totalizando aproximadamente 300, 450 e 600 pacotes, respectivamente, para cada taxa durante toda a simulação, já que o tamanho definido para para cada pacote de dados é de 1.472 bytes (11.776 bits). Essas taxas foram escolhidas para que se pudesse comparar os resultados obtidos para uma taxa de dados baixa, uma taxa média e uma taxa alta, sendo que a maior taxa de dados é o dobro da menor taxa.

Foram gerados fluxos de dados do nó remoto até os nós da rede *ad hoc*. As simulações foram feitas para até 5 fluxos, ou seja, de 1 até 5 nós de destino. Os nós de destino foram escolhidos entre os 25 nós da rede *ad hoc* e para cada cenário, as simulações foram feitas para as 5 quantidades de fluxos. Assim, a taxa de envio de dados do `OnOff` foi dividida pela quantidade de fluxos existentes, visando analisar o efeito que a quantidade de fluxos pode gerar no desempenho da rede, ou seja, a quantidade de clientes que a rede poderia atender.

Capítulo 4

Avaliação de Desempenho

Neste capítulo apresentamos os resultados obtidos em nossas simulações. Inicialmente, são apresentadas as definições das medidas de desempenho utilizadas para avaliação, que são a vazão fim-a-fim, atraso médio por pacote, taxa de entrega de pacotes e índice de justiça. Em seguida, são apresentados os resultados de simulação para diferentes cenários de mobilidade, condições de canal e número de fluxos de dados escoados.

4.1 Medidas de Desempenho

Foram consideradas as seguintes figuras de mérito para avaliação de desempenho do escoamento de tráfego via redes *ad hoc*: taxa de entrega de pacotes de dados nos destinatários, vazão média do fluxo, atraso fim-a-fim e índice de justiça. Os valores necessários para os cálculos foram obtidos através da classe `ns3::FlowMonitor`, que monitora e reporta as estatísticas dos fluxos de pacotes durante a simulação.

A taxa de entrega de pacotes, PDR (do inglês *Packet Delivery Ratio*) é a razão entre a quantidade de pacotes recebidos e a quantidade de pacotes transmitidos em um fluxo, expressa em termos percentuais, ou seja,

$$\text{PDR} = \left(\frac{\Sigma \text{ pacotes recebidos}}{\Sigma \text{ pacotes transmitidos}} \right) \times 100. \quad (4.1)$$

Para as simulações realizadas, a PDR foi calculada da seguinte forma:

$$\text{PDR} = \left(\frac{\Sigma \text{ pacotes recebidos de todos os fluxos}}{\Sigma \text{ pacotes transmitidos de todos os fluxos}} \right) \times 100. \quad (4.2)$$

Assim, os resultados obtidos para a PDR nas simulações indicam a taxa média de entrega de pacotes por fluxo.

A vazão média fim-a-fim é calculada como a razão entre a soma do total de bits recebidos e o intervalo de tempo compreendido para recepção desses dados, ou seja,

$$\text{Vazão} = \left(\frac{\text{bits recebidos}}{\Delta t} \right), \quad (4.3)$$

sendo Δt o intervalo de tempo sobre o qual todos os bits foram recebidos.

Assim, é possível obter a quantidade de bits recebidos por segundo por fluxo de dados. Nessas simulações são calculadas as vazões médias por fluxo, da seguinte forma:

$$\text{Vazão Média por Fluxo} = \left(\frac{\Sigma(\text{bytes recebidos por fluxo}) \times 8}{\text{número de fluxos} \times \Delta t} \right), \quad (4.4)$$

sendo que a multiplicação por 8 converte a unidade de bytes em bits.

O atraso médio por pacote é calculado como o intervalo de tempo entre o instante que a camada de rede repassa o pacote para a camada de enlace até o instante em que o pacote é entregue à camada de rede. Nas simulações são calculados o atraso médio dos pacotes, em todos os fluxos, a partir da soma de todos os atrasos por pacote em todos os fluxos, divididos pelo total de pacotes transmitidos em todos os fluxos. O atraso médio é uma medida que reflete o desempenho da rede. Assim, quanto maior o atraso, o pacote gasta mais tempo até chegar no seu destino e pior é a qualidade da rede. E quanto menor o atraso, melhor é o desempenho da rede. Para essas simulações o atraso médio por pacote foi calculado da seguinte forma:

$$\text{Atraso Médio por Pacote} = \left(\frac{\Sigma \text{atrasos de pacotes por fluxo}}{\Sigma \text{pacotes recebidos por fluxo}} \right). \quad (4.5)$$

O índice de justiça indica o quão justo é o compartilhamento de canal na rede, quanto mais próximo de 1 mais justo é o compartilhamento de canal. O índice de justiça utilizado é o de Jain *et al.* [25] definido por:

$$\text{Índice de Justiça} = \left(\frac{(\Sigma_f V_f)^2}{n \times \Sigma_f (V_f)^2} \right), \quad (4.6)$$

sendo V_f a vazão fim-a-fim do fluxo f e n o número de fluxos.

4.2 Variação de Parâmetros

Com o objetivo de obter uma análise comparativa entre diferentes cenários, variamos o valor do parâmetro K de Rice e a taxa de envio de pacotes da aplicação. Para cada valor de K em uma determinada taxa também variamos o número de fluxos e a velocidade em que nós se movimentavam na rede. A Tabela 4.1 mostra os 9 cenários diferentes utilizados nas simulações, em cada um destes cenários avaliamos três diferentes velocidades dos nós (0 m/s, 1 m/s e 3 m/s) e para cada um destes valores variamos também o número de fluxos escoados de um a cinco. Repetimos essa operação para 15 topologias diferentes (posicionamento inicial dos nós) e calculamos a média e o desvio padrão das medidas de desempenho das simulações, que são: vazão média por fluxo, atraso médio fim-a-fim por pacote, taxa de entrega de pacotes por fluxo, e índice de justiça.

Simulamos os 9 cenários alterando no código o valor dos parâmetros. Esses parâmetros foram variados devido à relevância que eles possuem no desempenho da rede, visando obter cenários mais realistas. Por exemplo, o aumento na taxa de dados da aplicação pode gerar filas maiores ao longo dos nós participantes de uma rota, o que gera mais atraso e mais perdas de pacotes. A variação no parâmetro K contribui para uma modelagem mais realista do canal de propagação sem fio,

Cenários de simulação		
	Taxa de Dados (b/s)	Desvanecimento
Cenário 1	117.760	$K = \infty$
Cenário 2	117.760	Distribuição de Rice, $K = 2$
Cenário 3	117.760	Distribuição de Rayleigh, $K = 0$
Cenário 4	176.640	$K = \infty$
Cenário 5	176.640	Distribuição de Rice, $K = 2$
Cenário 6	176.640	Distribuição de Rayleigh, $K = 0$
Cenário 7	235.520	$K = \infty$
Cenário 8	235.520	Distribuição de Rice, $K = 2$
Cenário 9	235.520	Distribuição de Rayleigh, $K = 0$

Tabela 4.1: Cenários de simulação utilizados.

incluindo assim o efeito do desvanecimento em pequena escala, resultado das flutuações rápidas do sinal recebido devido à combinação de múltiplas versões do sinal transmitido. As diferenças nos resultados para os diferentes cenários foram notórias, como veremos mais adiante.

4.3 Resultados de Simulações

4.3.1 Taxa de Entrega de Pacotes

Inicialmente, apresentamos os resultados para taxa de entrega de pacotes para diferentes níveis de desvanecimento de canal e mobilidade, considerando uma taxa de geração de pacotes fixa na aplicação. As Figuras 4.1, 4.2 e 4.3 apresentam os resultados para $K = \infty$, $K = 2$ e $K = 0$, respectivamente, considerando uma taxa de geração de dados a 117.760 b/s apenas (ou seja, Cenários 1, 2 e 3 da Tabela 4.1). É possível perceber nesses três gráficos que a PDR diminui significativamente com o aumento da velocidade dos nós. Por exemplo, para o caso de 2 fluxos na Figura 4.1, a PDR sofre uma decaimento de 6,1 % do caso estático para o caso com velocidade de 1 m/s. Para o caso de velocidade 3 m/s há uma decaimento de 44,8% em relação ao caso estático. Assim, a PDR é maior quando os nós estão estáticos. No cenário 1 (Figura 4.1), que possui taxa de 117.760 b/s e $K = \infty$, percebemos que a PDR é alta, com valores acima de 90% para o caso estático e para todos os conjuntos de fluxos. Já para o cenário com os nós se movendo a 3 m/s, a taxa de entrega de pacotes é diminuída drasticamente.

A mobilidade nos nós pode afetar a formação de rotas pelo protocolo OLSR, podendo causar quebra de rotas e muitas atualizações de tabela de roteamento, já que a posição dos nós muda rapidamente. Isso afeta então a escolha dos nós MPRs, que pode mudar rapidamente, sendo necessário recalcular os MPRs e também o envio de mais mensagens TC. É justificável então a grande perda de pacotes para o caso em que a velocidade é 3 m/s, já que os nós se movem rapidamente, tornando o ambiente instável, sujeito a falhas nos enlaces bidirecionais e dificuldade

na criação de novos enlaces bidirecionais.

É possível notar, através da Figura 4.1, que à medida que o número de fluxos aumenta, o comportamento intra-grupo se mantém relativamente constante. Ou seja, o aumento da subdivisão da taxa para mais fluxos, com destinos diferentes, não traz grande variação na queda das PDRs de uma velocidade para outra. Assim, a relação interna entre as PDRs para as três velocidades não sofre grande mudança com o aumento do número de fluxos.

Observando os resultados obtidos para diferentes valores do parâmetro K de Rice na Figura 4.2 e na Figura 4.3, percebe-se que a PDR sofre decaimento, apresentando valores mais baixos para o caso com $K = 0$. Observa-se na Figura 4.2, em que $K = 2$, que a PDR sofre um decaimento de 45,7% em comparação com a Figura 4.1, em que $K = \infty$, considerando 1 fluxo na velocidade de 1 m/s. Para $K = 0$, a PDR sofre um decaimento 49,9%. Nota-se que a PDR também varia com o número de fluxos das simulações. Na Figura 4.2, para o caso estático, temos uma PDR de aproximadamente 57% para 1 fluxo, enquanto que pra 5 fluxos a PDR é dada por aproximadamente 42%. Já no caso em que a velocidade é 3 m/s, obtemos resultados diferentes do esperado. Temos uma PDR de aproximadamente 25% para 1 fluxo e de 28% para 5 fluxos, apresentando, portanto, maior taxa de entrega de pacotes com o aumento do número de fluxos. Esse comportamento, observado também na Figura 4.1 e na Figura 4.3, pode ser explicado constatando-se que quando a velocidade é de 3 m/s, ocorrem mudanças na topologia da rede de forma mais rápida, provocando mais perdas de pacotes. Mas quando o número de fluxos é aumentado, existe maior chance de que pelo menos em um fluxo os pacotes consigam chegar até o destino, aumentando então a média da taxa de entrega de pacotes para todos os fluxos.

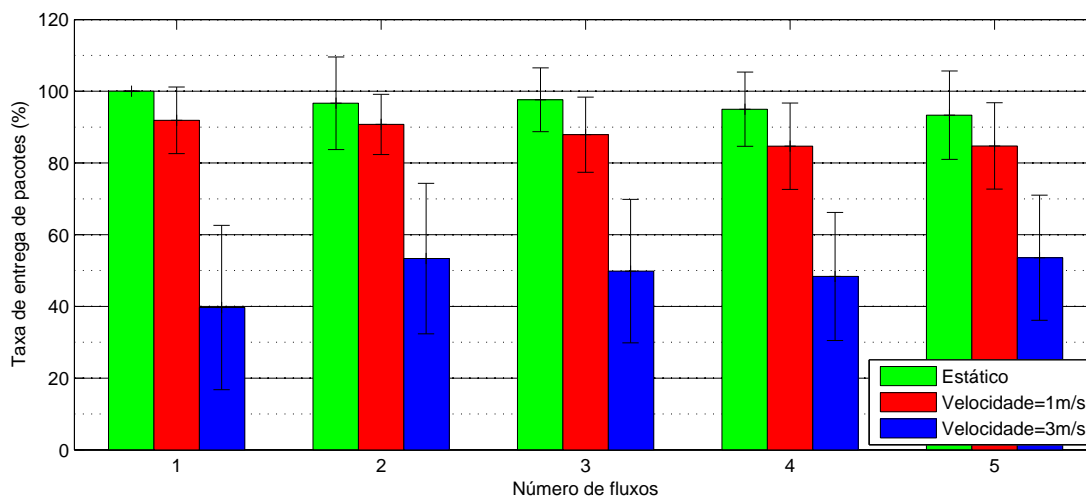


Figura 4.1: Taxa de Entrega de Pacotes para Cenário 1: $K = \infty$ e taxa= 117.760 b/s.

A PDR para os três cenários com taxa de 176.640 b/s, pode ser vista nas Figuras 4.4, 4.5 e 4.6. Percebe-se que houve uma melhora na PDR na maioria dos casos para essa taxa do que nos três cenários anteriores, que possuem menor taxa de envio de dados. Por exemplo, na Figura 4.5 podemos notar que a PDR teve um aumento de 22,2 % com $K = 2$, em comparação com a Figura

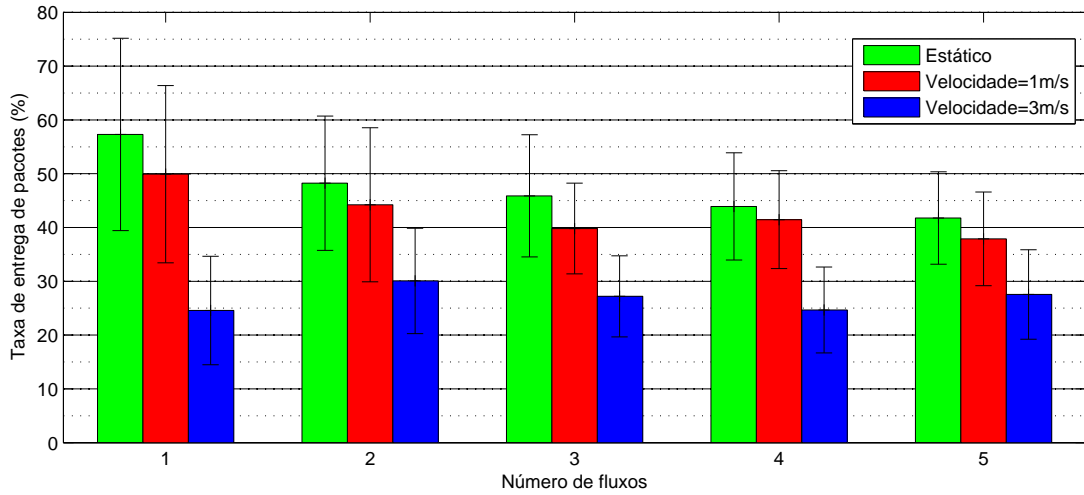


Figura 4.2: Taxa de Entrega de Pacotes para Cenário 2: $K = 2$ e taxa= 117.760 b/s.

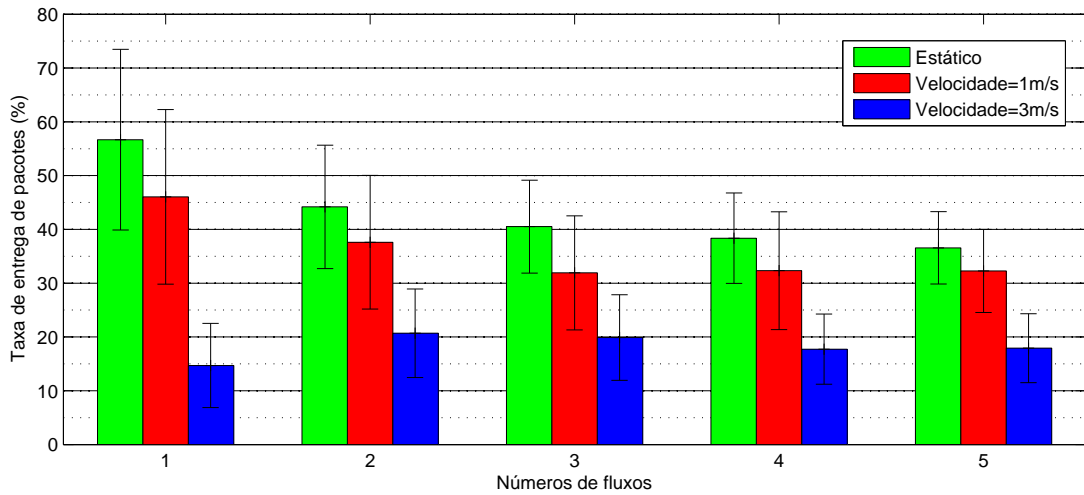


Figura 4.3: Taxa de Entrega de Pacotes para Cenário 3: $K = 0$ e taxa= 117.760 b/s.

4.2, em que a taxa de dados é de 117.760 b/s, considerando 4 fluxos no caso estático. Nota-se então que o aumento da taxa de dados provocou um aumento razoável na PDR.

Comparando a Figura 4.4 com a Figura 4.6 é possível perceber que o desvanecimento modelado para $K = 0$ resulta em um pior desempenho da rede simulada, com mais perda de pacotes, devido aos efeitos de propagação do canal. É possível observar um decaimento de 47,1 % do cenário com $K = \infty$ para o cenário com $K = 2$ e de 57,3% para o cenário com $K = 0$, considerando 5 fluxos de dados para a velocidade de 3 m/s. Com a Figura 4.6 podemos ver os resultados das simulações em que os nós de destino apenas recebem componentes através de multipercursos, enquanto que na Figura 4.5 os nós de destino também recebem um raio direto, ou seja, uma componente dominante. Dessa forma, essa grande diferença entre as PDRs das Figuras 4.5 e 4.6 demonstram o predomínio

da componente direta com $K = 2$ e a contribuição de potência dessa componente, tornando menor a influência do multipercurso.

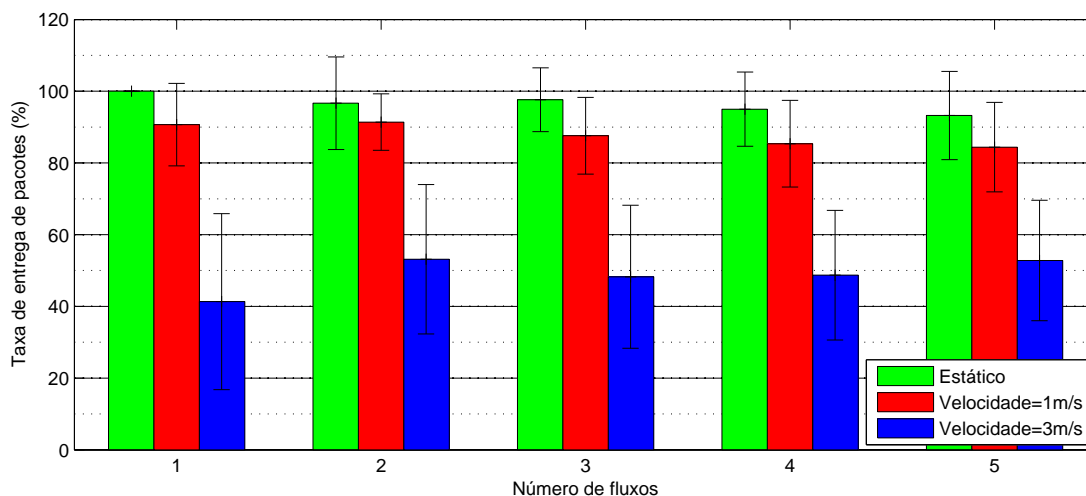


Figura 4.4: Taxa de Entrega de Pacotes para Cenário 4: $K = \infty$ e taxa= 176.640 b/s.

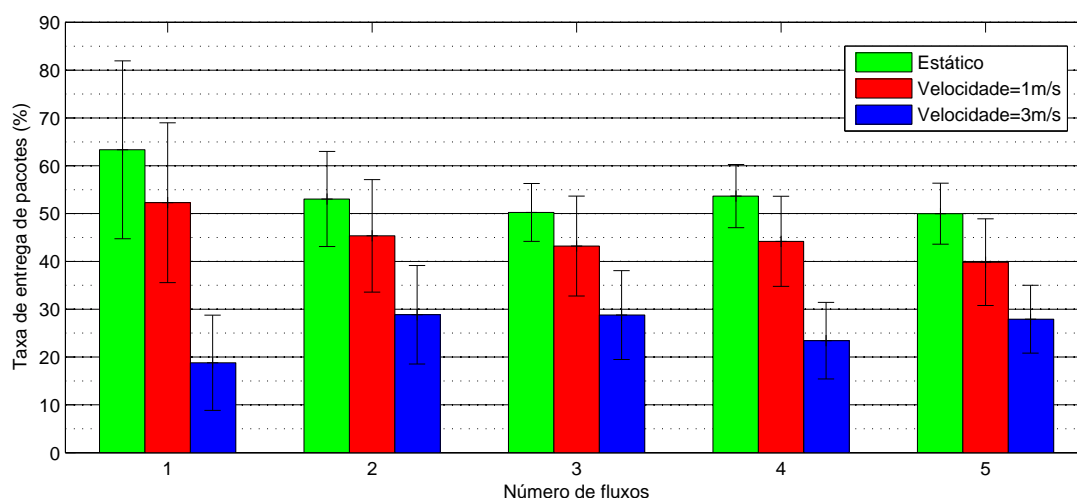


Figura 4.5: Taxa de Entrega de Pacotes para Cenário 5: $K = 2$ e taxa= 176.640 b/s.

A PDR para os cenários com taxa de 235.520 b/s pode ser vista nas Figuras 4.7, 4.8 e 4.9. Na maioria dos casos, temos resultados semelhantes comparando os resultados para velocidades, número de fluxos e valores de K diferentes. Por exemplo, a PDR para a velocidade de 1 m/s para 2 fluxos com $K = 2$, é de 44,2 % para a taxa de 117.760 b/s, enquanto que para a taxa de 235.520 b/s é de 40,8 %, o que representa um decaimento de somente 7,7 %. Assim como nessa situação, para outros cenários, a PDR também sofre pouco decaimento com o aumento da taxa de dados da aplicação.

Nota-se que em todos os resultados para PDR, a PDR para 1 fluxo foi menor do que para 2

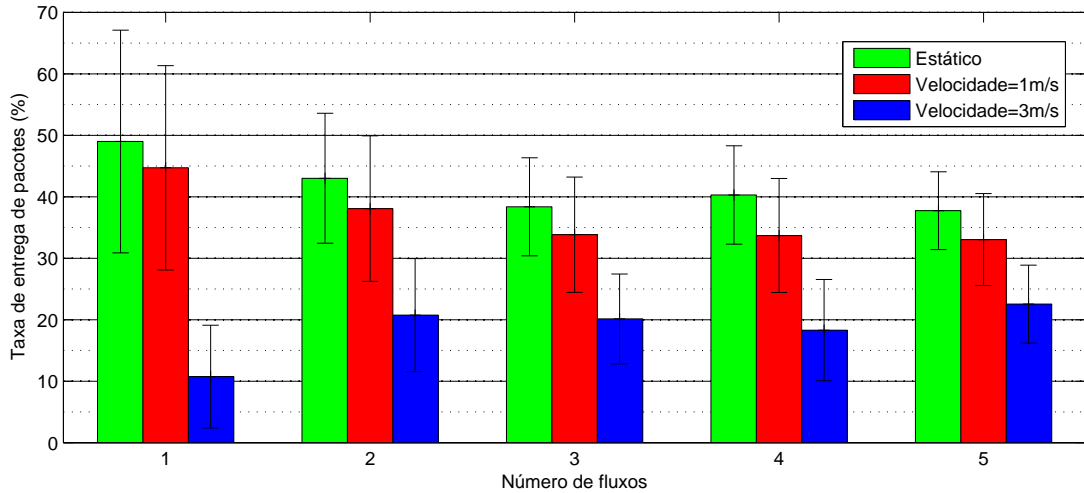


Figura 4.6: Taxa de Entrega de Pacotes para Cenário 6: $K = 0$ e taxa= 176.640 b/s.

fluxos ou mais, em todos os cenários. Foi observado durante as simulações que quando há apenas 1 fluxo, os resultados podem ser muito divergentes, pois quando o nó de destino está próximo ao nó proxy, a probabilidade de que o nó seja alcançado pelo nó proxy é grande. Mas ocorreram casos em que o nó de destino se encontrava distante do nó proxy, isolado em alguma posição ou em um conjunto de nós vizinhos que não eram ouvidos pelos vizinhos do nó proxy, devido ao limiar de detecção de energia e ao CCA. A formação de grupos de nós vizinhos distantes a mais de 150 metros de outros grupos de nós faz com que alguns nós não sejam alcançados pelo nó proxy. Assim, se o nó de destino estiver em algum desses grupos de nós que conseguem se comunicar entre si mas não conseguem se comunicar com outros grupos, a PDR desse nó é zero, pois o roteamento de pacotes até esse nó não é possível. Como a PDR foi calculada como a média das PDRs para cada fluxo, quando há apenas um fluxo e ele é zero, a PDR média então é igual a zero. Quando há mais que um fluxo, é provável que pelo menos 1 fluxo tenha PDR maior que zero, resultando quase sempre em uma média maior que zero. Isso justifica então a PDR média para um fluxo ser sempre menor em comparação com 2 ou mais fluxos, pois dependendo da topologia utilizada, a PDR para algumas topologias foi frequentemente zero, resultando em baixas médias de PDR.

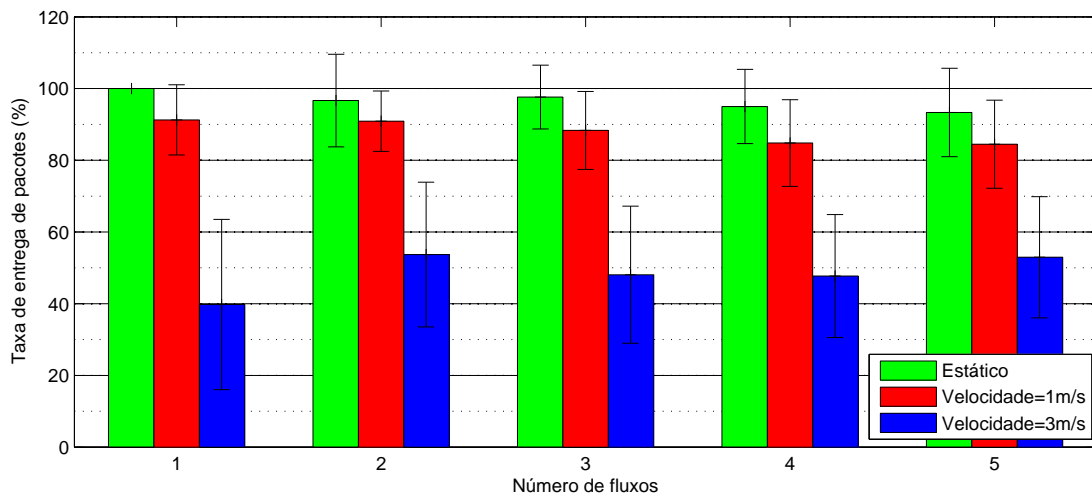


Figura 4.7: Taxa de Entrega de Pacotes para Cenário 7: $K = \infty$ e taxa= 235.520 b/s.

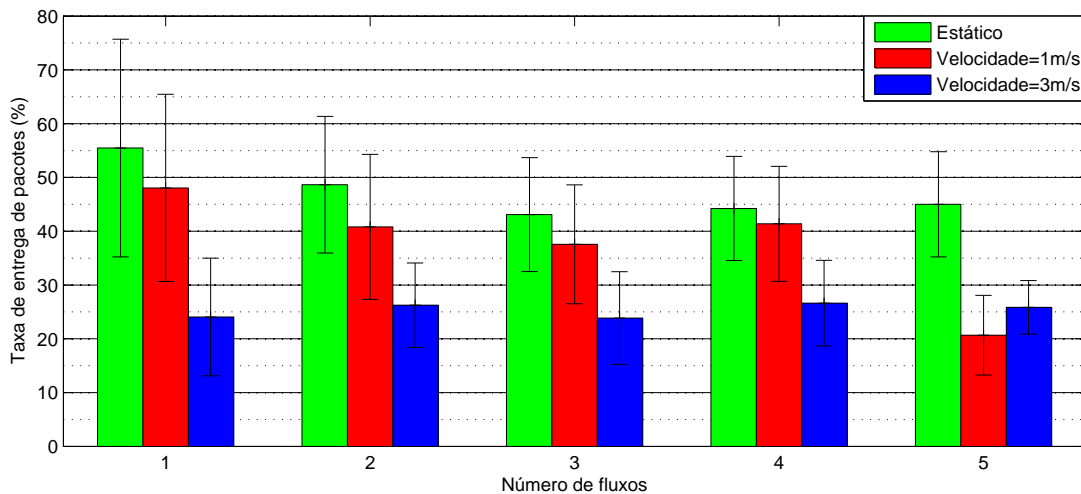


Figura 4.8: Taxa de Entrega de Pacotes para Cenário 8: $K = 2$ e taxa= 235.520 b/s.

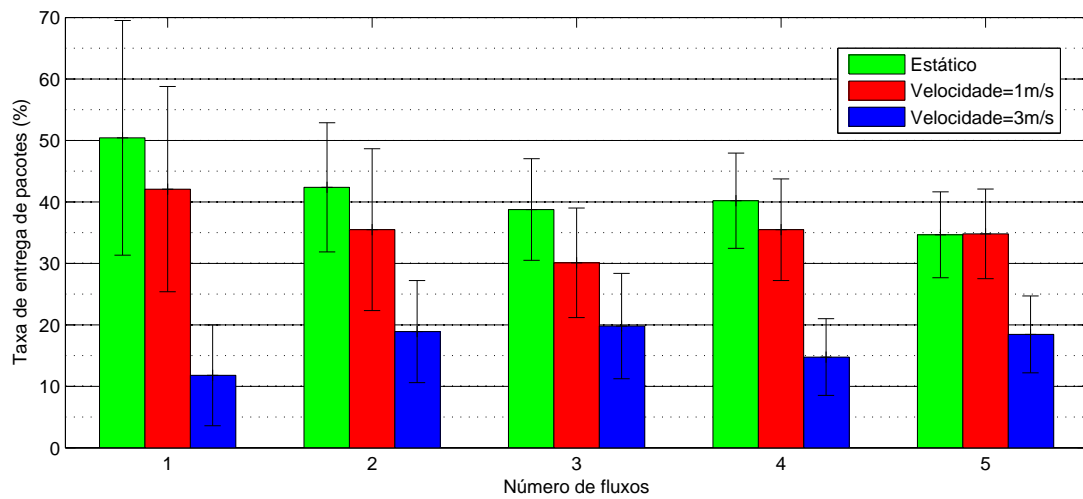


Figura 4.9: Taxa de Entrega de Pacotes para Cenário 9: $K = 0$ e taxa= 235.520 b/s.

4.3.2 Vazão Média por Fluxo

A vazão indica a taxa de recepção efetiva dos dados em bits, e é um parâmetro necessário para a operação adequada de qualquer tipo de aplicação. Assim, dependendo da aplicação utilizada, os nós transmissores devem atender à uma vazão mínima exigida pela aplicação. As Figuras 4.10 a 4.18 mostram os resultados obtidos para vazão nos 9 cenários diferentes.

As Figuras 4.10, 4.11 e 4.12 apresentam a vazão com a menor taxa simulada, de 117.760 b/s. Essa taxa representa aplicações que requerem menor vazão, como aplicações de voz. A maior vazão recebida para essa taxa de dados foi de aproximadamente 55 kb/s, no cenário 1, para 1 fluxo, $K = \infty$, no caso estático. Assim, dos 117 kb/s originais, apenas metade da vazão é recebida, 55 kb/s. Isso indica então que metade da vazão é perdida com o repasse de dados via múltiplos saltos. Já para 5 fluxos, a vazão recebida deveria ser de 23 kb/s, dividindo a taxa pelos 5 fluxos, mas a vazão obtida é aproximadamente 14 kb/s, ou seja, aproximadamente 40% da vazão é perdida.

Podemos perceber que a vazão tem um maior decaimento do caso estático para a velocidade de 3 m/s, para todos os conjuntos de fluxos. Por exemplo, no caso em que $K = 2$ (Figura 4.11), temos um decaimento da vazão de aproximadamente 13% do caso estático para o caso com velocidade = 1m/s e de 57 % para o caso com velocidade = 3 m/s, considerando apenas 1 fluxo. Essa diferença inter-grupo é menor quando aumentamos o número de fluxos, obtendo menor decaimento da vazão com o aumento da velocidade.

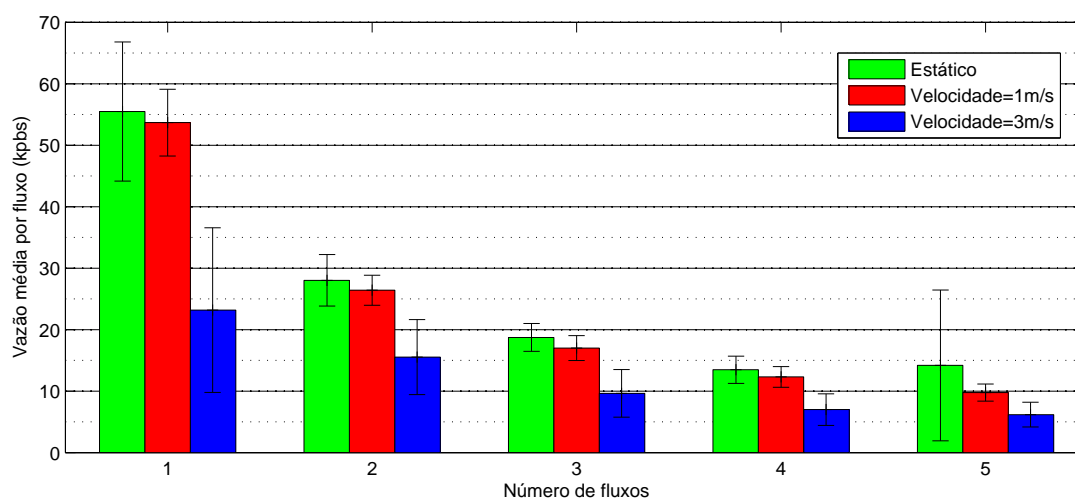


Figura 4.10: Vazão média por fluxo para Cenário 1: $K = \infty$ e taxa= 117.760 b/s.

As Figuras 4.13, 4.14 e 4.15 mostram a vazão média por fluxo para a taxa de dados de 176.640 b/s, que correspondem aos cenários com $K = \infty$, $K = 2$ e $K = 0$, respectivamente. Para esses cenários, a maior vazão obtida foi aproximadamente 88 kb/s para $K = \infty$ no cenário estático, com apenas 1 fluxo. Já a menor vazão obtida foi aproximadamente 4 kb/s para $K = 0$, no cenário com velocidade dos nós de 3 m/s para 5 fluxos de dados. Podemos observar na Figura 4.13, que dos 176 kb/s transmitidos, chegam aproximadamente 88 kb/s para 1 fluxo no caso estático, ou seja, aproximadamente 50 % da vazão transmitida. Já para 5 fluxos, quando dividimos a taxa por 5 fluxos,

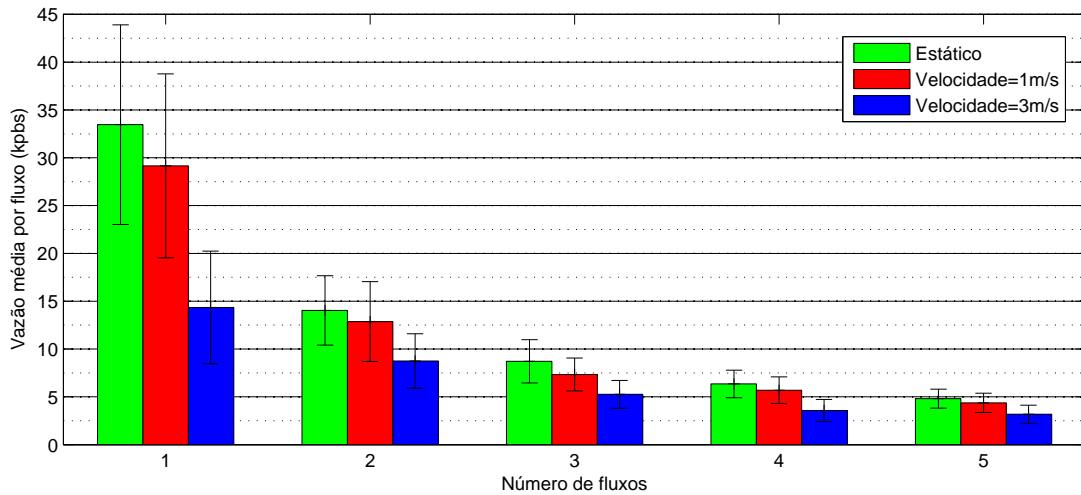


Figura 4.11: Vazão média por fluxo para Cenário 2: $K = 2$ e taxa= 117.760 b/s.

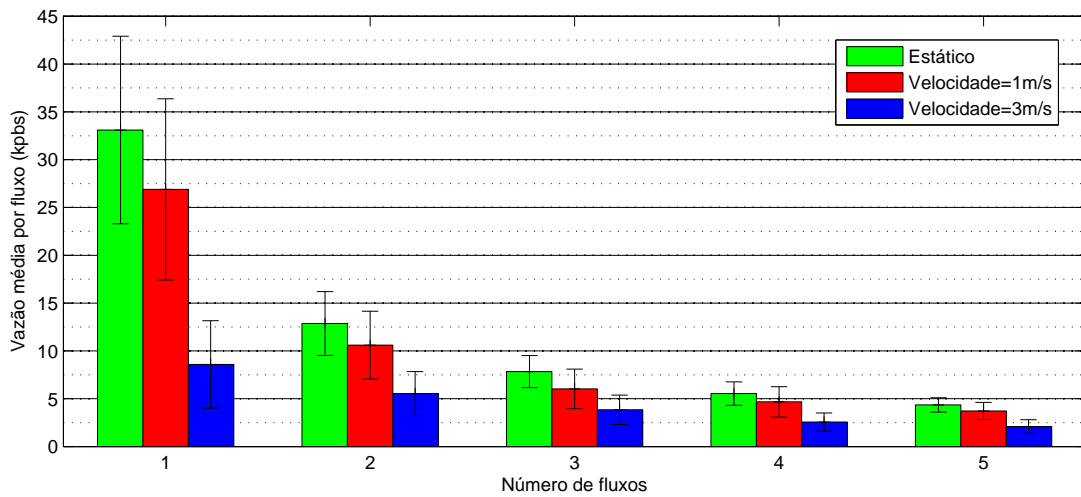


Figura 4.12: Vazão média por fluxo para Cenário 3: $K = 0$ e taxa= 117.760 b/s.

percebemos que dos 35 kb/s transmitidos, aproximadamente 16 kb/s chegam ao receptor, o que representa 46% da vazão transmitida, ou seja, há um decaimento maior da vazão para 5 fluxos de dados do que para apenas 1 fluxo.

É possível notar que a vazão diminui de forma relevante com o aumento de fluxos. Podemos observar isso no gráfico da Figura 4.13, em que a vazão sofre um decaimento de 74,7% de 1 fluxo para 5 fluxos no caso estático. Isso se deve ao fato de que a taxa de envio de pacotes foi dividida pela quantidade de fluxos nessas simulações, para que fosse possível avaliar o efeito do aumento de fluxos com um nó que tem a mesma taxa de envio de pacotes, independente do número de fluxos. Além disso, todo o tráfego é escoado pelo nó proxy, que é o gargalo de todas as transmissões. Esse fato também contribui para que a vazão diminua com o aumento da quantidade de fluxos. Caso os

fluxos fossem divididos com outros nós de gargalo, é provável que os resultados obtidos para vazão seriam maiores, já que a taxa de dados em cada nó seria dividida por uma número menor de fluxos. Dessa forma, percebe-se que a vazão sempre diminui com o aumento da quantidade de fluxos, já que a taxa de dados da aplicação também é dividida proporcionalmente à quantidade de fluxos.

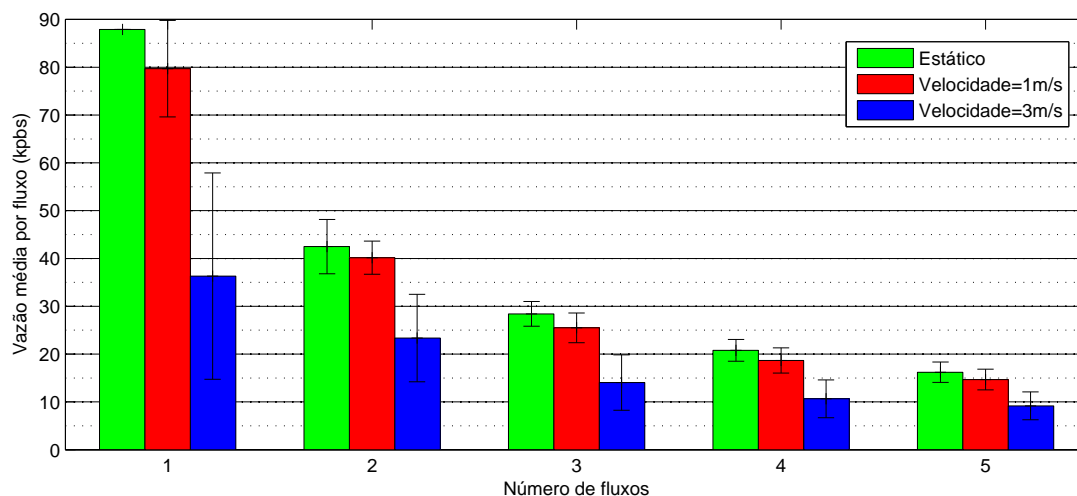


Figura 4.13: Vazão média por fluxo para Cenário 4: $K = \infty$ e taxa= 176.640 b/s.

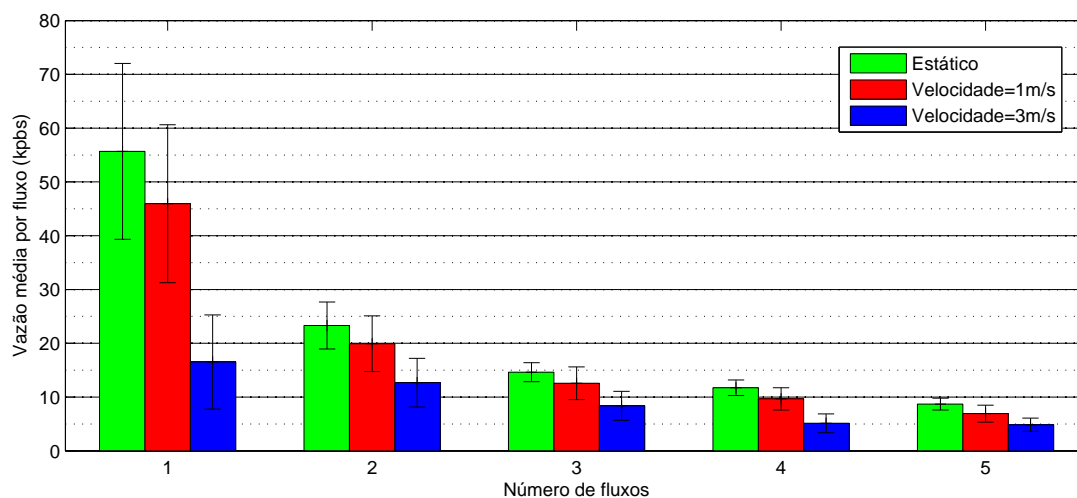


Figura 4.14: Vazão média por fluxo para Cenário 5: $K = 2$ e taxa= 176.640 b/s.

As Figuras 4.16, 4.17 e 4.18 mostram a vazão média por fluxo para a maior taxa de dados simulada, de 235.520 b/s, que correspondem aos cenários com $K = \infty$, $K = 2$ e $K = 0$, respectivamente. Nesses cenários, a maior vazão obtida foi de aproximadamente 118 kb/s, para 1 fluxo no caso estático, visto na Figura 4.16. Já a menor vazão obtida, foi de aproximadamente 4 kb/s no cenário com os nós se movendo a 3 m/s e com 5 fluxos de dados, na Figura 4.18. Nota-se que a vazão diminui significativamente com o cenário 9, em que $K = 0$, como pode ser visto na Figura 4.18. Em relação aos dados obtidos para $K = \infty$, nota-se um decaimento de 51,2% para o

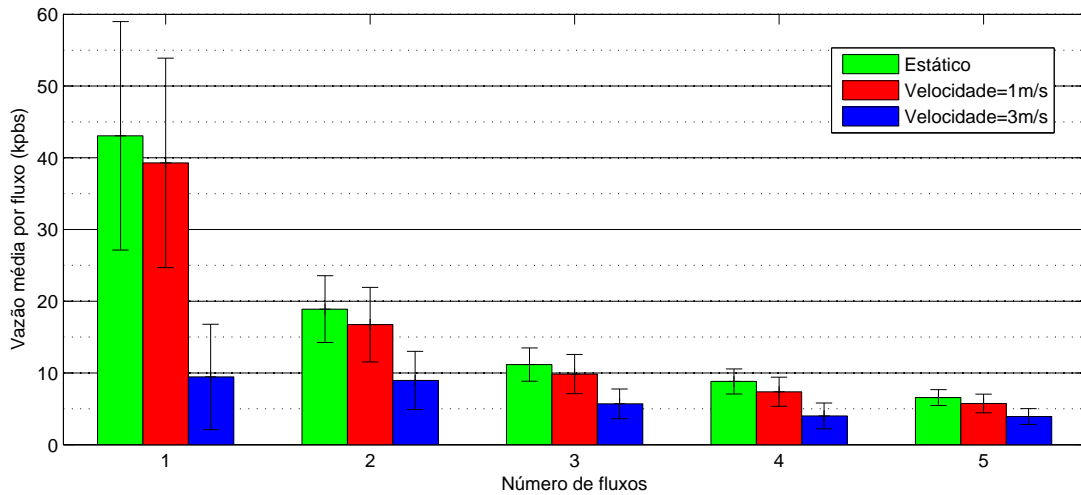


Figura 4.15: Vazão média por fluxo para Cenário 6: $K = 0$ e taxa= 176.640 b/s.

cenário com $K = 2$ e de 58,2% para $K = 0$, considerando 4 fluxos de dados e velocidade dos nós de 1m/s. Podemos perceber então que o efeito do desvanecimento em pequena escala provocou grande decaimento também para vazão obtida nas simulações, assim como na PDR.

Além disso, é possível perceber também que quando a velocidade dos nós aumenta, a vazão tende a cair, como pode ser visto na Figura 4.17, em que a vazão para 1 fluxo no caso estático é de aproximadamente 65 kb/s enquanto que com a velocidade de 3 m/s é de somente 28 kb/s. Podemos observar no gráfico da Figura 4.16, que vazão tem maior decaimento quando a velocidade é aumentada para o caso de apenas 1 fluxo. Nesse caso a vazão cai 16,6% do cenário estático para o cenário com a velocidade dos nós de 1 m/s. Já para a velocidade de 3 m/s a vazão tem um maior decaimento, de 80% em relação ao cenário estático.

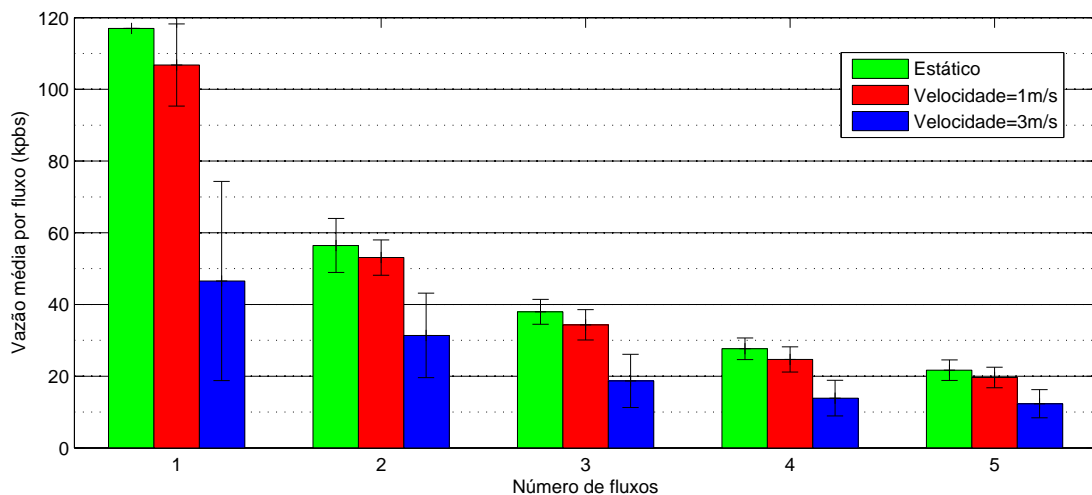


Figura 4.16: Vazão média por fluxo para Cenário 7: $K = \infty$ e taxa= 235.520 b/s.

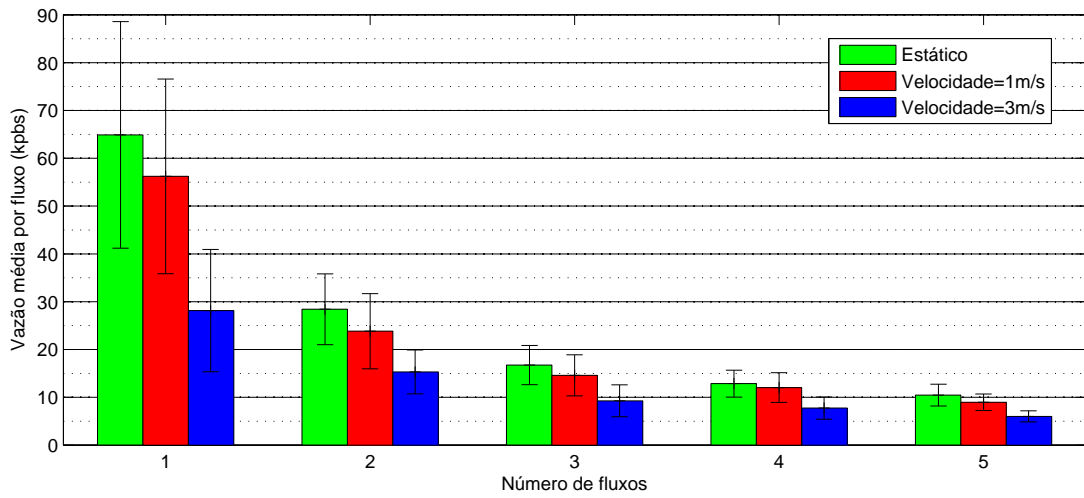


Figura 4.17: Vazão média por fluxo para Cenário 8: $K = 2$ e taxa= 235.520 b/s.

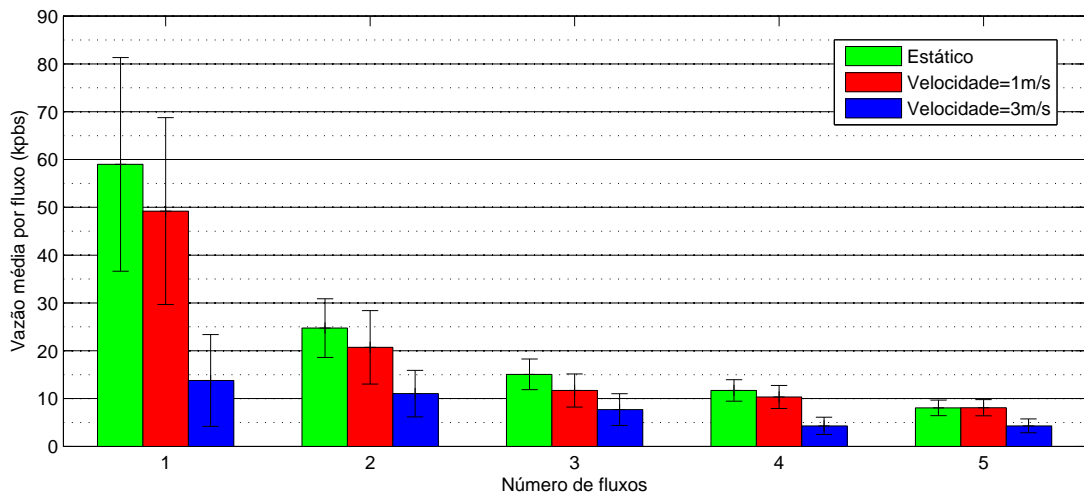


Figura 4.18: Vazão média por fluxo para Cenário 9: $K = 0$ e taxa= 235.520 b/s.

Podemos concluir que para as três taxas de dados geradas e as vazões obtidas, a rede simulada poderia suportar aplicações que requerem uma vazão menor que 100 kb/s, como aplicações transacionais, voz, aplicações Web e transferência de arquivos. Já para aplicações como vídeo e teleconferências, a rede simulada, com as três taxas de dados consideradas, não seria adequada, já que não iria proporcionar vazões suficientes para esse tipo de aplicação, como 500 kb/s ou 1 Mb/s. Para suportar aplicações que requerem maior vazão, seria necessário aumentar a taxa de dados, porém, isso poderia ser prejudicial para outras medidas de desempenho como a taxa de entrega de pacotes e o atraso, que é abordado adiante.

4.3.3 Atraso Médio por Pacote

O atraso médio por pacote representa o tempo médio total necessário para entregar um pacote do nó fonte até o nó destino. O atraso médio por pacote é calculado como o intervalo de tempo entre o instante que a camada de rede repassa o pacote para a camada de enlace até o instante em que o pacote é entregue à camada de rede. Estão incluídos nesse atraso o atraso de propagação, o atraso de fila, o atraso de processamento, o atraso de transmissão em cada salto e o atraso devido à contenção do canal, causado pelo mecanismo de transmissão do IEEE 802.11. A soma desses atrasos representa o tempo de entrega de informação para a aplicação.

O atraso para os cenários com taxa de dados da aplicação de 117.760 b/s pode ser visto nas Figuras 4.19, 4.20 e 4.21. Esses cenários possuem uma mesma taxa de dados, variando o parâmetro K de uma para outra. No cenário 1, que possui taxa de dados de 117.760 b/s e $K = \infty$, pode-se observar que o maior atraso foi de aproximadamente 115 ms, obtido para 5 fluxos com velocidade de 3m/s. O menor atraso foi alcançado para o caso estático com apenas 1 fluxo, que foi de aproximadamente 50 ms.

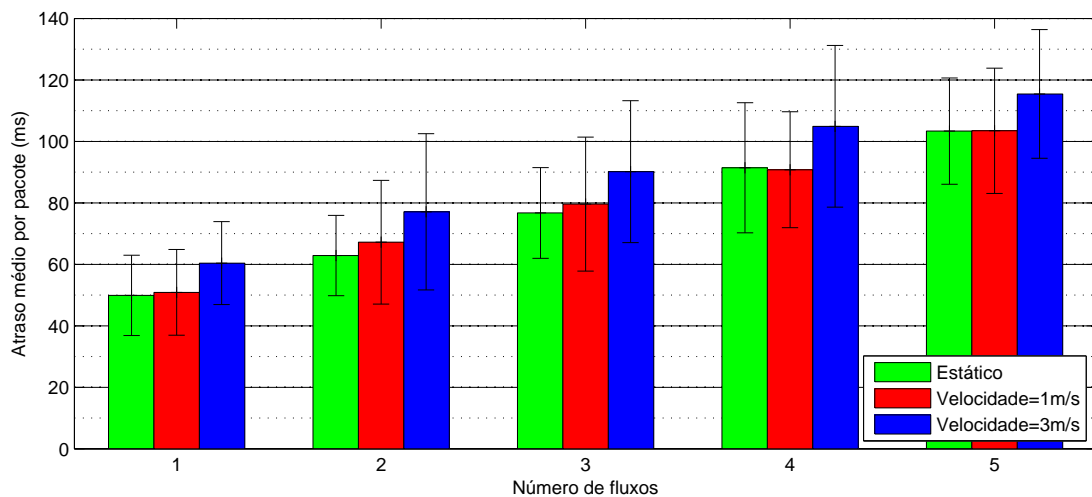


Figura 4.19: Atraso médio por pacote para Cenário 1: $K = \infty$ e taxa= 117.760 b/s.

O atraso médio por pacote para os cenários 4, 5 e 6, com taxa de dados da aplicação de 176.640 b/s podem ser observados nas Figuras 4.22, 4.23 e 4.24, respectivamente. O cenário com $K = \infty$

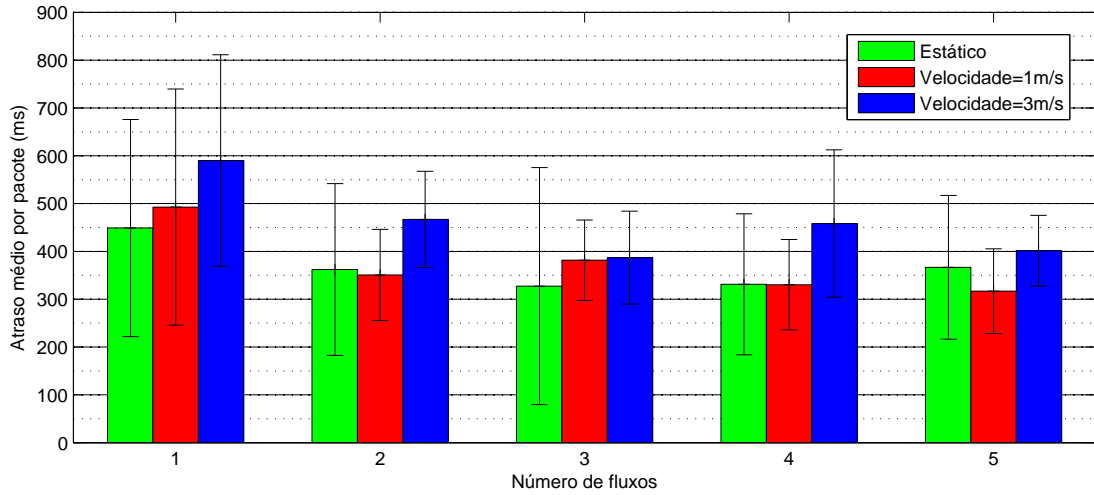


Figura 4.20: Atraso médio por pacote para Cenário 2: $K = 2$ e taxa= 117.760 b/s.

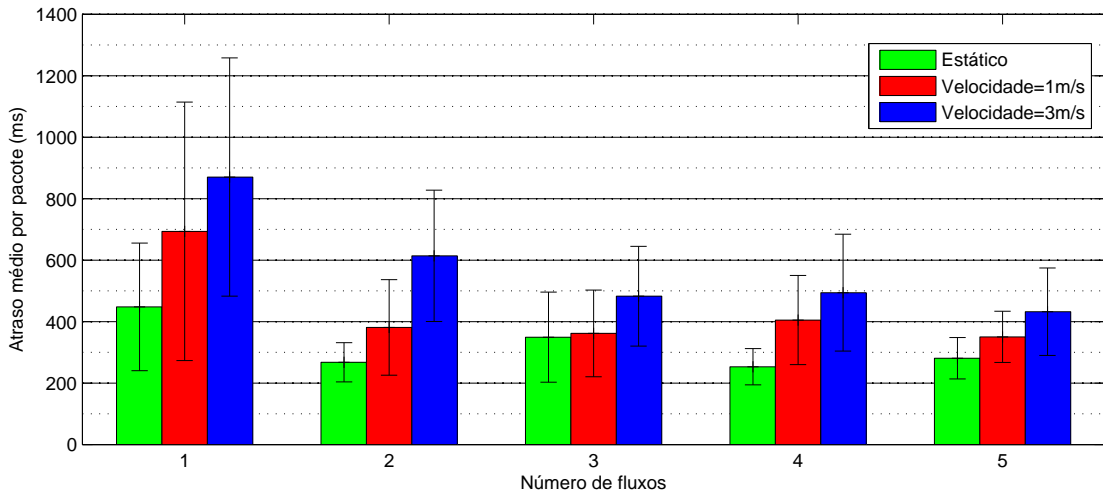


Figura 4.21: Atraso médio por pacote para Cenário 3: $K = 0$ e taxa= 117.760 b/s.

apresenta menores atrasos se comparado com $K = 2$ e para $K = 0$, pois um pior canal tem mais contenção, provocando maiores atrasos. É possível perceber na Figura 4.22 que o atraso cresce proporcionalmente com o aumento de fluxos para o cenário com $K = \infty$. Já nas Figuras 4.23 e 4.24 podemos notar que o atraso tende a diminuir com o aumento de fluxos com $K = 2$ e $K = 0$. Ou seja, o aumento do número de fluxos para os cenários com $K = \infty$ provoca um maior atraso médio por pacote, enquanto que nos cenários com $K = 2$ e $K = 0$ o aumento do número de fluxos provoca um menor atraso médio por pacote. É provável que a inclusão do desvanecimento em pequena escala tenha causado maior efeito para o caso de 1, 2 e 3 fluxos, devido à posição dos nós de destino para esses casos. Já para os casos de 4 e 5 fluxos, o efeito do desvanecimento provocou um crescimento menos significativo do atraso. Pressupõe-se que os nós adicionados para o caso de 4 e 5 fluxos estavam melhor posicionados em relação ao nó fonte do que os nós para 1, 2 e 3

fluxos e por isso o atraso foi aumentado de forma mais significativa para $K = 2$ e $K = 0$ com um menor número de fluxos. Na Figura 4.22, em que $K = \infty$ podemos perceber que o atraso é aumentado em 103,3% de 1 fluxo para 5 fluxos com velocidade de 1 m/s. Já para $K = 2$, houve um decaimento no atraso médio por pacote de 39,2%, e para $K = 0$ houve um decaimento de 45,2%. É provável que isso tenha ocorrido devido a alguns fluxos terem menos saltos e menos disputa de canal, fazendo com que na média, tenhamos menor atraso com o aumento de fluxos.

Na Figura 4.23, com um cenário que possui taxa de dados de 176.640 b/s e $K = 2$, é possível perceber que o atraso permaneceu na faixa entre 300 ms e 500 ms para a maioria da quantidade de fluxos. Esse é um valor tolerável de atraso para algumas aplicações comuns em redes móveis, mas nem todas, pois de acordo com [26], é preferido um atraso menor que 150 ms para aplicações de áudio e vídeo. Já na Figura 4.24, podemos ver que o atraso chegou a 1000 ms com $K = 0$. Isso é justificado devido ao aumento da probabilidade de perdas, gerando mais retransmissões de quadros e aumentando o tempo de *backoff* necessário para as estações poderem transmitir os quadros.

O aumento da velocidade dos nós também proporciona um aumento no atraso médio por pacote. Ao observar a Figura 4.24, podemos perceber que o atraso não sofre grande variação do caso estático para o caso em que a velocidade dos nós é de 1 m/s. Mas para a velocidade de 3 m/s a variação já é maior. Para 1 fluxo, o atraso sofre um aumento de 6% do cenário estático para o cenário com velocidade dos nós de 1m/s. Esse aumento já é maior para a velocidade de 3 m/s, pois chega a 35% de aumento em relação ao caso estático. Podemos perceber então que uma maior mobilidade dos nós proporciona maior atraso para que o pacote chegue ao seu destino.

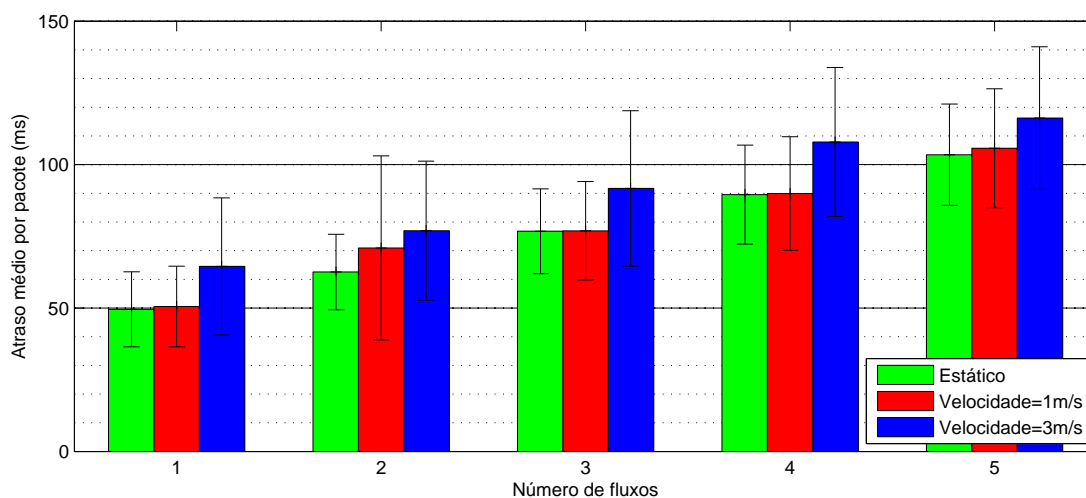


Figura 4.22: Atraso médio por pacote para Cenário 4: $K = \infty$ e taxa= 176.640 b/s.

O atraso médio por pacote para os cenários 7, 8 e 9, que possuem taxa de dados de 235.520 b/s, podem ser observados nas Figuras 4.25, 4.26 e 4.27, respectivamente. É possível notar que há um aumento no atraso médio por pacote conforme o aumento da taxa de dados. Por exemplo, para 3 fluxos de dados e $K = 2$, o atraso médio por pacote foi aumentado em 18,2 % da taxa de 117.760 b/s para a taxa de 176.640 b/s, e teve um aumento de 45,2% da taxa de 176.640 b/s

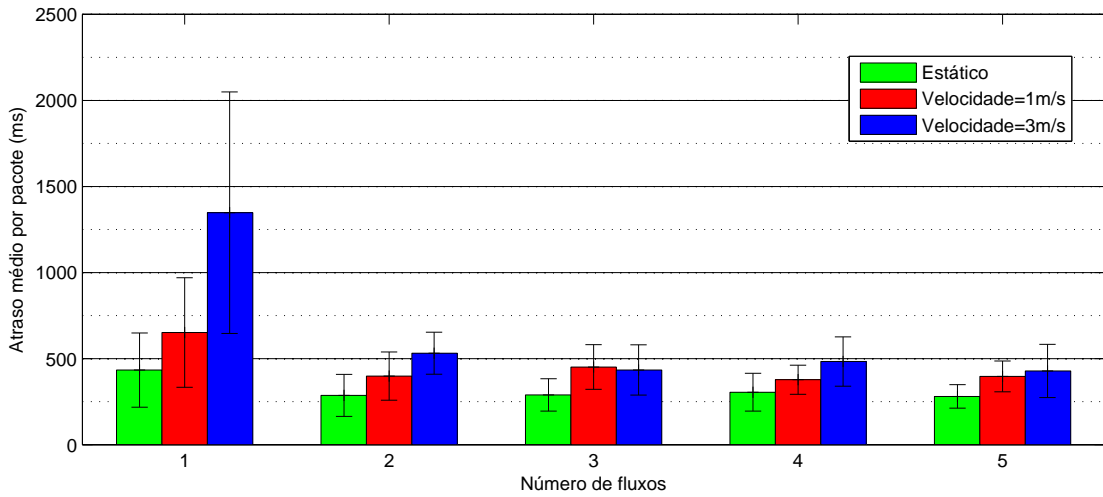


Figura 4.23: Atraso médio por pacote para Cenário 5: $K = 2$ e taxa= 176.640 b/s.

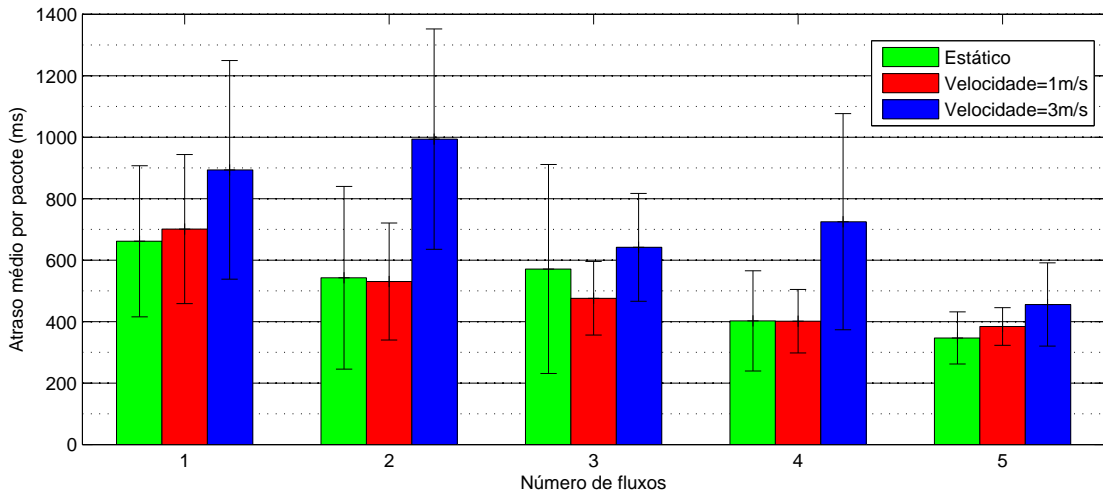


Figura 4.24: Atraso médio por pacote para Cenário 6: $K = 0$ e taxa= 176.640 b/s.

para a maior taxa, de 235.520 b/s. Portanto, o aumento na taxa de dados proporciona também um aumento do atraso médio por pacote. Isso ocorre, devido às filas de pacotes que são formadas quando há mais pacotes a serem transmitidos por segundo, ou seja, os pacotes ficam um maior tempo na fila aguardando serem enviados ao próximo nó da rede, sendo esse tempo maior quando há congestionamento na rede.

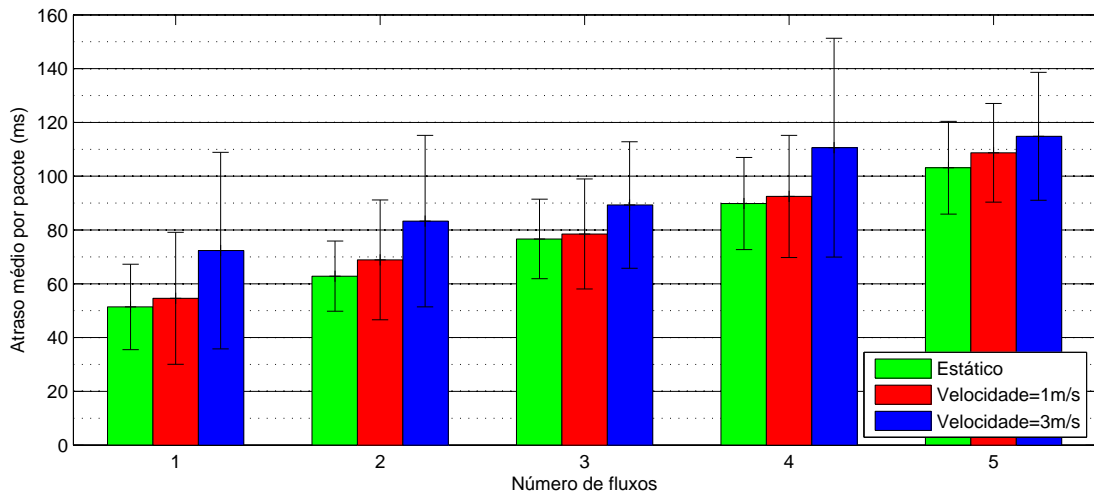


Figura 4.25: Atraso médio por pacote para Cenário 7: $K = \infty$ e taxa= 235.520 b/s.

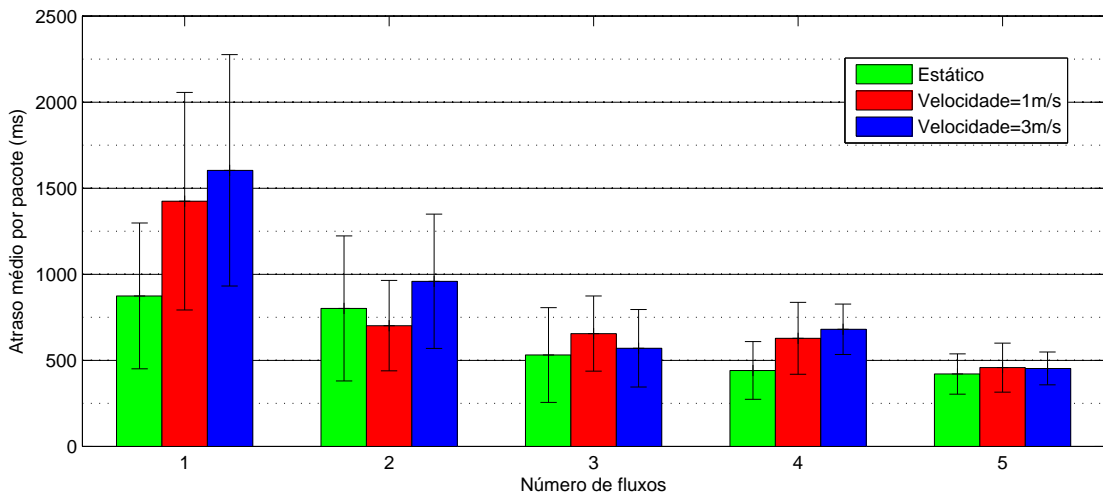


Figura 4.26: Atraso médio por pacote para Cenário 8: $K = 2$ e taxa= 235.520 b/s.

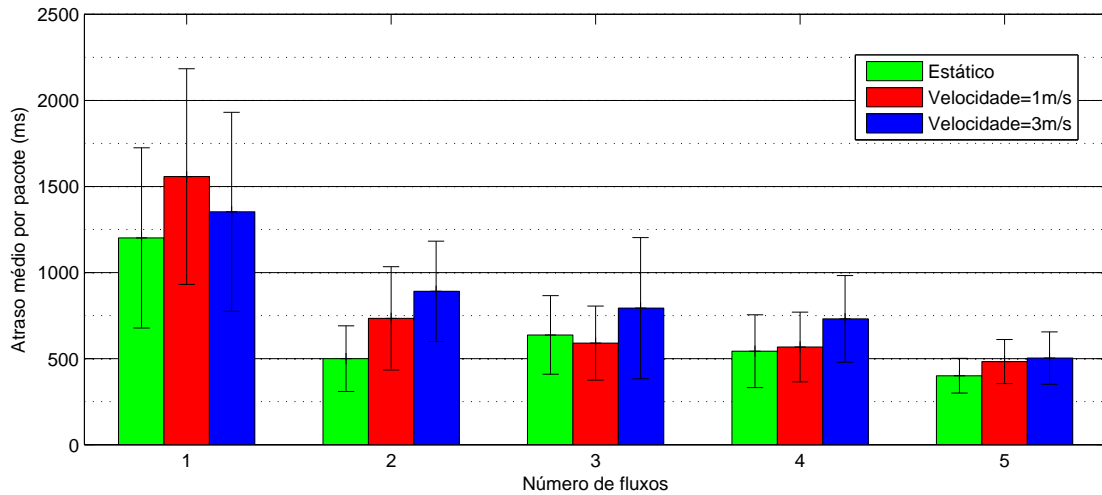


Figura 4.27: Atraso médio por pacote para Cenário 9: $K = 0$ e taxa= 235.520 b/s.

Em relação ao parâmetro K de Rice, os valores obtidos para o atraso sofreram grande variação de $K = \infty$ para $K = 2$ e $K = 0$. Considerando o caso com 5 fluxos de dados e velocidade = 3 m/s, o atraso médio encontrado para a maior taxa de dados simulada foi de 114,8 ms com $K = \infty$. Esse valor sofreu um aumento de 294,4% para o cenário com $K = 2$ e um aumento de 337,8% para o cenário com $K = 0$.

Podemos concluir que o atraso médio por pacote com valor menor que 150 ms, adequado para transmissões de áudio e vídeo, só foi obtido nas simulações com $K = \infty$. Para os outros valores de K , a rede suportaria apenas aplicações tolerantes a atraso, como e-mail e aplicações Web. Dessa forma a rede não seria tão adequada para aplicações como áudio ou vídeo interativo, para as quais é requerido um atraso menor que 150 ms, assim como teleconferências.

4.3.4 Índice de Justiça

O Índice de Justiça mede o quão justo é o compartilhamento da vazão entre os fluxos. Os resultados obtidos estão nos gráficos das Figuras 4.28 a 4.36, para todos os nove cenários de simulação. Foi possível perceber que o compartilhamento do canal é mais justo, ou seja, mais próximo de 1 quanto menor for o número de fluxos. Nota-se também que o compartilhamento do canal é menos justo quando há desvanecimento de pequena escala, ou seja, os valores do índice de Justiça são mais distantes de 1. Observa-se que não houve mudanças significativas do Índice de Justiça para as diferentes taxas de dados utilizadas. Os valores mais próximos de 1 foram alcançados para cenários com $K = \infty$ e com apenas 2 fluxos e o pior valor, ou seja, mais distante de 1 foi alcançado com $K = 0$ para 5 fluxos, como pode ser observado na Figura 4.30.

A análise percentual com relação às diferentes taxas mostra que o índice de justiça decai 32,6% levando em consideração o maior valor resultado das simulações na taxa de envio de dados de 117.760 b/s e o menor valor resultado na taxa de 235.520 b/s. Esse valor foi calculado com relação

à média dos valores de índice de justiça para os três valores de K para as três velocidades de cada taxa de envio de dados. Isto porque os valores do índice de justiça não variam significativamente para as diferentes taxas com relação aos valores de K , por exemplo, os valores para $K = 0$ ficaram em torno de 0,7 para todas as taxas simuladas. Para o número de fluxos, a média feita para todas velocidades das três taxas de envio de dados revelou que o maior valor médio do índice de justiça foi de 0,85 para 2 fluxos e o menor valor médio foi de 0,61 para 5 fluxos, o que demonstra uma queda de 28,2%.

Para a análise do índice de justiça é interessante observar que para uma taxa fixa de 117.760 b/s, a mudança da velocidade dos nós de 0 m/s para 1m/s gera um aumento de 0,11% no índice de justiça, e uma mudança da velocidade de 0 m/s para 3 m/s gera uma diminuição de 6,23% no índice de justiça. Para uma taxa fixa de 176.640 b/s, há um ganho de 1,36% com aumento da velocidade de 0 m/s para 1 m/s e uma diminuição de 5,39% com aumento para 3 m/s. Por último, a taxa de 235.520 b/s causa diminuição no índice de justiça de 1,09% e 7,57% nas duas mudanças de velocidades anteriores. Concluimos então que taxas menores de envio de dados pela aplicação trazem uma melhora na justiça da rede com o aumento da velocidade, até certo valor, após esse valor limite a justiça da rede começa a piorar. Já para taxas maiores de envio de dados, o índice de justiça piora mais rapidamente com o aumento da velocidade.

Por fim, uma última análise geral em relação ao número de fluxos, levando em consideração a média do índice de justiça para todas taxas, velocidades e para todos os valores de K , mostra que em média, o aumento do número de fluxos de 2 para 3 gera uma diminuição de aproximadamente 9,53% no índice de justiça, para um aumento de 3 fluxos para 4 fluxos, o índice de justiça diminui 1,95% e um aumento de 4 fluxos para 5 fluxos causa uma diminuição de 6,13%. Assim percebemos que há uma perda bem menor no índice de justiça quando aumentamos de 3 para 4 fluxos.

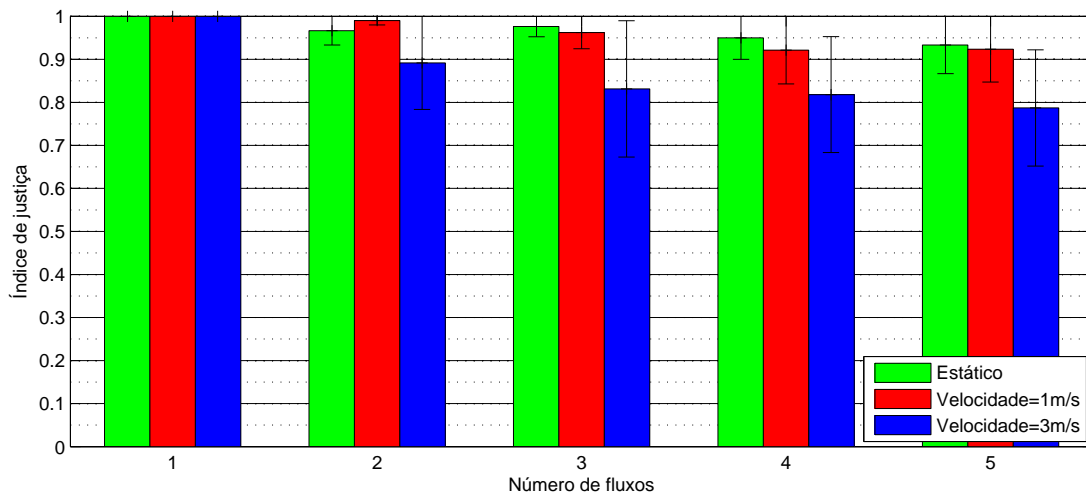


Figura 4.28: Índice de justiça para Cenário 1: $K = \infty$ e taxa= 117.760 b/s.

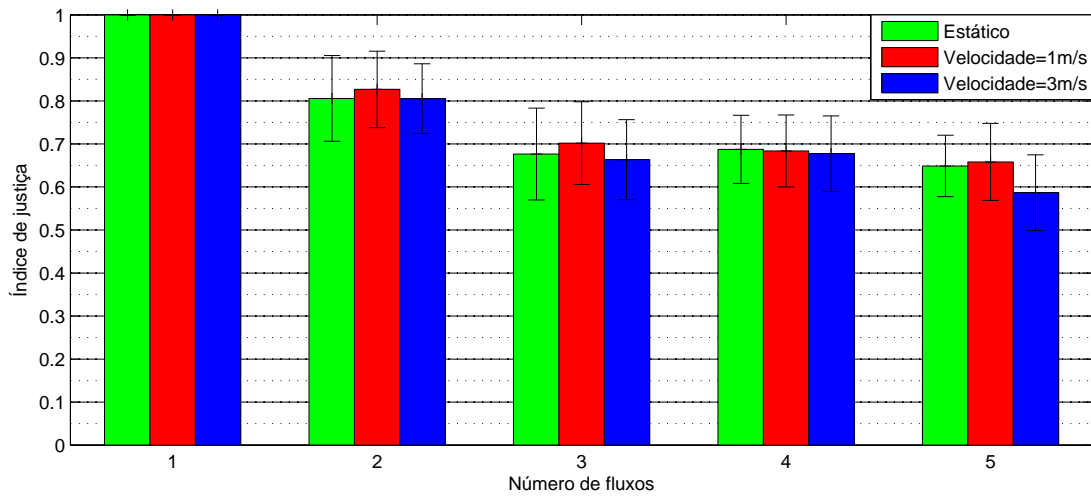


Figura 4.29: Índice de justiça por pacote para Cenário 2: $K = 2$ e taxa= 117.760 b/s.

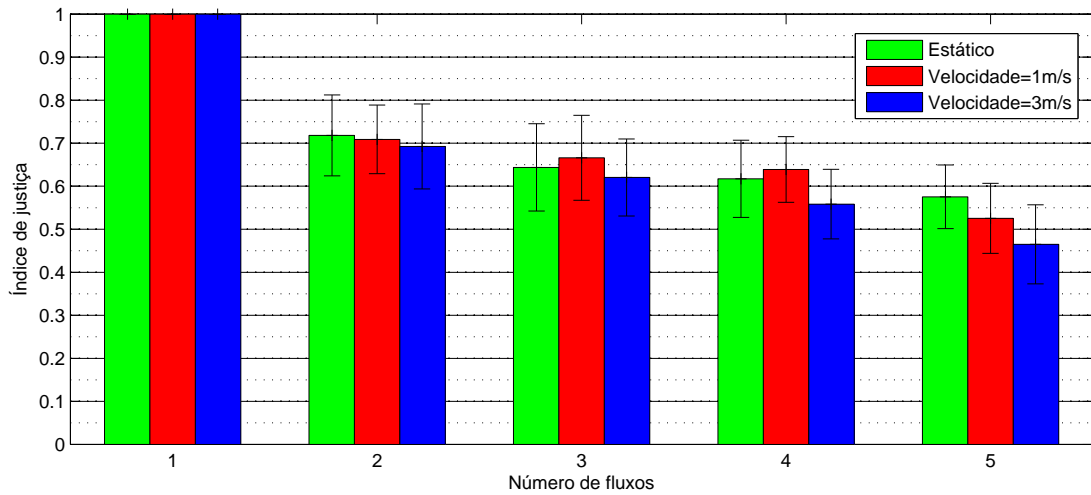


Figura 4.30: Índice de justiça para Cenário 3: $K = 0$ e taxa= 117.760 b/s.

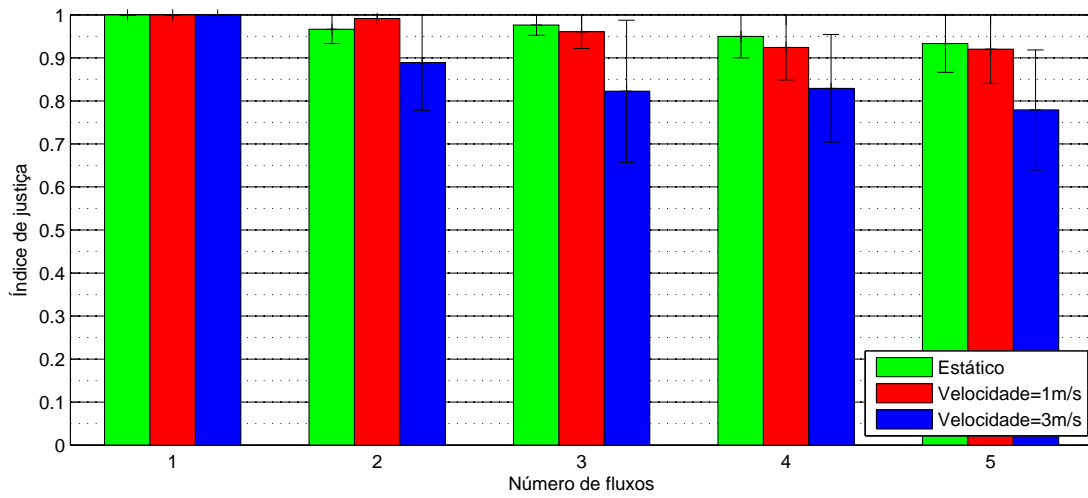


Figura 4.31: Índice de justiça para Cenário 4: $K = \infty$ e taxa= 176.640 b/s.

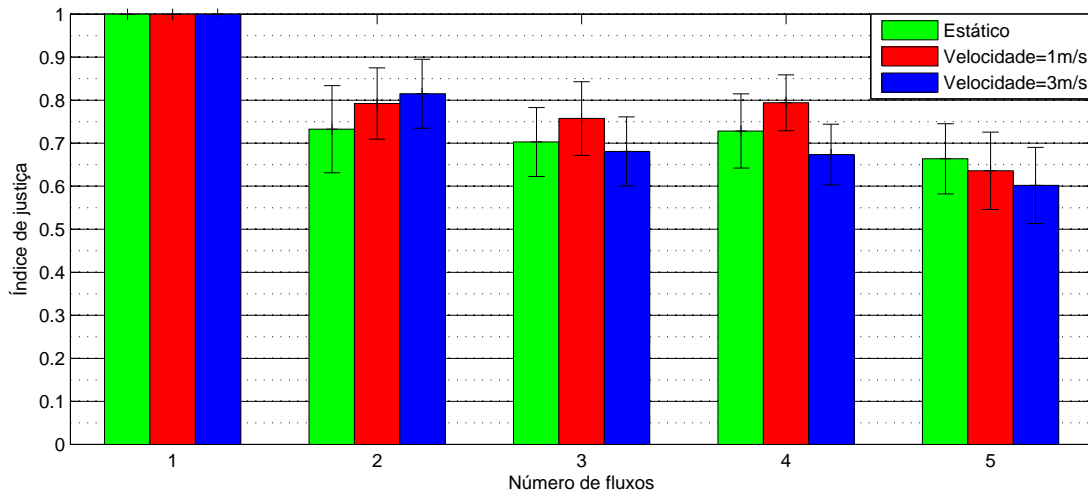


Figura 4.32: Índice de justiça para Cenário 5: $K = 2$ e taxa= 176.640 b/s.

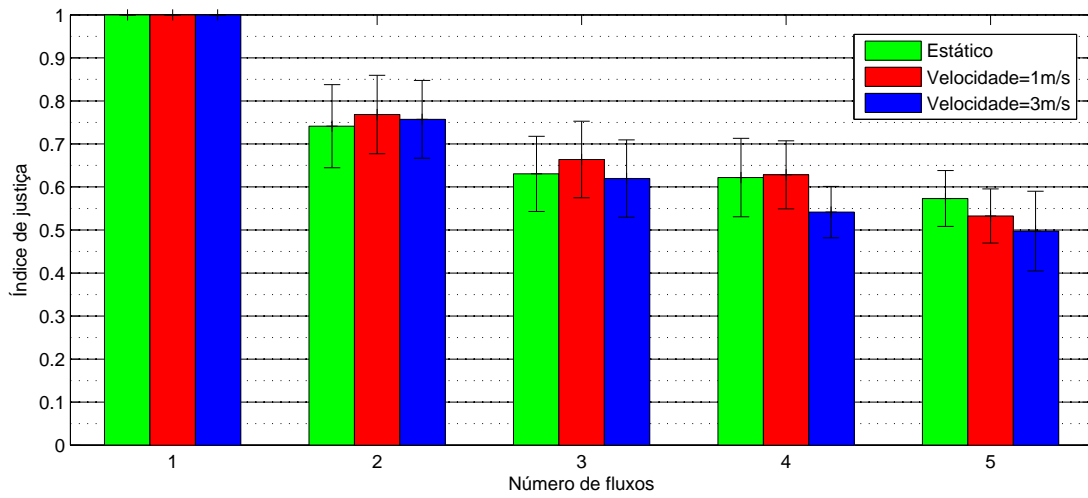


Figura 4.33: Índice de justiça para Cenário 6: $K = 0$ e taxa= 176.640 b/s.

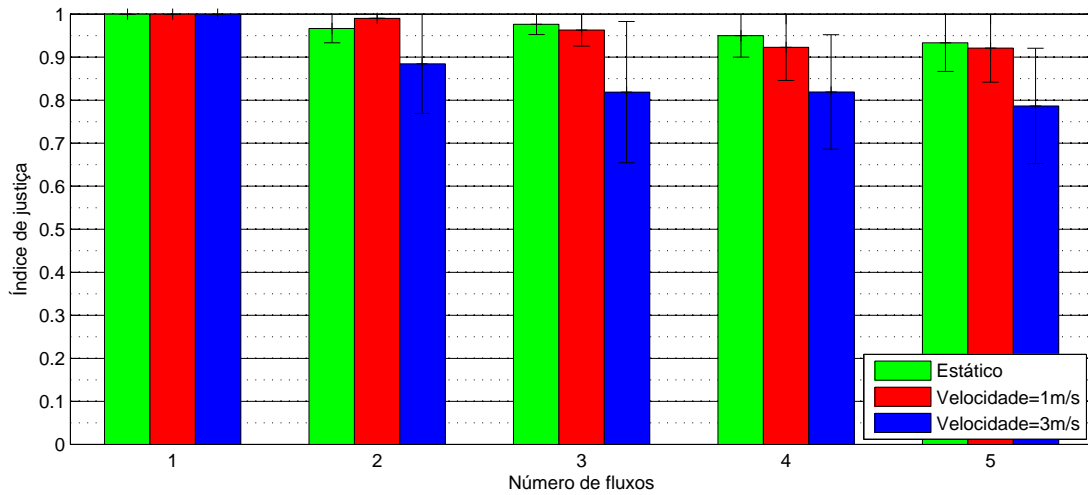


Figura 4.34: Índice de justiça para Cenário 7: $K=\infty$ e taxa= 235.520 b/s.

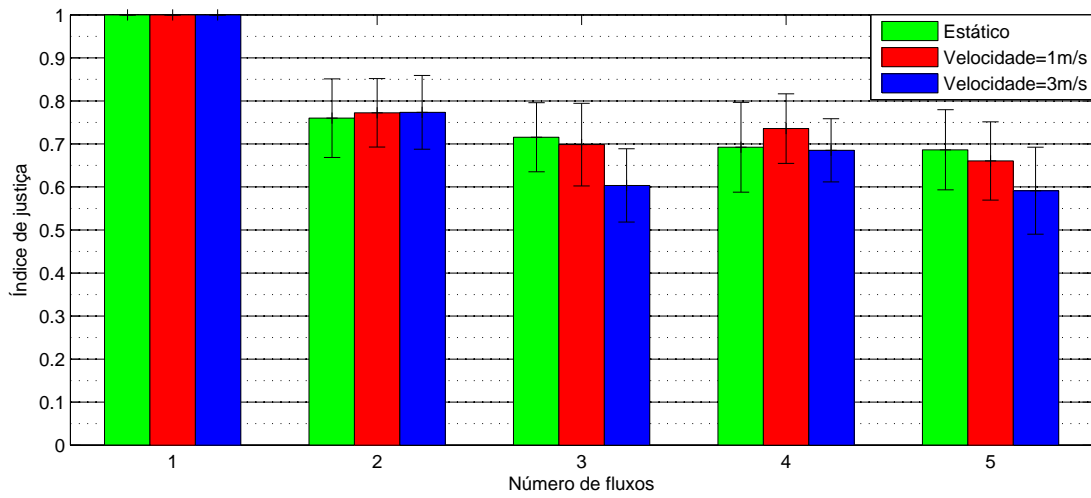


Figura 4.35: Índice de justiça para Cenário 8: $K = 2$ e taxa= 235.520 b/s.

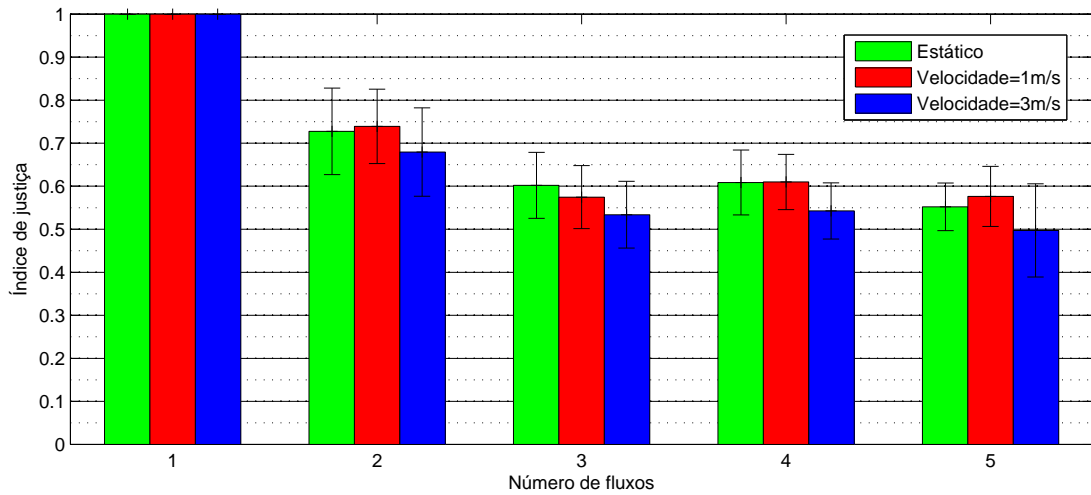


Figura 4.36: Índice de justiça para Cenário 9: $K = 0$ e taxa= 235.520 b/s.

Capítulo 5

Conclusões

Nesse trabalho, foi implementado em ns-3, o escoamento de tráfego de dados móveis de uma rede LTE/4G para uma rede WiFi em modo *ad hoc*. As simulações foram realizadas com diversos cenários, variando-se a taxa de envio de dados da aplicação, a velocidade dos nós, a quantidade de fluxos e o parâmetro de desvanecimento em pequena escala. Com os resultados obtidos para a medição da PDR, vazão, atraso e índice de justiça, foram montados gráficos a fim de que fosse possível analisar o desempenho da rede. No capítulo anterior foi apresentada uma análise qualitativa e quantitativa para os resultados obtidos. Neste capítulo são apresentadas as conclusões obtidas a partir da análise realizada.

De todos os parâmetros analisados para a PDR, a variação na velocidade dos nós foi um dos mais relevantes. Observamos que o aumento da velocidade dos nós provoca o aumento significativo na perda de pacotes. No caso em que os nós estão estáticos, foram obtidas PDRs médias de até 100%, sendo que o menor valor obtido foi de aproximadamente 35%. Já com a velocidade dos nós de 1 m/s, a maior PDR obtida foi de aproximadamente 90% e a menor foi de aproximadamente 20%. Para a velocidade de 3 m/s a perda de pacotes foi ainda maior, pois a maior PDR obtida foi de aproximadamente apenas 53%, enquanto que a menor chegou a aproximadamente 12%. Portanto, as PDRs mais altas foram obtidas para o caso com os nós estáticos, enquanto que as mais baixas foram obtidas para a velocidade de 3 m/s. Assim, a rede simulada seria mais viável em cenários com pouca mobilidade, para que a PDR não sofra diminuições drásticas como foi percebido na velocidade de 3 m/s.

É possível notar que a rede simulada suportaria aplicações como voz, transferência de arquivos e aplicações Web, pois de acordo com os resultados obtidos para vazão média por fluxo, nos melhores casos a vazão obtida foi no máximo 118 kb/s, para a maior taxa de dados da aplicação simulada. Com a vazão alcançada para as taxas de dados consideradas, a rede não suportaria aplicações como vídeo, que requerem uma vazão maior que 100 kb/s. A vazão poderia ser maior com o aumento na taxa de dados da aplicação. Mas como pôde ser visto anteriormente, o aumento da taxa de dados provoca mais perdas de pacotes, maior atraso e menores índices de justiça. Portanto, para obter maiores ganhos com a vazão, seria necessário que as aplicações utilizadas fossem mais tolerantes à perda de pacotes e ao atraso. Além disso, podemos perceber que o aumento do número de fluxos

provocou uma diminuição proporcional na vazão, já que a vazão foi dividida pelo número de fluxos. É importante notar então que o número de fluxos para o nó proxy da rede simulada mais adequado seria de até 3 fluxos, devido às perdas que podem ser provocadas na vazão para quantidades de fluxos maiores que 3.

Quanto ao atraso médio por pacote medido, podemos concluir que a rede simulada seria apropriada para aplicações mais tolerantes à atraso, como email e aplicações Web, já que os valores obtidos menores do que 150 ms foram apenas nos cenários com $K = \infty$. Nos melhores casos para os outros valores de K , o atraso permaneceu na faixa de 300 a 500 ms. Sendo assim, a rede simulada não seria adequada para aplicações sensíveis a atraso ou para aplicações como voz e vídeo que requerem atraso menor de 150 ms. Nota-se também que para os cenários com $K = 2$ e $K = 0$, é possível diminuir o atraso médio por pacote com o aumento do número de fluxos na rede.

Em relação ao índice de justiça, pode-se perceber com a análise realizada, que houve diminuição com o aumento do número de fluxos, como esperado. O menor índice de justiça obtido foi de aproximadamente 0,45, para 5 fluxos. Com a variação da velocidade dos nós, o índice de justiça não apresentou um resultado linear, podendo aumentar ou diminuir com o aumento da velocidade. É interessante notar que na maioria dos casos, podemos observar um índice de justiça maior que 0,6, apresentando então um compartilhamento razoavelmente justo da vazão entre os fluxos.

Com a variação do parâmetro K de Rice, os resultados obtidos demonstraram o efeito do desvanecimento rápido no desempenho da rede. Com a diminuição de K , houve um decaimento na PDR, na vazão e no índice de justiça e um aumento no atraso para todos os cenários simulados, refletindo o efeito das perdas de propagação causadas pelo desvanecimento rápido, e expondo uma das maiores dificuldades encontradas no funcionamento de redes IEEE 802.11. Nota-se também que a diminuição de K provocou maior impacto para a taxa de entrega de pacotes e para o atraso, pois os efeitos do desvanecimento rápido podem provocar perdas de pacotes, aumentando o número de retransmissões dos quadros e provocando então maiores atrasos fim-a-fim.

Com os resultados obtidos para as simulações, foi possível perceber também um dos maiores desafios em MANETs, que são os efeitos provocados pela mobilidade dos nós. Na análise realizada, o prejuízo para o desempenho na rede para todos os parâmetros foi notório com uma maior mobilidade dos nós. Como visto anteriormente, isso é esperado, já que há uma variação imprevisível na topologia da rede, provocando falhas nos enlaces bidirecionais, perdas de rotas, alterações nos nós escolhidos como MPRs, entrada e saída de nós na rede e recálculo das tabelas de roteamento pelo protocolo OLSR.

Como trabalhos futuros sugeridos após o desenvolvimento desse trabalho, podemos incluir a implementação de uma rede *ad hoc* com mais nós proxy funcionando como gargalo para o escoamento de tráfego de dados móveis, pois a rede poderia alcançar um melhor desempenho para a vazão se vários nós fossem escolhidos como nó proxy para fazerem o escoamento da rede celular, ao invés de apenas um. Para isso, seria interessante também implementar algoritmos de seleção de nós proxy, como é feito em [20]. Outros trabalhos interessantes para esse estudo, seria a implementação de outros protocolos de roteamento, como o AODV, para que fosse possível fazer uma análise comparativa entre o desempenho de cada protocolo.

Referências Bibliográficas

- [1] VISUAL Networking Index - Cisco. Acessado em Maio de 2014. Disponível em: <http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html>.
- [2] 4G Américas. Acessado em Maio de 2014. Disponível em: <<http://www.4gamericas.org/index.cfm?fuseaction=page§ionid=2491>>.
- [3] APTILO - WiFi Offload. Acessado em Maio de 2014. Disponível em: <<http://www.aptilo.com/mobile-data-offloading/wifi-offload-3g-4g/>>.
- [4] MOBILE Data Offloading. Acessado em Maio de 2014. Disponível em: <<http://www.mobiledataoffloading.com/mobile-data-offloading-standard-what-is-andsf/>>.
- [5] CISCO Solutions. Acessado em Maio de 2014. Disponível em: <http://www.cisco.com/c/en/us/solutions/collateral/service-provider/service-provider-wi-fi/white_paper_c11-701018.html>.
- [6] AIJAZ, A.; AGHVAMI, H.; AMANI, M. A survey on mobile data offloading: technical and business perspectives. *Wireless Communications, IEEE, IEEE*, v. 20, n. 2, p. 104–112, 2013.
- [7] JIMAA, S.; CHAI, K. K.; CHEN, Y.; ALFADHL, Y. Lte-a an overview and future research areas. In: IEEE. *Wireless and Mobile Computing, Networking and Communications (WiMob), 2011 IEEE 7th International Conference on*. [S.l.], 2011. p. 395–399.
- [8] ABDULLAH, M.; YONIS, A. Performance of lte release 8 and release 10 in wireless communications. In: IEEE. *Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), 2012 International Conference on*. [S.l.], 2012. p. 236–241.
- [9] AKYILDIZ, I. F.; GUTIERREZ-ESTEVEZ, D. M.; REYES, E. C. The evolution to 4g cellular systems: Lte-advanced. *Physical Communication*, Elsevier, v. 3, n. 4, p. 217–244, 2010.
- [10] ZHIOUA, G. E. M.; LABIOD, H.; TABBANE, N.; TABBANE, S. Lte advanced relaying standard: A survey. *Wireless personal communications*, Springer, v. 72, n. 4, p. 2445–2463, 2013.
- [11] GAST, M. *802.11 wireless networks: the definitive guide*. [S.l.]: O'Reilly Media, Inc., 2005.

- [12] CROW, B. P.; WIDJAJA, I.; KIM, J. G.; SAKAI, P. T. Ieee 802.11 wireless local area networks. *Communications Magazine, IEEE*, IEEE, v. 35, n. 9, p. 116–126, 1997.
- [13] WALKE, B. H.; MANGOLD, S.; BERLEMANN, L. *IEEE 802 wireless systems: protocols, multi-hop mesh/relaying, performance and spectrum coexistence*. [S.l.]: John Wiley & Sons, 2007.
- [14] KUROSE, J.; ROSS, K. *Redes de computadores e internet*. São Paulo: Person, 2006.
- [15] OPNET. Acessado em Maio de 2014. Disponível em: <<http://www.riverbed.com/products/performance-management-control/opnet.html>>.
- [16] NS-3, the network simulator. Acessado em Maio de 2014. Disponível em: <<http://www.nsnam.org/>>.
- [17] CHLAMTAC, I.; CONTI, M.; LIU, J. J.-N. Mobile ad hoc networking: imperatives and challenges. *Ad Hoc Networks*, Elsevier, v. 1, n. 1, p. 13–64, 2003.
- [18] JACQUET, P.; MUHLETHALER, P.; CLAUSEN, T.; LAOUITI, A.; QAYYUM, A.; VIENNOT, L. Optimized link state routing protocol for ad hoc networks. In: IEEE. *Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International*. [S.l.], 2001. p. 62–68.
- [19] LEE, K.; LEE, J.; YI, Y.; RHEE, I.; CHONG, S. Mobile data offloading: how much can wifi deliver? In: ACM. *Proceedings of the 6th International Conference*. [S.l.], 2010. p. 26.
- [20] HAN, B.; HUI, P.; KUMAR, V.; MARATHE, M. V.; PEI, G.; SRINIVASAN, A. Cellular traffic offloading through opportunistic communications: a case study. In: ACM. *Proceedings of the 5th ACM workshop on Challenged networks*. [S.l.], 2010. p. 31–38.
- [21] BALASUBRAMANIAN, A.; MAHAJAN, R.; VENKATARAMANI, A. Augmenting mobile 3g using wifi. In: ACM. *Proceedings of the 8th international conference on Mobile systems, applications, and services*. [S.l.], 2010. p. 209–222.
- [22] LUO, H.; RAMJEE, R.; SINHA, P.; LI, L. E.; LU, S. Ucan: a unified cellular and ad-hoc network architecture. In: ACM. *Proceedings of the 9th annual international conference on Mobile computing and networking*. [S.l.], 2003. p. 353–367.
- [23] RAPPAPORT, T. S. *Wireless Communications: Principles and Practice*. 2. ed. New Jersey, NJ, EUA: Prentice Hall, 2002.
- [24] MOLISCH, A. F. *Wireless communications*. [S.l.]: John Wiley & Sons, 2010.
- [25] JAIN, R.; CHIU, D.-M.; HAWKES, W. R. *A quantitative measure of fairness and discrimination for resource allocation in shared computer system*. [S.l.]: Eastern Research Laboratory, Digital Equipment Corporation Hudson, MA, 1984.
- [26] CACHEDA, R. A. et al. Qos requirements for multimedia services. In: *Resource Management in Satellite Networks*. [S.l.]: Springer, 2007. p. 67–94.

ANEXOS

I. SCRIPT

```
/* -*- Mode: C++; c-file-style: "gnu"; indent-tabs-mode:nil; -*- */
/*
 *
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License version 2 as
 * published by the Free Software Foundation;
 *
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
 *
 *
 * Authors: Camila Lumy Nakano <milalumy@gmail.com>
 * Thayane Rodrigues Viana <thayaneviana@hotmail.com>
 */
#include <fstream>
#include <string.h>
#include "ns3/csma-helper.h"
#include "ns3/csma-net-device.h"
#include "ns3/lte-helper.h"
#include "ns3/epc-helper.h"
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/ipv4-global-routing-helper.h"
#include "ns3/internet-module.h"
#include "ns3/mobility-module.h"
#include "ns3/lte-module.h"
#include "ns3/applications-module.h"
#include "ns3/point-to-point-helper.h"
#include "ns3/netanim-module.h"
#include "ns3/olsr-module.h"
```

```

#include "ns3/ipv4-static-routing-helper.h"
#include "ns3/config-store-module.h"
#include "ns3/wifi-module.h"
#include "ns3/flow-monitor-module.h"
#include "ns3/pointer.h"
#include <iostream>
#include <vector>
#include <cstdio>
#include <cstdlib>
#include <stdexcept>
#include <ctime>
#include <iomanip>
using namespace ns3;
using namespace std;
NS_LOG_COMPONENT_DEFINE ("LTE-Wifi");
const double PI = 3.14159265358979323846;
const double lambda = (3.0e8 / 2.407e9);
bool useTwoRay = true;
double txPowerDbm = 10;
double factorCca = 1.5;
double distancia = 150;
double height = 1.2;
double stop = 90;
double warmup = 30;
void PrintRoutingTable (Ptr<Node> node);
double rxPowerDbm (double distance, double height, double txPowerDbm, bool useTwoRay);
void CourseChange (std::string context, Ptr<const MobilityModel> model);
void ComputeResults (void);
struct sim_Result
{
uint64_t sumRxBytesByFlow;
uint64_t sumRxBytesQuadByFlow;
uint64_t sumLostPktsByFlow;
uint64_t sumRxPktsByFlow;
uint64_t sumTxPktsByFlow;
uint64_t sumDelayFlow;
uint64_t nFlows;
/* Throughput Average by Flow (bps) = sumRxBytesByFlow * 8 / (nFlows * time)
* Throughput Quadratic Average by Flow (bps) = sumRxBytesQuadByFlow * 64 / (nFlows * time * time)

```

```

* Net Aggregated Throughput Average by Node (bps) = sumRxBytesByFlow * 8 / (nodes * time)
* Fairness = sumRxBytesByFlow2 / (nFlows * sumRxBytesQuadByFlow)
* Delay per Packet (seconds/packet) = sumDelayFlow / sumRxPktsByFlow
* Lost Ratio (%) = 100 * sumLostPktsByFlow / sumTxPktsByFlow
*/
double thrpAvgByFlow;
double thrpAvgQuadByFlow;
double thrpVarByFlow;
double netThrpAvgByNode;
double fairness;
double delayByPkt;
double lostRatio;
double pdr;
sim_Result ()
{
sumRxBytesByFlow = 0;
sumRxBytesQuadByFlow = 0;
sumLostPktsByFlow = 0;
sumRxPktsByFlow = 0;
sumTxPktsByFlow = 0;
sumDelayFlow = 0;
nFlows = 0;
}
} data;
//===== COMEÇO DA MAIN =====
int
main (int argc, char *argv[])
{
//===== LOG's =====
//LogComponentEnable ("OnOffApplication", LOG_LEVEL_INFO);
//LogComponentEnable ("FlowMonitor", LOG_LEVEL_ALL);
//LogComponentEnable ("OlsrRoutingProtocol", LOG_LEVEL_ALL);
//LogComponentEnable ("OlsrRoutingTable", LOG_LEVEL_ALL);
//LogComponentEnable ("Ipv4StaticRouting", LOG_LEVEL_ALL);
//Aparecer no arquivo externo: ./waf -run olsr-lte > log.out 2>&1
//===== PARÂMETROS =====
uint16_t numberOfNodes = 1;
uint16_t numWifi = 25;
uint16_t port = 80;

```

```

uint32_t packetSize = 1472;
uint32_t numFluxos = 1;
double distance = 10.0;
double RxNoiseFiguredB = 7;
uint32_t dist = 600;
double dataRate = 117760;
//double dataRate_f = (double) dataRate / numFluxos;
FILE *top;
double px,py,pz;
top = fopen("Topologia/Topologia1/Topologia1.txt","r");
double k = 0;
std::string phyMode ("DsssRate1Mbps");
Config::SetDefault ("ns3::ArpCache::PendingQueueSize", UintegerValue (400));
Config::SetDefault ("ns3::LteRlcUm::MaxTxBufferSize", UintegerValue (10000000));
CommandLine cmd;
cmd.AddValue ("numFluxos", "Numero de fluxos", numFluxos);
cmd.AddValue ("dataRate", "Taxa de tráfeço", dataRate);
cmd.AddValue ("k", "K", k);
cmd.Parse (argc, argv);
//===== LTE Helpers =====
Ptr<LteHelper> lteHelper = CreateObject<LteHelper> ();
Ptr<EpcHelper> epcHelper = CreateObject<EpcHelper> ();
lteHelper->SetEpcHelper (epcHelper);
lteHelper->SetSchedulerType("ns3::PffMacScheduler");
Ptr<Node> pgw = epcHelper->GetPgwNode ();
//===== LTE Remote Host e p2p interno =====
// Remoto
NodeContainer remoteHostContainer;
remoteHostContainer.Create (1);
Ptr<Node> remoteHost = remoteHostContainer.Get (0);
// p2p interno
PointToPointHelper p2ph;
p2ph.SetDeviceAttribute ("DataRate", DataRateValue (DataRate ("100Gb/s")));
p2ph.SetDeviceAttribute ("Mtu", UintegerValue (1500));
p2ph.SetChannelAttribute ("Delay", TimeValue (Seconds (0.010))); //0.010
NetDeviceContainer internetDevices = p2ph.Install (pgw, remoteHost);
// Internet
InternetStackHelper internet;
internet.Install (remoteHost);

```

```

//===== Criação dos nós LTE-p2p-wifi =====
NodeContainer ueNodes;
NodeContainer enbNodes;
enbNodes.Create(numberOfNodes);
ueNodes.Create(numberOfNodes);
NodeContainer p2pNodes;
p2pNodes.Create (1);
p2pNodes.Add(ueNodes.Get(0));
NodeContainer wifiNodes;
wifiNodes.Create (numWifi);
wifiNodes.Add(p2pNodes.Get(0));
NodeContainer allNodes;
allNodes.Add(remoteHostContainer.Get(0));
allNodes.Add(enbNodes.Get(0));
allNodes.Add(ueNodes.Get(0));
allNodes.Add(p2pNodes.Get(0));
// Ponteiros
Ptr<Node> ueNode = ueNodes.Get (0);
// Internet
internet.Install (ueNodes);
OlsrHelper olsr;
Ipv4StaticRoutingHelper ipv4static;
Ipv4ListRoutingHelper list;
list.Add (olsr, 10);
list.Add (ipv4static, 10);
internet.SetRoutingHelper(olsr);
for (uint16_t i = 0; i < numWifi; i++)
{
internet.Install (wifiNodes.Get(i));
allNodes.Add(wifiNodes.Get(i));
}
internet.Reset();
internet.SetRoutingHelper(list);
internet.Install (wifiNodes.Get(numWifi));
//===== Mobilidade =====
Ptr<ListPositionAllocator> positionAllocLTE = CreateObject<ListPositionAllocator> ();
Ptr<ListPositionAllocator> positionAllocWIFI = CreateObject<ListPositionAllocator> ();
//LTE
for (uint16_t i = 0; i < numberOfNodes + 1; i++)

```

```

{
positionAllocLTE->Add (Vector(distance * i, 0, 0));
}

//reads node's position from file
for(int i=0;i<numWifi + 1;i++)
{
fscanf(top, "%lf %lf %lf",&px,&py,&pz);
positionAllocWIFI->Add(Vector(px,py,pz));
}

//WIFI
//Modelo
MobilityHelper mobility;
mobility.SetMobilityModel("ns3::ConstantPositionMobilityModel");
mobility.SetPositionAllocator(positionAllocLTE);
mobility.Install(enbNodes);
mobility.Install(ueNodes);
mobility.SetPositionAllocator(positionAllocWIFI);
mobility.SetMobilityModel ("ns3::RandomDirection2dMobilityModel",
"Bounds", RectangleValue (Rectangle (0, dist, 0, dist)),
"Speed", StringValue ("ns3::ConstantRandomVariable[Constant=1.0]"),
"Pause", StringValue ("ns3::ExponentialRandomVariable[Mean=0.5]"));
mobility.Install(wifiNodes);

//Imprime a posição dos nós
for (int j=numWifi; j>0; j-){
std::ostringstream oss;

oss «
"/NodeList/«< wifiNodes.Get (numWifi - j)->GetId () «
"/$ns3::MobilityModel/CourseChange";
Config::Connect (oss.str (), MakeCallback (&CourseChange));
}

//mobility tracing
AsciiTraceHelper ascii;
MobilityHelper::EnableAsciiAll (ascii.CreateFileStream ("Topologia/Topologia1/v2/mob-olsr/lte-wifi-OLSR.mob"));

//===== LTE devices e Ipv4 para o UE =====
//Devices
NetDeviceContainer enbLteDevs = lteHelper->InstallEnbDevice (enbNodes);
NetDeviceContainer ueLteDevs = lteHelper->InstallUeDevice (ueNodes);

//Ipv4 para o UE
Ipv4InterfaceContainer ueIpIface;

```

```

ueIpIface = epcHelper->AssignUeIpv4Address (NetDeviceContainer (ueLteDevs));
//Infra-estrutura
lteHelper->Attach (ueLteDevs.Get(0), enbLteDevs.Get(0));
//===== Point to Point devices =====
PointToPointHelper pointToPoint;
pointToPoint.SetDeviceAttribute ("DataRate", DataRateValue (DataRate ("100Gb/s")));
pointToPoint.SetDeviceAttribute ("Mtu", UIntegerValue (1500));
pointToPoint.SetChannelAttribute ("Delay", TimeValue (Seconds (0.010)));
NetDeviceContainer p2pDevices = pointToPoint.Install (p2pNodes.Get(0), p2pNodes.Get(1));
//===== Adhoc devices =====
//*****Wifi*****
WifiHelper wifi;
wifi.SetStandard (WIFI_PHY_STANDARD_80211b);
/*wifi.SetRemoteStationManager ("ns3::ConstantRateWifiManager",
"DataMode", StringValue ("DsssRate1Mbps"),
"RtsCtsThreshold", UIntegerValue (1000));*/
wifi.SetRemoteStationManager ("ns3::ConstantRateWifiManager",
"DataMode",StringValue (phyMode),
"ControlMode",StringValue (phyMode));
// turn off RTS/CTS for frames below 1000 bytes
Config::SetDefault ("ns3::WifiRemoteStationManager::RtsCtsThreshold", StringValue ("0"));
//*****Camada PHY*****
YansWifiPhyHelper wifiPhy = YansWifiPhyHelper::Default ();
wifiPhy.Set ("TxPowerStart", DoubleValue(txPowerDbm));
wifiPhy.Set ("TxPowerEnd", DoubleValue(txPowerDbm));
wifiPhy.Set ("TxGain", DoubleValue(0));
wifiPhy.Set ("RxGain", DoubleValue(0));
wifiPhy.Set ("EnergyDetectionThreshold", DoubleValue(rxPowerDbm (distancia, height, txPowerDbm, useTwoRay)));
wifiPhy.Set ("CcaMode1Threshold", DoubleValue(rxPowerDbm (distancia*factorCca, height,txPowerDbm, useTwoRay)));
wifiPhy.Set ("RxNoiseFigure", DoubleValue(RxNoiseFiguredB));
cout << "EnergyDetectionThresold:<< rxPowerDbm (distancia, height,txPowerDbm, useTwoRay) << endl;
cout << "CcaMode1Thresold:<< rxPowerDbm (distancia*factorCca, height,txPowerDbm, useTwoRay) << endl;
wifiPhy.SetPcapDataLinkType (YansWifiPhyHelper::DLT_IEEE802_11_RADIO);
//*****Channel*****
YansWifiChannelHelper wifiChannel;
wifiChannel.SetPropagationDelay ("ns3::ConstantSpeedPropagationDelayModel");
wifiChannel.AddPropagationLoss ("ns3::TwoRayGroundPropagationLossModel",
"Frequency", DoubleValue(2.407e9));
//wifiChannel.AddPropagationLoss ("ns3::FixedRssLossModel", "Rss",DoubleValue (rss));

```



```

//***** Desvanecimento *****//
double m;
m = (pow((k+1),2)) / (2*k+1); // "Wireless Communications"(Molisch)
wifiChannel.AddPropagationLoss ("ns3::NakagamiPropagationLossModel",
"m0", DoubleValue (m),"m1", DoubleValue (m),"m2", DoubleValue (m));
wifiPhy.SetChannel (wifiChannel.Create ());
//*****Camada MAC*****
NqosWifiMacHelper wifiMac = NqosWifiMacHelper::Default ();
wifiMac.SetType ("ns3::AdhocWifiMac");
//*****Install*****
NetDeviceContainer devices = wifi.Install (wifiPhy, wifiMac, wifiNodes);
//NetDeviceContainer deviceC = wifi.Install (wifiPhy, wifiMac, p2pNodes.Get(0));
//===== Ipv4 restante =====
Ipv4AddressHelper ipv4;
//RemoteHost
ipv4.SetBase ("1.0.0.0", "255.0.0.0");
Ipv4InterfaceContainer internetIpIfaces = ipv4.Assign (internetDevices);
//ADHOC
ipv4.SetBase ("172.1.1.0", "255.255.255.0");
Ipv4InterfaceContainer i = ipv4.Assign (devices);
//P2P
ipv4.SetBase ("7.1.1.0", "255.255.255.0");
Ipv4InterfaceContainer p2pInterfaces = ipv4.Assign (p2pDevices);
//===== Roteamento Estático =====
//Remote Host
Ipv4StaticRoutingHelper ipv4RoutingHelper;
Ptr<Ipv4StaticRouting> remoteHostStaticRouting =
ipv4RoutingHelper.GetStaticRouting (remoteHost->GetObject<Ipv4> ());
remoteHostStaticRouting->AddNetworkRouteTo (Ipv4Address ("7.0.0.0"), Ipv4Mask ("255.0.0.0"), 1);
remoteHostStaticRouting->AddNetworkRouteTo (Ipv4Address ("172.1.1.0"), Ipv4Mask ("255.255.255.0"), 1);
//PGW
Ptr<Ipv4StaticRouting> remoteHostStaticRouting79;
remoteHostStaticRouting79 = ipv4RoutingHelper.GetStaticRouting (pgw->GetObject<Ipv4> ());
remoteHostStaticRouting79->AddNetworkRouteTo (Ipv4Address ("172.1.1.0"),
Ipv4Mask ("255.255.255.0"), Ipv4Address ("0.0.0.0"), 1);
//UE node - (Relay de LTE/p2p)
Ptr<Ipv4StaticRouting> remoteHostStaticRouting2;
remoteHostStaticRouting2 = ipv4RoutingHelper.GetStaticRouting (ueNode->GetObject<Ipv4> ());
remoteHostStaticRouting2->AddNetworkRouteTo Ipv4Address ("1.0.0.0"), Ipv4Mask ("255.0.0.0"),

```

```

epcHelper->GetUeDefaultGatewayAddress (), 1);
remoteHostStaticRouting2->AddNetworkRouteTo Ipv4Address ("172.1.1.0"), Ipv4Mask ("255.255.255.0"),
Ipv4Address ("7.1.1.1"), 2);
//p2p(0) - (Relay de p2p/WIFI)
Ptr<Ipv4StaticRouting> remoteHostStaticRouting89;
remoteHostStaticRouting89 = ipv4RoutingHelper.GetStaticRouting (p2pNodes.Get(0)->GetObject<Ipv4> ());
remoteHostStaticRouting89->AddHostRouteTo (Ipv4Address ("1.0.0.2"), Ipv4Address ("7.1.1.2"), 2);
//WIFI
/*Ptr<Ipv4StaticRouting> remoteHostStaticRouting65;
remoteHostStaticRouting65 = ipv4RoutingHelper.GetStaticRouting (wifiNodes.Get(numCliente)->GetObject<Ipv4> ());
remoteHostStaticRouting65->AddHostRouteTo (Ipv4Address ("1.0.0.2"), i.GetAddress (numWifi), 1);*/
//===== Traces e Logs =====
pointToPoint.EnableAsciiAll (ascii.CreateFileStream ("Topologia/Topologia1/v2/trace-olsr/lte-wifi.tr-OLSR"));
pointToPoint.EnablePcapAll ("Topologia/Topologia1/v2/pcap-p2p-olsr/LTE-WIFI-p2p-OLSR");
wifiPhy.EnablePcap ("Topologia/Topologia1/v2/pcap-olsr/LTE-WIFI-OLSR", devices);
//Ptr<OutputStreamWrapper> stream = ascii.CreateFileStream ("routingtables.tr");
//remoteHostStaticRouting->PrintRoutingTable(stream);
//lteHelper->EnableTraces ();
//===== Aplicação =====
for (uint32_t n=19; n<19 +numFluxos; n++)
{
OnOffHelper onoff ("ns3::UdpSocketFactory",Address (InetSocketAddress (i.GetAddress (n), port))); //Destination
onoff.SetAttribute ("Remote", AddressValue(InetSocketAddress(i.GetAddress(n),port))); //Destination
onoff.SetAttribute ("OnTime", StringValue ("ns3::ConstantRandomVariable[Constant=1.0]"));
onoff.SetAttribute ("OffTime", StringValue ("ns3::ConstantRandomVariable[Constant=1.0]"));
onoff.SetAttribute ("DataRate", DataRateValue (DataRate (dataRate / numFluxos)));
onoff.SetAttribute("PacketSize", UIntegerValue (packetSize));
ApplicationContainer app = onoff.Install (remoteHostContainer.Get(0)); //Source
app.Start (Seconds (warmup+0.1));
app.Stop (Seconds (stop + 0.1));
PacketSinkHelper sink("ns3::UdpSocketFactory",InetSocketAddress(internetIpIfaces.GetAddress(0), port));//Source
ApplicationContainer sinkApp = sink.Install(wifiNodes.Get(n)); //Destination
sinkApp.Start(Seconds(warmup+0.1));
sinkApp.Stop(Seconds(stop + 0.1));
}
//===== Run e Outputs =====
PrintRoutingTable(wifiNodes.Get(numWifi));
PrintRoutingTable(pgw);
PrintRoutingTable(enbNodes.Get(0));

```

```

PrintRoutingTable(p2pNodes.Get(1));
PrintRoutingTable(p2pNodes.Get(0));*/
//===== Flow Monitor =====
FlowMonitorHelper flowmon;
Ptr<FlowMonitor> monitor = flowmon.Install(allNodes);
monitor->Start (Seconds (warmup)); // start monitoring after network warm up
monitor->Stop (Seconds (stop + 3)); // stop monitoring
Simulator::Stop(Seconds(stop + 3));
Simulator::Run ();
Ptr<Ipv4FlowClassifier> classifier = DynamicCast<Ipv4FlowClassifier> (flowmon.GetClassifier());
std::map<FlowId, FlowMonitor::FlowStats> stats = monitor->GetFlowStats();
for (std::map<FlowId, FlowMonitor::FlowStats>::const_iterator i = stats.begin (); i != stats.end (); ++i)
{
Ipv4FlowClassifier::FiveTuple t = classifier->FindFlow (i->first);
std::cout << "Flow << i->first << "("<< t.sourceAddress << "-> << t.destinationAddress << ")\\n";
std::cout << "Tx Bytes: << i->second.txBytes << "\\n";
std::cout << "Rx Bytes: << i->second.rxBytes << "\\n";
std::cout << "Throughput: << i->second.rxBytes * 8.0 / (stop - warmup) / 1024 / 1024 << "Mbps\\n";
std::cout << "Packet Delivery Ratio: << ((double)(i->second.rxPackets)/((double)(i->second.txPackets))*100 << "%\\n";
if (t.destinationPort == 80) // only http flows
{
data.nFlows++;
data.sumRxBytesByFlow += i->second.rxBytes; // sum flows
data.sumRxBytesQuadByFlow += i->second.rxBytes * i->second.rxBytes; // sum flows2
data.sumDelayFlow += i->second.delaySum.GetInteger (); // sum delays
data.sumRxPktsByFlow += i->second.rxPackets; // sum rx pkts
data.sumTxPktsByFlow += i->second.txPackets; // sum tx pkts
data.sumLostPktsByFlow += i->second.lostPackets; // sum lost pkts
}
}
monitor->SerializeToXmlFile ("Topologia/Topologia1/v2/FlowMonitor/FlowMonitor LteWifi OLSR.xml", true, true);
monitor->CheckForLostPackets();
Simulator::Destroy ();
ComputeResults ();
return 0;
}
double
rxPowerDbm (double distance, double height, double txPowerDbm, bool useTwoRay)
{

```

```

double lossPowerDbm;
if (useTwoRay){
double dCross = (4 * PI * height * height) / lambda;
if (distance <= dCross){
lossPowerDbm = 10 * log10( lambda*lambda / (16.0 * PI * PI * distance*distance));
} else {
lossPowerDbm = 10 * log10( (height*height*height*height) / (distance*distance*distance*distance) );
}
}
else {
lossPowerDbm = 10 * log10( lambda*lambda / (16.0 * PI * PI * distance*distance));
}
return txPowerDbm + lossPowerDbm;
}
void
CourseChange (std::string context, Ptr<const MobilityModel> model)
{
Vector position = model->GetPosition ();
NS_LOG_UNCOND (context «
"x = «< position.x « ", y = «< position.y);
}
void
PrintRoutingTable (Ptr<Node> node)
{
Ipv4StaticRoutingHelper helper;
Ptr<Ipv4> stack = node -> GetObject<Ipv4>();
Ptr<Ipv4StaticRouting> staticrouting = helper.GetStaticRouting(stack);
uint32_t numroutes=staticrouting->GetNRoutes();
Ipv4RoutingTableEntry entry;
std::cout « "Routing table for device: «< Names::FindName(node) «"\n";
std::cout « "Destination\t\tMask\t\t
\tGateway\t\t\tInterface\n";
for (uint32_t i =0 ; i<numroutes;i++) {
entry =staticrouting->GetRoute(i);
std::cout « entry.GetDestNetwork() « "\t\t«< entry.GetDestNetworkMask() « "\t\t\t«< entry.GetGateway() «
"\t\t\t«< entry.GetInterface() « "\n"; }
return; }
void
ComputeResults (void)

```

```

{
double deltaT = (stop - warmup);
// Throughput Average by Flow (bps)
data.thrpAvgByFlow = (double) data.sumRxBytesByFlow * 8 / (data.nFlows * deltaT);
// Throughput Quadratic Average by Flow (bps2)
data.thrpAvgQuadByFlow = (double) data.sumRxBytesQuadByFlow * 8*8 / (data.nFlows * deltaT*deltaT);
// Throughput Variance by Flow (bps2)
data.thrpVarByFlow = data.thrpAvgQuadByFlow - data.thrpAvgByFlow * data.thrpAvgByFlow;
// Network Aggregated Throughput Average by Node (bps)
//data.netThrpAvgByNode = (double) data.sumRxBytesByFlow * 8 / (numFluxos * deltaT);
// Fairness Jain's Index
data.fairness = (double) data.sumRxBytesByFlow * data.sumRxBytesByFlow / (data.nFlows * data.sumRxBytesQuadByFlow);
// Delay Mean by Packet (nanoseconds)
data.delayByPkt = (double) data.sumDelayFlow / data.sumRxPktsByFlow;
// Lost Ratio (%)
data.lostRatio = (double) 100 * data.sumLostPktsByFlow / data.sumTxPktsByFlow;
data.pdr = (double) 100 * data.sumRxPktsByFlow / data.sumTxPktsByFlow;
cout << "=====«< endl
« "Simulation results:«< endl
« "Throughput Average by Flow (kbps):\t«< data.thrpAvgByFlow / 1024.0 « endl
« "Throughput Deviation by Flow (kbps):\t«< sqrt (data.thrpVarByFlow) / 1024.0 « endl
// « "Network Aggregated Throughput Average by Node (kbps):\t«< data.netThrpAvgByNode / 1024.0 « endl
« "Fairness Jain's Index:\t«< data.fairness « endl
« "Delay Mean by Packet (seconds):\t«< data.delayByPkt / 1e9 « endl
« "Packet Lost Ratio (%):\t«< data.lostRatio « endl
« "Packet Delivery Ratio (%):\t«< data.pdr«endl« endl« endl;
cout << "Flows: «< data.nFlows « endl;
ofstream throughput;
throughput.open ("Topologia/Topologia1/v2/FlowMonitor/throughput-olsr/throughput-olsr-v2", ios::app);
throughput << "\t«< data.nFlows « "\t«< std::setprecision(4) « data.thrpAvgByFlow / 1024.0 « endl;
throughput.close ();
ofstream delay;
delay.open ("Topologia/Topologia1/v2/FlowMonitor/delay-olsr/delay-olsr-v2", ios::app);
delay << "\t«< data.nFlows « "\t«< std::setprecision(4) « data.delayByPkt / 1e6 « endl;
delay.close ();
ofstream fairness;
fairness.open ("Topologia/Topologia1/v2/FlowMonitor/fairness-olsr/fairness-olsr-v2", ios::app);
fairness << "\t«< data.nFlows « "\t«< std::setprecision(4) « data.fairness « endl;
fairness.close ();

```

```
ofstream pdr;
pdr.open ("Topologia/Topologia1/v2/FlowMonitor/PDR-olsr/pdr-olsr-v2", ios::app);
pdr << "\t<< data.nFlows << "\t<< std::setprecision(4) << data.pdr << endl;
pdr.close ();
ofstream packets;
packets.open ("Topologia/Topologia1/v2/FlowMonitor/packets-olsr/packets-olsr-v2", ios::app);
packets << "\t<< data.sumTxPktsByFlow << "\t<< std::setprecision (4) << data.sumRxPktsByFlow << endl;
packets.close (); }
```